Intelligent methods to identify incorrect reviews in cloud reputation systems

Quynh Ngoc Thuy Do

Supervisor: Associate Professor Farookh Khadeer Hussain

Co-supervisor: Doctor Christy Jie Liang

University of Technology Sydney

Faculty of Engineering and Information Technology

Centre for Artificial Intelligence

25th August, 2019

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Quynh Ngoc Thuy Do declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy degree, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 25th August 2019

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my principal supervisor, Associate Professor Farookh Khadeer Hussain for his support, kindness, gratitude, motivation and wisdom. Without his help, I could not have completed this journey. He not only guided my research and provided advice on writing the thesis and academic papers, he also provided guidance in every aspect of life. Words are not enough to thank him for everything he has done for me. Secondly, I would like to express my appreciation to my co-supervisor, Doctor Christy Jie Liang, who supported me until the very end of my PhD journey.

I would like to thank Dong Nai province, Vietnam for awarding me a scholarship. Many thanks to the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology for awarding me travel grants to present my research work.

I would like to thank my friends and colleagues: Khimmy, Raza, Asma, Walayat, Ikram and Yahya with whom I had a memorable time at UTS. I would like to thank Ms Michele Mooney for her services in proofreading this thesis.

Finally, I would like to thank my family, without whom I would not be the person I am today. To my wonderful parents, Mr Nghia Do and Mrs Loan Dinh, who have always sacrificed a lot for me and who have always been there to support me. I would like to thank my husband, Bang Nguyen and my two little children, Quynh-An Nguyen and The-An Nguyen for their unwavering support and sacrifices made for me. Thanks also to my uncle Tien Dinh, my brother Thac Do, my sister-in-law Lan Nguyen and my nephew Viet Do for their kind words and unwavering belief in me.

Quynh Ngoc Thuy Do

LIST OF PAPERS/ PUBLICATIONS INCLUDED

Do, Q.N.T., Zhilin, A., Junior, C.Z.P., Wang, G. & Hussain, F.K., 2016, ‘A network-based approach to detect spammer groups’, *2016 International Joint Conference on Neural Networks (IJCNN),* IEEE, pp. 3642-8.

Do, Q.N.T., Hussain, F.K. & Nguyen, B.T., 2017, ‘A fuzzy approach to detect spammer groups’, *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp. 1-6.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

With the widespread use of information technology in business, an increasing number of companies are looking for ways to reduce their overheads. The cost of IT development has been a barrier for both medium and large companies. In order to reduce cost, a new technology called cloud computing is used in many companies without private servers. When selecting which cloud provider to go with in the future, some enterprises will check certain websites for cloud reviews to see what previous cloud consumers thought about the various cloud providers. These kinds of reviews will affect their selection of a cloud provider. Therefore, the reliability of cloud reviews is very important to a cloud consumer so that they can choose a trustworthy cloud provider. In this thesis, four kinds of incorrect reviews in reputation systems are presented, namely ballot stuffing, bad mouthing, spammer groups and cliques. Previous studies on how to identify these types of incorrect reviews are also assessed in this study. Then, new methods to identify incorrect reviews, including ballot stuffing, bad mouthing, spammer groups and cliques are proposed. Finally, the solutions to identify ballot stuffing, bad mouthing and spammer groups are then validated.

THESIS SUMMARY

Reputation systems provide a method for supporting and building trust amongst different parties in online environments. The main concept of a reputation system is to let an agent rate the performance of other agents. We use the term 'agent' loosely here and it may refer to a software agent, web service or a product. A reputation score is then derived based as a mathematical function of ratings on a given agent, which will be used by other agents to determine whether or not to transact with that agent later on. Reputation systems are different from trust referral system, in which agents exchange general recommendations about other agents (Josang et al. 2003; Yolum & Singh 2003).

In reputation systems, agents rate another agent to describe their experience of a particular transaction with that agent. Therefore, the previous ratings are important as they give advice to strangers who want to interact with each other (or with a new agent) for the first time. Reputation systems are effective in providing motivation for honest behaviour and preventing dishonest behaviour amongst the agents (Buchegger & Le Boudec 2003). Finding ways to decrease or avoid unfairly high ratings, unfairly low ratings, spammer groups and cliques (or groups of agents) writing fake reviews is a fundamental issue in reputation systems, when ratings from other agents are being considered. This is because the trusting agents cannot control the reliability of the ratings when these ratings are provided by agents which are out of its control.

To make sure that a reputation system is robust, there is a need for a strong and effective mechanism to protect against unfair ratings. Furthermore, there is a need for intelligent mechanisms or methods to identify spammer groups and cliques giving fake reviews to an agent (Whitby et al. 2004).

The purpose of this research is to develop intelligent methods and algorithms to identify four types of incorrect reviews. For the scope of this research, we focus on four types of incorrect reviews as follows:

- Ballot stuffing: a seller compromises with a number of buyers to ask them to give him an unfairly high rating. In doing so, these buyers inflate the seller's reputation

which can result in increased customers and orders for products at a higher price (Dellarocas 2000).

- Bad mouthing: a seller compromises with a number of buyers to ask them to give unfairly low ratings to its enemy. In doing so, these buyers damage the reputation of the seller's enemy which can result in significantly reduced customers and orders (Dellarocas 2000).
- Spammer group: a group of reviewers who work together to write fake reviews on target products to promote or demote these products (Mukherjee et al. 2011).
- Clique: a group of agents who work together to promote their products. Agents in a clique form a review circle as one agent will receive a review from another agent and will also provide a review on a different agent.

Firstly, this thesis reviews the current literature on ways to identify these kinds of incorrect reviews on reputation systems. Secondly, the research issues, research questions and research objectives are introduced. Then, several new solutions are presented to identify ballot stuffing, bad mouthing, spammer groups and cliques. The solutions to identify ballot stuffing, bad mouthing and spammer groups are then validated, after which the thesis concludes with the research contributions and research plan.

CHAPTER 1:
INTRODUCTION

1.1. Introduction

In this chapter, we present an overview of the importance of identifying incorrect reviews on cloud reputation systems in section 1.2 and several issues related to incorrect reviews are raised in section 1.3.

In section 1.4, we list all the objectives of this particular study. Section 1.5 discusses the scope of the thesis and clearly outlines what is in the scope and what is outside the scope. Section 1.6 presents the significance of the thesis in the context of identifying incorrect reviews on cloud reputation systems. The subsequent section presents a brief introduction of each of the remaining ten chapters that encompass this thesis. The last section concludes the first chapter and sets the scene for the second chapter.

1.2. The importance of identifying incorrect reviews in cloud reputation systems

An incorrect review is considered a type of spam review in which a user gives an untrustworthy and untruthful opinion on products and services for payment in order to mislead other users (Kolhe 2014). These kinds of reviews appear on the retailer's site and other third-party review websites such as www.getapp.com, www.cloudreviews.com, etc. and therefore, can be very damaging. For example, a reviewer who writes a good review on a product for which every other reviewer has given a high rating is not very harmful. However, if a reviewer writes a bad review on a product for which other reviewers have given a good review, this can be very damaging because there is a possibility it will affect the customers' decision as to whether to buy the product (Dellarocas 2000; Duh et al. 2013). As a spammer can prepare a review carefully in order to mislead other customers' buying decisions, detecting these types of reviews by simply manually reading them is difficult and time-consuming, if not impossible (Yeh and Akoglu 2015; Farooq and Khanday 2016). Moreover, reviews that are duplicates or near-duplicates which are written by the same reviewers on different products or by different reviewers on the same or different products are also considered to be incorrect reviews (Jindal and Liu 2008). Incorrect reviews are fake reviews that are intended to mislead readers by giving positive reviews or negative reviews to promote or demote certain targeted products (Dellarocas 2000; Duh et al. 2013). These kinds of reviews can also monitor the expressions of consumers so businesses can change their production and marketing strategies accordingly. Therefore, they will also benefit business organisations (Dellarocas 2000; Duh et al. 2013).

Due to the damage they cause, it is extremely important to identify incorrect reviews on cloud reputation systems so that customers are not misled in their decisions to choose a cloud providers or a cloud product.

In this thesis, we divided the incorrect reviews into four categories: ballot stuffing, bad mouthing, spammer groups and cliques. The goal of ballot stuffing is to sharply increase the popularity and the level of acceptance of products in the market, which could result in more money for endorsement deals (Dellarocas 2000). On the other hand, bad mouthing aims to drive providers out of the market (Dellarocas 2000). This situation has already attracted the attention of the media around the world (Dellarocas 2000). The generation of fake opinions by fake users can translate into increased fame, visibility and significant financial gains for companies and individuals. According to Duh et al. (2013), by just altering the number of followers on social channels, it is possible to change a viewer's deduction about a brand or a product.

Reviews are supposed to reflect genuine user experience and the opinions of consumers. However, incorrect reviews can mislead the sentiment on the target products. It is damaging for both genuine businesses and consumers. Therefore, it is important to detect such fake reviews.

There are some challenges in identifying spamming reviews. According to Ye and Akoglu (2015), opinion spam is widespread, and there are at least two main reasons to explain why the detection of incorrect reviews is a mostly open and challenging issue. These two reasons are as follows:

(1) Humans are not able to identify incorrect reviews based on text, therefore it is extremely difficult to use manual labelling which makes supervised methods inapplicable.

(2) People who write incorrect reviews are often professionals. These fraudulent reviewers are paid by businesses to provide detailed and genuine-looking reviews.

A spammer group is a group of reviewers who work together to write fake reviews to either demote or promote a specific product. Due to the sheer size of a group, this type of incorrect review is even more serious than ballot stuffing and bad mouthing as it takes control of the whole sentiment on a product (Mukherjee et al. 2011).

Forming cliques is a new method for producing incorrect reviews which is identified and discussed in this thesis. The term clique in cloud reputation systems refers to a group of agents who work together to promote their products. Agents in a clique can form a group and provide reviews on each other. An agent will receive reviews from other agents (within the clique) and also will review a different agent. A clique can be damaging because it can affect the decision of the customers as to whether to purchasing various products.

1.3. Issues related to identifying incorrect reviews

Based on our research, there are several issues related to identifying incorrect reviews as follows:

- The software used to detect incorrect reviews is currently very expensive and is unaffordable for many businesses, especially small and medium businesses. Only large organisations and government tend to be able to afford this software (Ilakiya & Felciah 2015).
- In the case of detecting ballot stuffing and bad mouthing, most of the approaches are not able to solve the problem of reviewers providing a large number of ratings within a short period of time, for example, the approach of BRS and the Bayesian network-based model (Zhang et al. 2008).

- In the case of detecting ballot stuffing and bad mouthing, most of the approaches are not capable of detecting ballot stuffing and bad mouthing when the majority of the ratings on a provider are unfair (Zhang et al. 2008).
- Most of the approaches do not take into account the relationship between reviews, reviewers and products. Based on our research, there are different factors to confirm whether a reviewer, a review or a product is suspicious or not. For example, a reviewer is considered to be suspicious if that reviewer has a much higher rating deviation compared to the average rating of a reviewed product, or a reviewer only posts one review for one product, or that reviewer only posts positive or negative reviews, or that reviewer posts too many good or bad reviews for the same product, or that reviewer posts duplicate reviews; a review is considered to be suspicious if that review is a duplicate or near-duplicate review; a product is suspected of receiving incorrect reviews if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. A reviewer is the person who writes a review on a product, so there is a close relationship between the reviewer and the product. Therefore, taking into account the relationship between reviewers, reviews and products is important to identify an incorrect review.
- All the methods to detect spammer groups include two steps: finding all groups of reviews first and then finding the spammer groups later (Duh et al. 2013; Mukherjee et al. 2011), which can be time-consuming because it takes a long to find all the groups which is not necessary, especially when there are a large number of reviews.
- Most of the work on identifying incorrect reviews only investigate travel or restaurant reviews (Jindal & Liu 2008; Li et al. 2011). However, cloud computing has been advancing at an impressive rate in recent years and this is likely to increase in the near future. New services are being developed constantly, such as cloud infrastructure, security and platform as a service, to name a few. Due to the vast pool of available services, review websites have been created to help customers make decisions for their business. Therefore, there is a need to identify incorrect reviews on cloud reputation systems.

- An effective system to build trust between cloud users and cloud providers is a feedback rating-based reputation system. However, it is unfortunate that such a reputation system is missing from the current major cloud providers such as Amazon, Microsoft and Google. Therefore, it is much more difficult for cloud users to select a trusted cloud service from a cloud provider (Qi et al. 2014). To address this challenge, a cloud reputation system, which is a reputation system tailored to cloud services, is proposed. A cloud reputation system is crucial in building trust between cloud users and cloud providers because it not only has the advantages of e-Commerce reputation systems, it also complies with cloud characteristics (Qi et al. 2014).

1.4. Objectives of the thesis

The previous sections outlined the importance of identifying incorrect reviews in cloud reputation systems and the issues related to incorrect reviews. This thesis is an effort to propose solutions to identify incorrect reviews in cloud reputation systems by proposing at least one methodology to identify each type of incorrect review. The objectives of this thesis are summarised as follows:

- To develop an intelligent approach to identify ballot stuffing in cloud reputation systems.
- To develop an intelligent approach to identify bad mouthing in cloud reputation systems.
- To develop an intelligent approach to identify spammer groups in cloud reputation systems.
- To develop an intelligent approach to identify cliques in cloud reputation systems.
- To validate the methods developed to identify ballot stuffing, bad mouthing and spammer groups using data crawled from the website www.getapp.com.

1.5. Scope of the thesis

This thesis presents some intelligent methods to identify four types of incorrect reviews in cloud reputation systems, namely ballot stuffing, bad mouthing, spammer groups and cliques. It should be noted that this thesis focuses only on proposing intelligent methodologies to detect these four types of incorrect reviews so other types of incorrect reviews are not taken into account in this thesis. The dataset used to validate all the methods is crawled from the website www.getapp.com (Alkalbani et al. 2016) which is a website that has many reviews on cloud reputation systems.

1.6. Significance of the thesis

To the best of the researcher's knowledge, at the time of writing, this thesis is the first and only attempt to identify ballot stuffing, bad mouthing, spammer groups and cliques in cloud reputation systems. Specifically, the significance of this thesis arises from the following:

- There is some work on identifying ballot stuffing and bad mouthing, but most does not use the network-based method. The network-based method calculates the suspicion score based on the relationship between reviews, reviewers and products which is proved to be more accurate in this thesis. Moreover, most of the previous methods on identifying ballot stuffing and bad mouthing only investigate travel or restaurant reviews, not reviews related to cloud-based services. This thesis will be the first to develop and apply algorithms to detect incorrect reviews on a cloud dataset.
- There is not much research on detecting spammer groups in reviews, in general. If a method exists, it has been applied to travel and restaurant reviews only, not on reviews related to cloud services. This thesis not only proposes two approaches to identify spammer groups, it also focuses on detecting spammer groups in a cloud dataset. Moreover, previous methods on detecting spammer groups find all the groups first and then identify the spammer groups later.

- This thesis is the first to define the concept of cliques in the review area, which is one type of incorrect review. It also proposes one method to identify cliques in cloud reputation systems which has never been done before.

1.7. Plan of the thesis

In this thesis, we provide some methodologies to identify incorrect reviews in cloud reputation systems. To achieve its objectives, this thesis is organized in ten chapters. A brief summary of each chapter is given in this section:

- Chapter 2: Chapter 2 provides a literature review of the existing literature on intelligent methods to identify incorrect reviews. This chapter also presents the problems associated with the current literature with regard to approaches to identify four types of incorrect reviews. Additionally, all the research problems that are addressed in this thesis are identified in this chapter based on the thorough literature review.
- Chapter 3: Chapter 3 defines all the problems that will be addressed in this thesis. Moreover, all terminologies that we use to define the problems mentioned in this thesis are also presented. Furthermore, different research methodologies are discussed in this chapter in order to select the one that is most suitable for this research.
- Chapter 4: Chapter 4 presents an overview of the solution to each of the issues identified in Chapter 3.
- Chapter 5: Chapter 5 presents an intelligent method to identify ballot stuffing in cloud reputation systems. This is a network-based method comprising four steps using a review graph which consists of three types of nodes, *m* reviewers, *n* reviews and *p* products. Each of these four steps is explained in detail in this chapter.
- Chapter 6: Chapter 6 presents an intelligent method to identify bad mouthing in cloud reputation systems. This is a network-based method to identify bad mouthing comprising four steps using a review graph which consists of three types of nodes,

m reviewers, *n* reviews and *p* products. Each of these four steps is explained in detail in this chapter.

- Chapter 7: Chapter 7 presents two intelligent methods to identify spammer groups in cloud reputation systems. Using the approach to address research question 1 to identify ballot stuffing, we conclude which reviewer has the highest suspicion score; subsequently, we then use the Pearson correlation coefficient, Spearman, K-mean clustering, fuzzy K-means clustering to find the group to which the suspicious reviewer belongs. We use K-means clustering and fuzzy K-means clustering to find the group to which the most suspicious reviewer belongs, then we compare these two methods. The detailed working of the two intelligent methods to identify spammer groups is presented in Chapter 7.
- Chapter 8: Chapter 8 presents an intelligent method to identify cliques in cloud reputation systems. As a clique in cloud reputation systems is first defined in this thesis, there was no existing research on how to detect this type of incorrect review. The method that we propose is a graph method which comprises eight steps and is presented in detail in this chapter.
- Chapter 9: Chapter 9 presents the solution implementation for each of all the methods mentioned in Chapter 5, Chapter 6 and Chapter 7. This chapter also presents the prototypes that are engineered to validate the intelligent methods to identify ballot stuffing, bad mouthing and spammer groups. This chapter also presents the results obtained from the prototypes. Due to the time restriction of this thesis, a solution implementation for the method mentioned in Chapter 8 is not presented. The solution implementation for the method to identify cliques will be presented in future research.
- Chapter 10: Chapter 10 concludes the thesis by giving a summary of the results achieved in this thesis, together with the potential for future work.

1.8. Conclusions

In this chapter, we introduced the importance of identifying incorrect reviews in cloud reputation systems. Then, several issues related to incorrect reviews were pointed out.

Additionally, we discussed the objectives of this study. The scope and significance of this thesis on cloud reputation systems was also presented. Finally, we presented the plan of this thesis.

1.9. References

[1] Alkalbani, A.M., Ghamry, A.M., Hussain, F.K. and Hussain, O.K., 2016, ‘Harvesting Multiple Resources for Software as a Service Offers: A Big Data Study’, In *International Conference on Neural Information Processing*, pp. 61-71, Springer, Cham.

[2] Dellarocas, C., 2000, ‘Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems’, In *Proceedings of the Twenty-First International Conference on Information Systems*, Association for Information Systems, pp. 520-5.

[3] Duh, A., Štiglic, G. & Korošak, D., 2013, ‘Enhancing identification of opinion spammer groups’, In *Proceedings of International Conference on Making Sense of Converging Media*, ACM, p. 326.

[4] Farooq, S. & Khanday, H.A., 2016, ‘Opinion Spam Detection: A Review’, *International Journal of Engineering Research and Development*, vol. 12, no. 4, pp. 1-8.

[5] Ilakiya, K.S. & Felciah, M.M.L.P., 2015, ‘Challenges and techniques for Sentiment Analysis: a survey’, *International Journal of Computer Science and Mobile Computing,* vol. 4, no. 3, pp. 301-7.

[6] Jindal, N. & Liu, B., 2008, ‘Opinion spam and analysis’, In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ACM, pp. 219-30.

[7] Kolhe, N.M., Joshi, M.M., Jadhav, A.B. & Abhang, P.D., 2014, ‘Fake reviewer groups' detection system’, *Journal of Computer Engineering (IOSR-JCE)*, vol. *16*, no. 1, pp. 6-9.

[8] Li, F., Huang, M., Yang, Y. & Zhu, X., 2011, ‘Learning to identify review spam’, In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, p. 2488.

[9] Mukherjee, A., Liu, B., Wang, J., Glance, N. & Jindal, N., 2011, ‘Detecting group review spam’, In *Proceedings of the 20th International Conference Companion on World Wide Web*, ACM, pp. 93-4.

[10] Qi, L., Ni, J., Yan, C., Xia, X. & Ma, C., 2014, ‘Why are Reputation Systems Absent from Cloud Services: Reason and Solution’, *The Sixth International Conferences on Advanced Service Computing*, pp. 9-14.

[11] Ye, J. & Akoglu, L., 2015, ‘Discovering opinion spammer groups by network footprints’, In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Cham, pp. 267-82.

[12] Zhang, J., Sensoy, M. & Cohen, R., 2008, ‘A detailed comparison of probabilistic approaches for coping with unfair ratings in trust and reputation systems’, In *Privacy, Security and Trust, 2008. PST'08. Sixth Annual Conference on,* IEEE, pp. 189-200.

CHAPTER 2:
LITERATURE REVIEW

2.1. Introduction

In this chapter, we present an overview of the existing literature on intelligent methods to identify incorrect reviews.

In section 2.2, we list the existing definitions of the four types of incorrect reviews which are ballot stuffing, bad mouthing, spammer groups and cliques. This section also lists the definition of cloud reputation systems.

In section 2.3, we present some previous approaches to detect ballot stuffing and bad mouthing. These approaches can be categorized into two types, internal-based and external-based approaches.

Section 2.4 lists a few approaches to identify spammer groups. These approaches fall into three types, linguistic-based, behaviour-based and network-based approaches.

In section 2.5, we identify several approaches to detect cliques in general, however this is not related to reviews on cloud reputation systems as this is the first time, to the best of the researcher's knowledge, that cliques have been investigated in relation to cloud reputation systems.

In section 2.6, the shortcomings of the existing approaches to identify the four types of incorrect reviews are pointed out. We also carried out a thorough and critical evaluation of the existing approaches.

Finally, section 2.7 concludes the chapter.

2.2. Definitions of incorrect reviews in the literature

2.2.1. Definition of ballot stuffing

When a seller compromises with a number of buyers to ask them for an unfairly high rating, this is called ballot stuffing. In doing so, these buyers inflate the reputation of the seller which could result in an increased number of customers who order products at a higher price (Dellarocas 2000).

2.2.2. Definition of bad mouthing

When a seller compromises with a number of buyers to ask them to give an unfairly low rating to its enemy, this is called bad mouthing. In doing so, these buyers deflate the reputation of the seller's enemy and which could result in a reduced number of customers ordering products (Dellarocas 2000).

2.2.3. Definition of spammer groups

A spammer group is a group of reviewers who work together to write fake reviews on target products in order to promote or discredit these products (Mukherjee et al. 2011). spammer groups are very damaging as there are a large number of people in a group. Some spammer groups can even control a product's reputation. Due to the harmful nature of spammer groups, several studies have investigated ways to identify such groups in reputation systems.

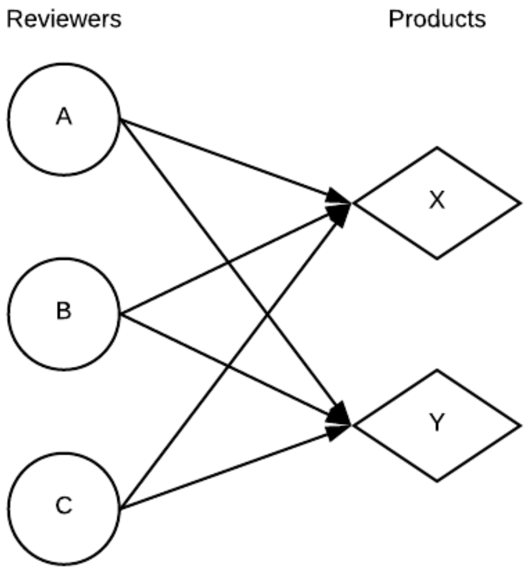


Figure 1: Pictorial representation of a spammer group

Figure 1 shows an example of a spammer group. In this figure, reviewer A, reviewer B and reviewer C form a spammer group and together they write good or bad reviews on product X and product Y.

2.2.4. Definition of cliques

As mentioned in Chapter 1, there is no definition of a clique in the area of reviews. However, there is a definition of a clique in general. In general, a clique is a group of people who interact with each other more regularly and intensely than others in the same settings (Salkind 2008).

In the context of this thesis, we regard a clique as a group of agents who work together to promote their products. Agents in a clique form a review circle as one agent will receive a review from another agent and will also provide a review on a different agent.

Figure 2 shows an example of a clique. A group consists of three agents, agent X, agent Y and agent Z. These three agents will form a clique if agent X writes good reviews on agent Y's products, agent Y writes good reviews to promote agent Z's products and agent Z does the same thing for agent X's products.

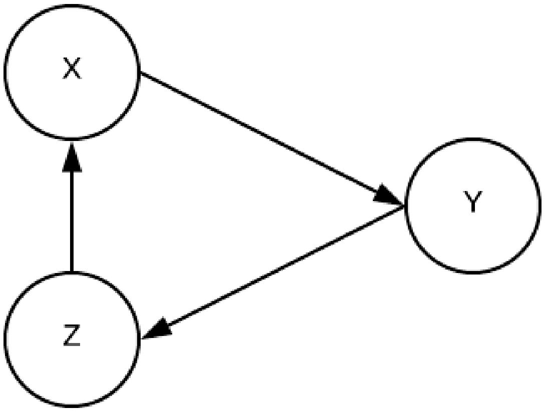
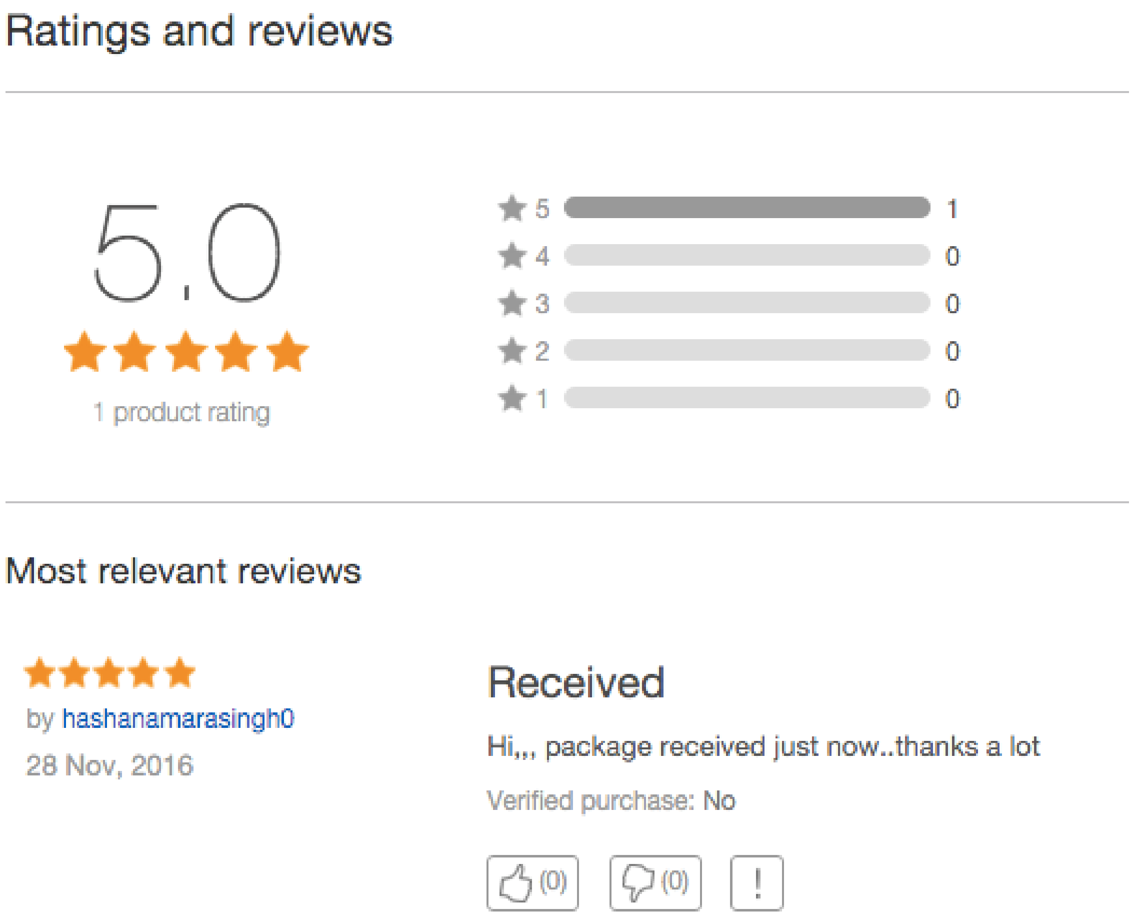


Figure 2: Pictorial representation of a clique

2.3. Approaches to identify ballot stuffing and bad mouthing

The methods for identifying unfair ratings in reputation systems can be categorized according to internal factors and external factors of the reviews. By internal factors, we mean that the methods for identifying the unfair ratings consider the rating values themselves, not any attributes or features of the reviews (Whitby et al. 2004). In this case, only statistical properties are used for recognizing potentially unfair ratings. By external factors, we mean that the methods take into account not only the internal factors, but also other elements such as the reputation of the raters (Whitby et al. 2004). Consider eBay as an example. When internal factors are used to identify unfair ratings on eBay, only the 1 to 5-star ratings are taken into account. Figure 3 shows an example of an internal factor used to identify incorrect reviews. When external factors are used, other factors are also considered, such as the reputation of the raters. Figure 4 illustrates some external factors that can be used to counter incorrect reviews, such as the reputation of the raters (100% positive feedback is illustrated in this example). Other factors that could be taken into account are: the number of ratings given by a rater, the number of positive ratings, the number of negative ratings, and the number of neutral reviews.



Ratings and reviews

5.0

★★★★★

1 product rating

★ 5 1

★ 4 0

★ 3 0

★ 2 0

★ 1 0

Most relevant reviews

★★★★★

by hashanamarasingh0

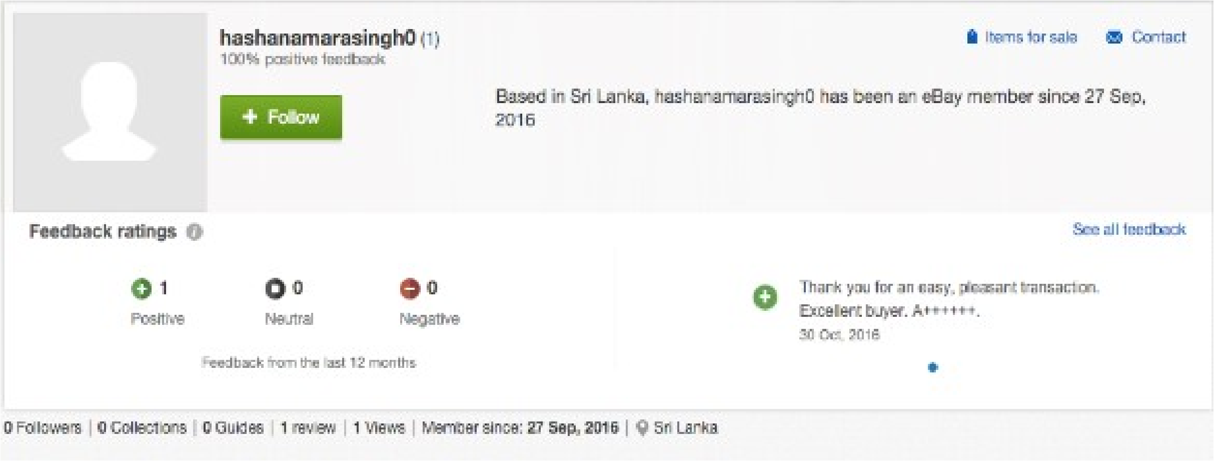
28 Nov, 2016

Received

Hi,,, package received just now..thanks a lot

Verified purchase: No

Figure 3: Example of internal factors



hashanamarasingh0 (1)

100% positive feedback

Based in Sri Lanka, hashanamarasingh0 has been an eBay member since 27 Sep, 2016

Feedback ratings

1 Positive

0 Neutral

0 Negative

Feedback from the last 12 months

Thank you for an easy, pleasant transaction. Excellent buyer. A++++++.

30 Oct, 2016

0 Followers | 0 Collections | 0 Guides | 1 review | 1 Views | Member since: 27 Sep, 2016 | Sri Lanka

Figure 4: Example of external factors

2.3.1. Internal factor-based methods to identify ballot stuffing and bad mouthing

In this section, a selection of previous work identifying ballot stuffing and bad mouthing that use internal factors is discussed. By internal factor-based methods, we mean that the methods are only based on analyzing and comparing the ratings values themselves, not any other values.

A method using internal factors to detect unfair ratings was presented by Dellarocas (2000). To avoid unfairly low ratings and negative discrimination, a mechanism of controlled anonymity is introduced. In this case, the buyer and the seller do not know the identity of the other, they only know their reputation. However, hiding identities is not effective in all domains, especially for hotel and restaurant ratings where concealing identities is impossible. Nevertheless, this method works well on eBay and other customer-to-customer auction websites. To decrease the effect of unfairly high ratings and positive discrimination, first a technique called collaborative filtering is used to identify the nearest neighbors of a buyer agent, according to similar preferences with other buyer agents of commonly rated seller agents. Then, a cluster filtering technique is used to exclude unfairly high ratings provided by those neighbors. The neighbors' ratings may be separated into two clusters, one with the lower ratings, and one with the higher ratings. The lower-rated clusters include all the fair ratings, while the higher-rated clusters include all the unfairly high ratings. Therefore, the ratings in the higher-rated clusters are filtered out. This approach only takes into account the ratings in the most recent time window whose width is affected by the fair ratings' frequency. This cluster filtering technique deals with unfairly high ratings; considers the preference similarity between buyer agents and advisor agents; and copes with agents' ratings changes. In conclusion, Dellarocas (2000) applies different methods to detect ballot stuffing and bad mouthing. Using different methods to detect ballot stuffing and bad mouthing can be complicated and time consuming, given more action needs to be taken to detect both. For example, in the case of the method proposed by Dellarocas (2000), one needs to hide the identity of the users to avoid unfairly low ratings and apply a cluster filtering to deal with unfairly high ratings. These authors mention that their method works well for avoiding bad mouthing and ballot stuffing on eBay and other customer-to-customer websites; however, their work has not been applied to cloud provider reviews.

Mukherjee et al. (2011) propose an iterative filtering approach to exclude unfair feedback provided by advisor agents. In particular, the reviews provided by each advisor agent include ratings that support both good and bad reputations for a seller agent but are defined as a beta distribution. The feedback is examined as fair feedback if the seller agent's accumulated reputation is between the lower and upper boundaries of the feedback. This technique is efficient if the majority of ratings are fair. The authors do not propose any method to accommodate bad mouthing. Similar to Dellarocas (2000), their work has not been applied to cloud provider reviews.

A novel spam detection method to identify ballot stuffing reviews was proposed in Fayazbaksh and Sinha (2012). Their work focuses on network structures, which consist of reviewers, reviews, and products. A review graph is built with three types of nodes: reviewers, reviews and, products. Their proposed method consists of two stages. In the first stage, a suspicion score is calculated for each node in the review graph in three different ways. The second stage comprises a forward update and a backwards update. For example, a product's suspicion score is increased if it primarily receives reviews from suspicious reviewers. This approach has only been evaluated on restaurant and hotel reviews; their work has not been applied to cloud provider reviews.

2.3.2. External-based methods to identify ballot stuffing and bad mouthing

In this section, we discuss external factor-based methods to identify ballot stuffing and bad mouthing. By external factors, we mean that in addition to the internal factors, other elements, such as the reputation of the raters, will be used to determine the weight of the ratings.

A general method for ballot stuffing developed by Chen and Singh (2001) is 'global match - global confidence'. This is one of the methods that uses external factors in identifying ballot stuffing and bad mouthing. All the ratings given to each product are obtained, and then automatically used to compute reputations for raters. Their proposed method can be illustrated in three steps. First, the quality and confidence values of each of the raters' ratings for each object in a category are computed. The frequency distribution of all ratings given to each product are used to determine the quality value, also called a local match (LM). A piecewise function is used to calculate the confidence level, called local confidence (LC).

Second, the accumulated quality and confidence values of all the ratings for each category of objects are calculated by incorporating the LM and the LC for each object in the category. The accumulated quality is called the global match (GM), and the confidence values are called the global confidence (GC). The last step is to compute the rater's reputation using the GM and the GC of the raters for each category. Raters that are less reputable will give ratings that affect the accumulated reputations of seller agents less. The proposed method only works for ballot stuffing. The authors demonstrate how their proposed algorithm can be applied to e-commerce and auction websites, like eBay. The shortcoming of this method is that they have not applied it to cloud service reviews.

A probabilistic model was applied to a beta reputation system proposed by Jøsang and Ismail (2002) to evaluate the reputations of seller agents. This reputation engine is based on a beta probability density function. This model considers the rating for an advisor agent as either a good reputation or a bad reputation, which can be examined as binary events in the beta probability distribution. The amount of both good and bad reputation ratings from multiple advisor agents is accumulated and then used to determine the reputation of a seller agent. The proposed approach deals with the temporal aspect of ratings and is effective when the unfair ratings are generated by changing the situation, but not intentional change and intuitive differences. This method works for both ballot stuffing and bad mouthing. The study is suitable for supporting electronic contracts and building trust between players in e-commerce, due to its sound theoretical basis in statistics, its flexibility, and its simplicity. However, similar to the method proposed by Chen and Singh (2001), the applicability of this method to cloud reviews has not been investigated.

A robust reputation system for mobile ad-hoc networks (RRSMAN) was proposed by Buchegger and Le Boudec (2003). In their proposed method, each node in the network in this fully distributed system manages a reputation rating and a trust rating of the other nodes. The trust rating for a node indicates how likely that node is to give true advice. The reputation rating for a node indicates how accurately it engages with the node holding the rating. Based on evidence collected previously, both the reputation rating and the trust rating, which is held by node *i* for node *j*, are updated using a modified Bayesian function. The weight of evidence is based on its order of being gathered. The reputation rating that node *i* holds for node *j* is updated in agreement with node *k*'s advice only if node *k* is truthful or the advice is consistent with the reputation held by node *i*. The advice is recognized as being consistent if the difference between itself and the reputation held by node *i* is not equal to or more than a fixed deviation threshold. This approach is effective in dealing with falsely disseminated information. However, there are some disadvantages in this study. First, as previously mentioned, the evidence's weight is pursuant to the order in which they are collected, not according to the exact time that the evidence is gathered. Therefore, there is no difference in the weight of the evidence that was collected one day ago versus one month ago. Second, this approach only takes into account the current reputation ratings of one node against another node to establish the preference similarity between two nodes. The third problem is related to the way RRSMAN integrates advice. All the advice from other nodes is given equal weight when these nodes are truthful, or each piece of advice is consistent. Furthermore, this approach has not been applied to cloud service reviews.

TRAVOS, proposed by Teacy et al. (2006), is a probabilistic model for agent-based virtual organizations. This model deals with inaccurate reputation advice by performing two functions. The first function evaluates the exactness of the current reputation advice depending on the number of correct and incorrect advice in the past, which is related to the current advice. The second function changes the reputation advice in accordance with its exactness. This function can help decrease inexact advice. It can also help reduce untruthful ratings that an advisor agent gives a seller agent on multiple occasions. However, the shortcoming is that the study assumes that a seller agents' action does not change over time, which is not always true. Moreover, this model has not been applied to cloud service reviews.

Liu et al. (2014) proposed an integrated CLUStering-Based approach (iCLUB) to effectively filter incorrect reviews. This approach adopts clustering techniques and integrates two components which are local and global. By local component, we mean that only buyers' knowledge about the sellers are currently evaluated (target sellers). By global component, we mean that the knowledge of the buyers about the other sellers that the buyers have previously received is also taken into account. The global component is very useful especially when the buyers do not have much experience with the target sellers. This approach does not work when the majority of the reviews are incorrect (Liu et al. 2014; Jiang et al. 2013). Furthermore, the applicability of this approach to cloud reviews has not been investigated.

2.4. Approaches to identify spammer groups

Techniques that have already been employed to detect single spammers, or groups of them, can be divided into three different categories: linguistic-based methods, behaviour-based methods and network-based methods.

We regard linguistic-based methods as the methods which use the textual content of reviews to identify incorrect reviews.

We regard behaviour-based methods as methods that analyse reviewers' regular patterns to determine if they are writing incorrect reviews.

We regard network-based methods as methods that use a review network which is a graph consisting of reviewers and products, or reviews, reviewers and products to identify spammer groups.

2.4.1. Linguistic-based methods to identify spammer groups

We define *linguistic-based methods* as methods that use the textual content of reviews to identify spam. This method was the first to identify product review spamming behaviour.

Jindal and Liu (2010) categorize spam into three types: fake reviews, reviews on brand only, and non-reviews. Their method first detects duplicate reviews, then supervised learning with manually labeled training examples is used to identify the reviews on brand only and non-reviews. Another method is used to detect fake reviews, as it is not easy to manually label training examples for this type of review. Duplicate spam reviews are used as positive training examples; the other reviews are used as negative examples. This method mainly relies on text similarity.

Duh et al. (2013) asserted that fake reviewers are of concern to consumer-generated media and proposed an approach to enhance the detection of spammer groups. Mukherjee et al. (2012) indicated that spammer groups could be detected using very simple linguistic features, such as word and part-of-speech n-gram features. However, Duh et al. (2013) considered

Mukherjee et al. (2012)'s proposed linguistic features to be inefficient and responded with some additional techniques: linguistic inquiry and word counts. The enhancements proposed by Duh et al. (2013) could yield better results for Mukherjee et al. (2012)'s algorithm. The shortcoming of this method is that they have not applied it to cloud service reviews.

Another work by Ott et al. (2011) confirms that the combination of simple linguistic features, like n-grams and psycho-linguistics guarantees better results than n-grams alone. Psycho-linguistic features can be more important than the similarities between group members and their content. The drawback of this technique is that it is only good for one type of spamming activity, and even the most sophisticated algorithms can be easily misled by groups of spammers that write many similar opinions. Such tactics make recently written posts look regular (Duh et al. 2013). Similar to Mukherjee et al. (2012), their work has not been applied to cloud provider reviews.

2.4.2. Behaviour-based methods to identify spammer groups

We regard *behaviour-based methods* for identifying spammer groups as the methods that analyse reviewers' regular patterns to infer if they are writing incorrect reviews.

Mukherjee et al. (2011) proposed a method to detect spammer groups based on their patterns. First, possible candidate groups are selected using data mining to identify their patterns. They are then grouped together, and several indices are calculated for each group. These indices include similarity of their content, group deviation, group size and time frame. Using these indicators, the groups are ranked according to their likelihood of being a spammer group. SVM rank is used as a tool to help the team rank these candidates. The technique was evaluated through experiments. A dataset was extracted from Amazon.com using a data mining method. 2273 potential spammer group candidates were detected, with a minimum of 2 members per group. The candidates were then ranked using the SVM ranking methodology. To evaluate the correctness of the ranking, three human judges analysed the results. Each judge was asked to determine whether any candidate in the top 100, middle 100 and bottom 100 groups was a real spammer group. The results show that the algorithm's efficiency is promising. Although the three judges worked individually, their decisions show high agreement that, in the top 100, almost all of the candidates were real spammer groups. However, due to the nature of judging, the middle 100 candidates were difficult to assess, so the judges' results show some variance. However, their work has not been applied to cloud provider reviews.

In 2012, Mukherjee et al. enhanced their previously proposed method to detect fake reviewer groups. Ratings and reviews from previous buyers are crucial to customers' decisions, hence, some provider cooperatives create fake reviews about their rivals' products or their own products to influence users' perceptions, an action which is called *opinion spamming*. Various research has been undertaken to detect fake reviews by analysing each comment individually. However, studies about spamming groups are limited. It is difficult to state if a single review is spam or not by just reading the comment. However, spammer groups earn money by writing reviews for multiple products. This pattern allows our algorithm to detect whether a reviewer belongs to a spammer group. To validate the accuracy of the algorithm, a dataset of 53,469 reviewers with 109,518 reviews about 39,392 products was selected from Amazon.com for testing (Jindal & Liu 2008). The original dataset was generated in 2006 and was updated in early 2010. It was also used by Lim et al. (2010) and Jindal et al. (2010). The data was analysed and grouped together using pattern mining. Each possible spammer group had at least two members and a minimum of three supporters. Human judges were employed to identify which candidates were a real spammer group; they also analysed and ranked these candidates, with the higher ranked candidates more likely to be spammer groups. A few indicators and a useful ranking algorithm were used to effectively rank each group. The team calculated the various indices from the dataset, which allowed them to build a relation model between the groups, the groups' members, and the target products using the GSMrank algorithm as a base. Other than GSMrank, several other techniques were used in the research to show the efficiency of GSMrank, and the results were compared. The final results show that GSMrank scored a 95% confidence level. Similar to the method proposed by Mukherjee et al. (2011), the applicability of this method to cloud reviews has not been investigated.

2.4.3. Network-based methods to identify spammer groups

We regard *network-based methods* as methods that use a review network which is a graph consisting of reviewers and products, or reviews, reviewers and products to identify spammer groups.

Ye and Akoglu (2015) also proposed a method to detect spammers. Their method focuses on detecting spammer groups using network footprints without relying on ancilliary information, such as review content or user behaviour. One of the key factors that affects the decision of buyers is the online reviews from past customers that show real experience with the product or service. For this reason, fake reviews create huge issues for consumers. Several studies have been conducted to solve this problem. Most focus on detecting whether a single review or comment is a spamming opinion. However, the act of spamming is usually a result of a campaign to promote or destroy a particular product's popularity (Mukherjee et al. 2011; Mukherjee et al. 2012). There is little work on how to detect a spammer group; furthermore, those studies attempting to identify spamming groups rely on behaviours or a linguistic methodology to decide whether a group is a spamming group. These approaches are considered non-robust, as spammers can adjust actions and languages to hide their intent. A new technique has been proposed that uses a reviewers' network footprint to detect their real intent. This method includes two features: a network footprint score (NFS) and GroupStrainer. NFS is the indicator that quantifies the effect of spamming activities on a product and includes two factors: neighbor diversity and self-similarity. Neighbor diversity indicates the variety of behaviours and activities within a set of reviewers; self-similarity measures the patterns in comments. By combining these two factors, we can measure the NFS of products. However, NFS is only accurate when the reviewer’s dataset is large enough (more than 20). After calculating the NFS feature of products, the next step is to group candidates to detect spamming groups. Locality-sensitive hashing is used to merge all similar data nodes together. To ensure the effectiveness of the approach, synthetic and real-life datasets were used to test the proposed technique. To evaluate NFS using the synthetic datasets, spam reviews were injected into the data at different rates, 10% and 30%, for both popular and random products. Other methods, such as Oddball, CatchSync and FraudEagle were used as baseline comparisons. The results show that FraudEagle combined with the NFS indicator almost perfectly detected spammer groups in the dataset. In terms of testing the performance GroupStrainer using real-life datasets, 20 spammer groups were injected into real-life datasets for randomly selected products. The results reveal that GroupStrainer produced high quality results for grouping spammers, even with high levels of camouflage. However, their work has not been applied to cloud service reviews.

A novel spam review detection method was proposed by Fayazbakhsh and Sinha (2012), based on the previous work of Wang et al. (2012). The difference between these two models is that the algorithm presented by Wang et al. (2012) uses minimal side information. Fayazbakhsh and Sinha (2012)'s proposed method consists of two stages. In the first stage, a suspicion score is calculated for each node in the review graph. A review graph has three types of nodes: reviewer, review and product. In the second stage, the calculated suspicion scores are iteratively updated via forward and backward updates. Similar to the method proposed by Ye and Akoglu (2015), the applicability of this method to cloud reviews has not been investigated.

2.5. Approaches to identify cliques

There is currently no research on identifying cliques in reputation systems in general and cloud reputation systems in particular. However, there are some studies on identifying cliques in a weighted directed graph which can be used in this case.

There is a wide variety of community or graph recognition and detection algorithms that have been developed in the last few years which vary in types of networks such as weighted or unweighted, or the community they can handle in addition to the techniques used. Weighted subgraphs are graphs where some nodes or edges are considered to be more significant than others.

A graph is defined as a collection of nodes (n) and edges (m) connecting these nodes. A clique of a graph G is defined as a complete subgraph of G and is usually maximum which refers to the largest possible clique size that can be detected. The problem of a clique is the cornerstone of finding particular complete subgraphs in a graph, which is referred to as the set of elements where each pair of elements is connected (Regineri 2007). However, the maximal complete subgraphs are not included in any other complete subgraphs.

For example, in the social networking approach, a clique is a subset of a network where each element in that network is more related and tied to each other than to other elements outside of the subset (Jamali & Abolhassani 2006). In terms of friendship ties, for example, it is very common for people to form "cliques" on many different bases such as age, ethnicity and others. The smallest "clique" is formed from only two objects which can be extended to become wider by forming strong or closely connected regions in a graph. N-P complete is an algorithm to find the existence of maximum clique size in a graph (Vassilevska 2009).

Another approach is the densest subgraph which relies on some objective function in order to identify a subgraph optimal density. Though some approximation detection model has been used to detect the highest clique density (Günnemann & Seidl 2010; Liu & Wong 2008), it is still a difficult problem because these algorithms generate a huge output. Plus, according to their definition, there is no size restriction on the output of quasi-cliques regarding only undirected and unweighted graphs.

Various partitioning graph methods have been used to divide the graph into smaller subgraphs or splitting up the graph by overlapping subgraphs (Günnemann & Seidl 2010). In this approach, each partition of the graph represents a complete hierarchy of the interesting subgraph which possibly adds a high runtime demand. Discovering interesting subgraphs with different strengths is the main issue.

For example, online commercial system finding such interesting subgraphs is based on the classification and clustering of graph data which considers that each graph has connections, directions and also weights between objects which are an important factor to identify the dense subgraphs. The dense finding can be important and useful in the online commercial system that might be used to target customer delivery services or any other services. Therefore, finding dense subgraphs by optimizing the directed and weighted graphs is an important approach.

Günnemann and Seidl (2010) proposed an algorithm called GDens, or Graph Density as an effective solution for high quality in clustering and runtime performance. Based on the GDens algorithm, they proposed a clique filter model to identify clustered subgraphs in a cloud review application dataset community. This research focuses on identifying clique reviews using the GDens algorithm.

2.6. Critical evaluation of existing approaches: an integrative view

2.6.1. Ballot stuffing and bad mouthing

In this part, we evaluate the existing approaches by checking their capabilities when detecting ballot stuffing and bad mouthing. These capabilities are majority, burstiness and relationship.

By majority, we mean that an approach is able to identify ballot stuffing and bad mouthing even when the majority of the ratings of a provider is unfair (Zhang et al. 2008). As mentioned previously, whist there are existing methods that can detect ballot stuffing and bad mouthing such as method proposed by Mukherjee et al. (2011), they work under the assumption that the majority of the ratings in the system are fair.

By burstiness, we mean that an approach should be able to deal with the situation when a large number of ratings are provided by reviewers within a short period of time (Zhang et al. 2008).

By relationship, we mean that an approach should be able to take into account the relationship between the reviewers, reviews and products. The calculation of the suspicion score based on the relationship between the reviewers, reviews and products will have a positive effect on the accuracy of an algorithm. This is verified in Chapter 9 of this thesis.

	Majority	Burstiness	Relationship
BRS (Jøsang & Ismail 2002)	No	No	No
TRAVOS (Teacy et al. 2006)	Yes	Yes	No
iCLUB (Liu et al. 2014; Jiang et al. 2013)	No	Yes	No
Our method	Yes	Yes	Yes

Table 1: Capabilities of existing approaches compared to our method

Table 1 lists a few of the existing research studies with their capabilities. BRS (Jøsang & Ismail 2002) does not have any of the three capabilities listed above. TRAVOS (Teacy et al. 2006) can deal with a situation when the majority of the reviews are unfair and when reviewers provide a large number of ratings within a short period of time. However, TRAVOS does not have a relationship capability. This means TRAVOS will not be as accurate as our method, which is verified in Chapter 9 of this thesis. iCLUB (Liu et al. 2014, Jiang et al. 2013) does not have majority or relationship capabilities, however, it works well with a burstiness capability.

2.6.2. Spammer groups

All of the previous studies focus on finding candidate groups first using frequent item set mining, then later checking if they are spammer groups as they found that labelling individual fake reviews or reviewers is harder than labelling groups. A method that finds out which one is spam and then checks if they are in a group can help save time by making use of the method to identify ballot stuffing and bad mouthing.

2.6.3. Cliques

Cliques can appear quite a lot in cloud reviews, but there is no current work on identifying cliques in cloud-based reviews, nor is there any research on detecting cliques using a weighted directed graph.

2.6.4. Shortcomings of the existing literature on identifying incorrect reviews

Based on a thorough review of the existing literature, we identify the following gaps or shortcomings in the literature:

- The software used in detecting incorrect reviews is currently very expensive and is unaffordable for all businesses, especially small and medium businesses. Only large organisations and government can afford to benefit from it (Ilakiya & Felciah 2015).
- In the case of detecting ballot stuffing and bad mouthing, most of the approaches are not able to solve the problem of reviewers providing a large number of ratings within a short period of time, for example, the approach of BRS and the Bayesian network-based model (Zhang et al. 2008).

- In the case of detecting ballot stuffing and bad mouthing, most of the approaches are not capable of detecting ballot stuffing and bad mouthing when the majority of the ratings of a provider are unfair (Zhang et al. 2008).
- Most of the approaches do not take into account the relationship between reviews, reviewers and products. Based on our research, there are different factors to confirm whether a reviewer, a review or a product is suspicious or not. For example, a reviewer is considered to be suspicious if that reviewer provides a much higher rating deviation compared to the average rating of a reviewed product, or a reviewer only posts one review for one product, or that reviewer only posts positive or negative reviews, or that reviewer posts too many good or bad reviews for the same product, or that reviewer posts duplicate reviews; a review is considered to be suspicious if that review is a duplicate or near-duplicate review; a product is suspected of getting review spams if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Moreover, a reviewer is a person who writes a review on a product, so there is a close relationship between them. Therefore, taking into account the relationship between reviewers, reviews and products is important to identify an incorrect review.
- All the methods to detect spammer groups include two steps which are finding all groups of reviews first and then finding the spammer groups later (Duh et al. 2013; Mukherjee et al. 2011), which can be time-consuming because it takes a long time to find all the groups which is not necessary, especially when there is a large number of reviews.
- Most of the work on identifying incorrect reviews only work on travel or restaurant reviews (Jindal & Liu 2008; Li et al. 2011). However, cloud computing has been advancing at an impressive rate in recent years and is likely to increase in the near future. New services are being developed constantly, such as cloud infrastructure, security and platform as a service, to name just a few. Due to the vast pool of available services, review websites have been created to help customers make decisions for their business. Therefore, there is a need to identify incorrect reviews on cloud reputation systems.

- There are only few research studies on identifying spammer groups. These studies find candidate groups first and then check whether these groups are spammer groups later. Therefore, there is no study that checks for spam reviews first and then checks if these reviews are in a group or not which can help make use of methods to identify bad mouthing and ballot stuffing.
- There is no definition of a clique in a cloud reputation system in the previous research. Also, there is no method to identify cliques in cloud-based reputation systems.
- There is no integration algorithm to identify the four types of incorrect reviews discussed in this thesis which are ballot stuffing, bad mouthing, spammer groups and cliques.
- An effective system to build trust between cloud users and cloud providers is a feedback rating-based reputation system. However, it is unfortunate that such a reputation system is missing from the current major cloud providers such as Amazon, Microsoft and Google. Therefore, it is much more difficult for cloud users to select a trusted cloud service from a cloud provider (Qi et al. 2014). In consideration of this challenge, a cloud reputation system, which is a reputation system tailored to cloud services, is proposed. A cloud reputation system is crucial in building trust between cloud users and cloud providers because it not only has the advantages of e-Commerce reputation systems but also complies with cloud characteristics (Qi et al. 2014).

2.7. Conclusions

In this chapter, we undertook an extensive survey of the existing approaches on identifying ballot stuffing, bad mouthing, spammer groups and cliques. We categorized the existing literature into various types of approaches. Finally, we discussed these approaches and their shortcomings.

In the next chapter, we define the problems that we intend to address in this thesis.

2.8. References

[1] Buchegger, S. & Le Boudec, J.Y., 2003, *A robust reputation system for mobile ad-hoc networks*, No. LCA-REPORT-2003-006.

[2] Chen, M. & Singh, J.P., 2001, 'Computing and using reputations for internet ratings', In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, ACM, pp. 154-62.

[3] Dellarocas, C., 2000, 'Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior', In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, ACM, pp. 150-57.

[4] Duh, A., Štiglic, G. & Korošak, D., 2013, 'Enhancing identification of opinion spammer groups', In *Proceedings of the International Conference on Making Sense of Converging Media*, ACM, p. 326.

[5] Fayazbakhsh, S.K. & Sinha, J., 2012, Review spam detection: a network-based approach. *Final Project Report: CSE*, *590*.

[6] Günnemann, S. & Seidl, T., 2010, 'Subgraph mining on directed and weighted graphs', In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, pp. 133-46.

[7] Ilakiya, K.S. & Felciah, M.M.L.P., 2015, 'Challenges and techniques for Sentiment Analysis: a survey', *IJCSMC March*.

[8] Jamali, M. & Abolhassani, H. 2006, 'Different aspects of social network analysis', *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 66-72.

[9] Jiang, S., Zhang, J. & Ong, Y.S., 2013, 'An evolutionary model for constructing robust trust networks', In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, pp. 813-20.

[10] Jindal, N. & Liu, B., 2008, 'Opinion spam and analysis', In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ACM, pp. 219-30.

[11] Jindal, N., Liu, B. & Lim, E.P., 2010, ‘Finding unusual review patterns using unexpected rules’, In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ACM, pp. 1549-52.

[12] Josang, A. & Ismail, R., 2002, ‘The beta reputation system’, In *Proceedings of the 15th Bled Electronic Commerce Conference*, vol. 5, pp. 2502-11.

[13] Li, F., Huang, M., Yang, Y. & Zhu, X., 2011, ‘Learning to identify review spam’, In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, p. 2488.

[14] Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B. & Lauw, H.W., 2010, ‘Detecting product review spammers using rating behaviors’, In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ACM, pp. 939-48.

[15] Liu, G. & Wong, L. 2008, ‘Effective pruning techniques for mining quasi-cliques’, In:ECML/PKDD, vol. 2, pp. 33–49.

[16] Liu, S., Zhang, J., Miao, C., Theng, Y.L. & Kot, A.C., 2014, ‘An integrated clustering-based approach to filtering unfair multi-nominal testimonies’, *Computational Intelligence*, vol. 30, no. 2, pp. 316-41.

[17] Mukherjee, A., Liu, B. & Glance, N., 2012, ‘Spotting fake reviewer groups in consumer reviews’, In *Proceedings of the 21st International Conference on World Wide Web,* ACM, pp. 191-200.

[18] Mukherjee, A., Liu, B., Wang, J., Glance, N. & Jindal, N., 2011, ‘Detecting group review spam’, In *Proceedings of the 20th International Conference Companion on World wide web*, ACM, pp. 93-4.

[19] Ott, M., Choi, Y., Cardie, C. & Hancock, J.T., 2011, ‘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,* Association for Computational Linguistics, vol. 1, pp. 309-19.

[20] Regineri, M. 2007, ‘Finding All Cliques of an Undirected Graph’, “Current Trends in IE" WS.

[21] Teacy, W.L., Patel, J., Jennings, N.R. & Luck, M., 2006, ‘Travos: Trust and reputation in the context of inaccurate information sources’, *Autonomous Agents and Multi-Agent Systems*, vol. 12, no. 2, pp.183-98.

[22] Vassilevska, V. 2009, ‘Efficient algorithms for clique problems’, *Information Processing Letters*, vol. 109, no. 4, pp. 254-57.

[23] Wang, G., Xie, S., Liu, B. & Yu, P.S., 2012, ‘Identify online store review spammers via social review graph’, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 61.

[24] Whitby, A., Jøsang, A. & Indulska, J., 2004, ‘Filtering out unfair ratings in bayesian reputation systems’, In *Proc. 7th Int. Workshop on Trust in Agent Societies*, vol. 6, pp. 106-17.

[25] Ye, J. & Akoglu, L., 2015, ‘Discovering opinion spammer groups by network footprints’, In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Cham, pp. 267-82.

[26] Zhang, J., Sensoy, M. & Cohen, R., 2008, ‘A detailed comparison of probabilistic approaches for coping with unfair ratings in trust and reputation systems’, In *Privacy, Security and Trust, 2008. PST'08. Sixth Annual Conference on,* IEEE, pp. 189-200.

CHAPTER 3:
PROBLEM DEFINITION

3.1. Introduction

The first chapter highlighted the importance of identifying incorrect reviews in cloud reputation systems. In the second chapter, we presented a review of the existing literature. As discussed in earlier chapters, it is very important to identify incorrect reviews in cloud reputation systems. As cloud computing is become increasingly popular nowadays, a growing number of people are using cloud resources. Some of the website portals where cloud reviews are available are www.getapp.com, www.cloudreviews.com, www.hostingadvice.com, etc. Before moving things to the cloud, consumers have to choose the one which is most appropriate for them. The average consumer will simply read the reviews of the previous consumers and choose a cloud provider based on these reviews. Therefore, detecting incorrect reviews in cloud reputation systems becomes an important problem that needs to be solved. There are some research studies which work on identifying incorrect reviews on restaurants and hotels which are mentioned in Chapter 2. Based on the thorough literature review presented and documented in Chapter 2, there are still many gaps in the existing literature in the identification of incorrect reviews in cloud reputation systems. The shortcomings in the existing literature are described in section 2.6.4.

In this chapter, we focus on formally defining the research problem being addressed in this thesis. In section 3.2, we define the key terms and concepts which are used to formally define a problem in this thesis. Section 3.3 defines the problem that is addressed in this thesis and section 3.4 outlines the research questions. In section 3.5, we present the research objectives and section 3.6 discusses the approach taken in this thesis to problem solving. Section 3.7 concludes the chapter.

3.2. Key terms and concepts

3.2.1. Cloud computing

Cloud computing is an emerging technology that will continue to bring advantages to business. Many researchers (Gill 2011; Gupta 2010; Hofmann & Woods 2010) define cloud computing as a multitenant architecture where multiple organisations are given services by one cloud provider. Cloud computing includes both the delivery of hosted services over the Internet as well as the hardware and software in the data centers that provide the services (Armbrust et al. 2010). In brief, cloud computing is a way of using business over the Internet.

3.2.2. Reputation system

The main concept of reputation systems is to let an agent rate the performance of other agents. In reputation systems, agents rate another agent to describe their experience of a particular transaction with that agent. Reputation systems are effective in providing a motivation for honest behaviour and preventing dishonest behaviour amongst the agents (Buchegger & Le Boudec 2003).

3.2.3. Incorrect review

An incorrect review is considered a user opinion on products and services that is extremely unrelated, untrustworthy and untruthful. These kinds of reviews are published on a retailer's site and third-party review portals such as www.getapp.com, www.cloudreviews.com, www.hostingadvice.com, etc., therefore they can be very damaging (Dellarocas 2000; Duh et al. 2013). Moreover, reviews that are duplicate or near-duplicate written by the same reviewers on different products or by different reviewers on the same products or different products are also considered as incorrect reviews. Incorrect reviews refer to fake reviews that mislead human readers by giving positive reviews or negative reviews in order to promote or demote some products that are targeted (Dellarocas 2000; Duh et al. 2013). These kinds of reviews can also monitor the expressions of the consumers so businesses can change their production and marketing strategies effectively. Therefore, they will also benefit business organisations (Dellarocas 2000; Duh et al. 2013).

3.2.4. Ballot stuffing

We regard *ballot stuffing* as an activity where a seller compromises with a number of buyers to ask them to give the seller an unfairly high rating. In doing so, these buyers inflate the reputation of the seller which may result in increased customers ordering products at a higher price (Dellarocas 2000).

3.2.5. Bad mouthing

We regard *bad mouthing* as an activity where a seller compromises with a number of buyers to ask them to give an unfairly low ratings to its enemy. In doing so, these buyers deflate the reputation of the seller's enemy which may result in a decreased number of customers ordering products or even no customers at all (Dellarocas 2000).

3.2.6. Spammer group

Mukherjee et al. 2011 defined a *spammer group* as a group of reviewers who work together to write fake reviews on target products in order to promote or discredit these products.

3.2.7. Clique

Salkind 2008 defined a *clique* as a group of people who interact with each other more regularly and intensely than others in the same settings.

In the context of this thesis, a *clique* is a group of agents who work together to promote their products. Agents in a clique form a review circle as one agent will receive a review from another agent and will also provide a review on a different agent.

3.2.8. Internal factor-based methods

We regard an internal factor-based method as methods that are based on analyzing and comparing the values themselves only, not any other values.

3.2.9. External factor-based methods

We regard external factor-based methods as methods other than internal factor-based methods, for example, the reputation of the raters will be used to determine the weight of the ratings.

3.2.10. Linguistic-based method

We regard linguistic-based methods as methods that use the textual content of reviews to identify spam. This method was the first to identify product review spamming behaviour.

3.2.11. Behaviour-based method

We regard behaviour-based methods as methods that analyze reviewers' regular patterns to infer if they are writing incorrect reviews.

3.2.12. Network-based method

We regard network-based methods as methods that use a review network which is a graph consisting of reviewers and products, or reviews, reviewers and products.

3.2.13. Agent

We regard an agent as an intelligent piece of software or a human being which is either acting for itself or on behalf of a company, business or an individual and has the power, authority and ability to autonomously make decisions or carry out business activities.

3.3. Problem definition

As discussed in Chapter 1, spammers may write fake positive or negative reviews in order to promote or demote a targeted product. These reviews can mislead humans in their decision to buy or not to buy a specific product. Moreover, they can also play an important role in monitoring the consumers' expressions so that businesses can change their production and marketing strategies to be more effective. In doing so, businesses gain a benefit (Dellarocas 2000; Duh et al. 2013). Due to the significant impact of incorrect reviews in various aspects, there is a high need to detect and delete incorrect reviews in general and in cloud reputation systems specifically.

Therefore, several researchers have made significant advances and impacts on methods to identify incorrect reviews. As pointed out in Chapter 2, four types of incorrect reviews are discussed in this thesis: ballot stuffing, bad mouthing, spammer group and clique.

As discussed in Chapter 2, we categorize the methods to identify ballot stuffing and bad mouthing into two types which are internal factor-based and external factor-based methods. By internal factor-based methods, we mean methods that are based on analyzing and comparing the values themselves only, not any other values. By external factor-based methods, we mean that other methods in addition to internal factors such as the reputation of the raters will be used to determine the weight of the ratings. In order to understand the capabilities of the existing ballot stuffing and bad mouthing methods, we reviewed them thoroughly and documented them in Chapter 2. The capabilities of the existing ballot stuffing and Bad-Mouthing methods are majority, burstiness and relationship. Majority is the capability of a method to identify ballot stuffing and bad mouthing even when the majority of the ratings of a provider are unfair (Zhang et al. 2008). Most of the approaches proposed to detect ballot stuffing and bad mouthing do not work well in relation to a Majority capability. Burstiness is the capability of an approach to deal with the situation when a large number of ratings are provided by advisors within a short timeframe (Zhang et al. 2008). Most of the methods to identify ballot stuffing and bad mouthing are not able to deal with the burstiness capability. By relationship, we mean the ability (of the method) to take into account the relationship between the reviewers, reviews and products. By calculating the suspicion score based on the relationship between the reviewers, reviews and products, the accuracy of an algorithm will be much improved. In chapter 9 of this thesis, we check the accuracy of the algorithms and compare them. Our method can take into account all three factors of majority, burstiness and relationship.

As discussed in Chapter 2, there are only a few research studies which focus on identifying spammer groups which can be categorized in three types: linguistic-based, behaviour-based and network-based. Linguistic-based methods are methods that use the textual content of reviews to identify spam. This method was the first to identify product review spamming behaviour. Behaviour-based methods are methods that analyze reviewers' regular patterns to infer if they are writing incorrect reviews. Network-based methods are methods that use a review network which is a graph consisting of reviewers and products, or reviews, reviewers and products. All of the existing research on detecting spammer groups will find candidate groups first and then check which groups among those groups are spammer groups. This can be more time consuming for grouping all reviews, especially when there is are large number of reviews. Due to the scope of the thesis, we did not do a simulation to check the time difference between our methods and other methods, however, it can be easily proved by theory that it will be more time consuming if we find all the groups first and check which groups are spammer groups rather than finding a single spam review and then find all the groups to which it belongs. In future work, this simulation will be carried out to check the time difference between our methods and other methods.

Furthermore, to the best of our knowledge, there is no definition of a clique in relation to cloud reputation systems even though there may be a lot of cliques in the real world. Therefore, no previous research was found to identify cliques in cloud reputation systems.

Based on the above overview and description of the problem, we formally define the problem that we intend to address in this thesis as follows:

How can incorrect reviews such as ballot stuffing, bad mouthing, spammer groups and cliques in cloud reputation systems be identified?

The next section describes the research issues that need to be addressed in order to solve the abovementioned problem.

3.4. Research questions

In order to address the above research questions, we identify the following sub-questions:

- Research question 1: How can ballot stuffing-based reviews be intelligently identified in cloud reputation systems?
- Research question 2: How can bad mouthing-based reviews be intelligently identified in cloud reputation systems?
- Research question 3: How can spammer groups be intelligently identified in cloud reputation systems?
- Research question 4: How can cliques be intelligently identified in cloud reputation systems?
- Research question 5: How can the methods developed above be validated on a dataset gathered from real cloud reputation systems?

3.5. Research objectives

Based on the research definition in the previous section, we define five research objectives that will be achieved in this thesis. They are as follows:

Research objective 1: Propose an intelligent method for identifying ballot stuffing in cloud reputation systems. Such a method would: (1) work well when the majority of the reviews are incorrect, (2) be able to deal with the situation when there is a large number of ratings provided by advisors within a short timeframe, and (3) use the relationship between reviews, reviewers and products so that it will be more accurate.

Research objective 2: Propose an intelligent method to identify bad mouthing in cloud reputation systems. Such a method would (1) work well when the majority of the reviews are incorrect, (2) be able to deal with the situation when there is a large number of ratings provided by advisors within a short timeframe, and (3) use the relationship between reviews, reviewers and products so that it will be more accurate.

Research objective 3: Propose an intelligent method to identify spammer groups in cloud reputation systems. Such a method would (1) utilize the method proposed in research objective 1 and research objective 2 to identify a single incorrect review first; then (2) find a group or a few groups to which the incorrect review found in (1) belongs.

Research objective 4: Define the concept clique and propose an intelligent method to identify cliques in cloud reputation systems.

Research objective 5: Validate the proposed methods to identify ballot stuffing, bad mouthing and spammer groups in cloud reputation systems. Due to the time restriction of the thesis, the method to identify cliques will not be validated in this thesis.

In this section, we clearly define each of the objectives in this thesis.

3.5.1. Research Objective 1: Propose an intelligent method for identifying ballot stuffing in cloud reputation systems.

As discussed in Chapter 2, there are various methods to identify ballot stuffing in the existing literature. Of these methodologies, we select three popular methodologies to identify ballot stuffing to compare with our method based on their accuracy, precision and recall. These three methods are BRS (Jøsang & Ismail 2002), TRAVOS (Teacy et al. 2006) and iCLUB (Liu et al. 2014, Jiang et al. 2013). As we detailed in section 2.6, most of the proposed methods are not efficient when more than half the reviews are ballot stuffing. Moreover, a few of the proposed methods do not work well when there is a large number of ratings provided by previous buyers in a short period of time. Furthermore, of the three selected methods, none use the relationship between reviews, reviewers and products to apply in their method to propose a more accurate algorithm to detect ballot stuffing. To address the three shortcomings, we need an intelligent method that would: (1) work well when the majority of the reviews are incorrect; (2) be able to deal with the situation when there is a large number of ratings provided by advisors within a short timeframe; (3) use the relationship between reviews, reviewers and products so that it will be more accurate. We propose a method that can also overcome all three shortcomings detailed in section 2.6. In addition, to address the shortcoming that the existing methods have not been applied to cloud reputation systems, our proposed method is implemented using a dataset collected from www.getapp.com using a web crawler (Alkalbani et al. 2016).

3.5.2. Research objective 2: Propose an intelligent method to identify bad mouthing in cloud reputation systems.

As mentioned in Chapter 2, there are various methodologies to identify bad mouthing. Of these, we select three popular methodologies to identify bad mouthing to compare with our method based on their accuracy, precision and recall. These three methods are BRS (Jøsang & Ismail 2002), TRAVOS (Teacy et al. 2006) and iCLUB (Liu et al. 2014, Jiang et al. 2013). As we presented in section 2.6, most of the proposed methods are not efficient when more than half the reviews are bad mouthing. Moreover, a few of the proposed methods do not work well when there is a large number of ratings provided by previous buyers in a short period of time. Furthermore, of the three selected methods, none of them use the relationship between reviews, reviewers and products to apply in their method to propose a more accuracy algorithm to detect bad mouthing. To address these three shortcomings, we need a method that would: (1) work well when the majority of the reviews are incorrect; (2) be able to deal with the situation when there is a large number of ratings provided by advisors within a short timeframe; and (3) use the relationship between reviews, reviewers and products so that it will be more accurate. We propose a method that can also overcome all three shortcomings listed in section 2.6. In addition, to address the shortcoming that the existing methods have not been applied to cloud reputation systems, our proposed method is implemented using a dataset collected from www.getapp.com using a web crawler (Alkalbani et al. 2016).

3.5.3. Research objective 3: Propose an intelligent method to identify spammer groups in cloud reputation systems

As discussed in Chapter 2, in the existing body of literature, there are only a few methods which identify spammer groups. The shortcoming with all the proposed methods is that they group all the reviews into different groups first and then use different algorithms to detect spammer groups. Grouping all the reviews first and then finding the spammer groups among all the groups takes more time than finding a single spam review first and then finding the relevant groups to which the identified single spam review belongs. Furthermore, as mentioned in Chapter 2, there is no integration or cooperation with other existing methods to detect all four types of spam reviews.

Therefore, in this thesis, in order to address this shortcoming, we propose an intelligent method to identify spammer groups that would: (1) utilize the methodology proposed in research issue 1 and research issue 2 to identify a single incorrect review first; (2) find a group or a few groups or users who posted the incorrect review found in (1).

The first approach that we propose uses K-means clustering to find the group to which the spam reviewer belongs, therefore it can detect one, and only one spammer group to which the reviewer belongs. In the second approach that we propose, we use a fuzzy K-means clustering algorithm to find all the groups to which a spam reviewer belongs. However, if a spam reviewer only writes reviews for one product, they are unlikely to earn much money. Therefore, a spam reviewer is more likely to work for more than one group. This is the reason we use a fuzzy K-means algorithm in our method instead of a K-means algorithm to detect spammer groups. In addition, to address the shortcoming in that the existing methods have not been applied to cloud reputation systems, our proposed method is implemented using a dataset collected from www.getapp.com using a web crawler (Alkalbani et al. 2016).

3.5.4. Research objective 4: Define the concept of cliques and propose an intelligent method to identify cliques in cloud reputation systems

To the best of the researcher's knowledge, there is no definition of the concept a clique in cloud reputation systems. This thesis will be the first to define a clique in cloud reputation systems. Furthermore, we apply a graph-based method to identify cliques in cloud reputation systems.

3.5.5. Research issue 5: Validate the proposed methods

There are many research studies on identifying ballot stuffing and bad mouthing in the existing literature. From an application viewpoint, none of the existing methods on identifying ballot stuffing and bad mouthing have been applied to or used in cloud reputation systems. Furthermore, from a technical viewpoint, there is a need to compare our method with the other proposed methods. Therefore, in this thesis, we introduce a framework for conducting experiments to compare different methods to identify incorrect reviews. To be specific, we compare our proposed method for ballot stuffing and bad mouthing with three popular methods which are BRS (Jøsang & Ismail 2002), TRAVOS (Teacy et al. 2006) and iCLUB (Liu et al. 2014, Jiang et al. 2013) using metrics such as accuracy, precision and recall.

As mentioned previously, there are only a few research studies on detecting spammer groups, and most of them find all the groups first and then decide which group is the spammer group. In this thesis, we propose two different approaches (compared to the existing methods to find spammer groups). Our methods make use of the ballot stuffing and bad mouthing reviews that we found previously, then find a groups or some groups to which these reviews belong. During the validation phase, we prove that our second method (fuzzy K-means clustering) outperforms our first method (K-means clustering) by using a test case. During the testing phase, we found that our second method (fuzzy K-means clustering) can detect more than one group while the first one (K-means clustering) can only detect one group. Reviewers are paid to write incorrect reviews, but they cannot earn enough money by only working for one group, hence it is necessary to detect all the groups to which an incorrect reviewer belongs. Therefore, our second method (fuzzy K-means clustering) is much better than our first method (K-means clustering). We also apply our method to the real dataset and compare our two methods using the t-test algorithm. In addition, to address the shortcoming that the existing methods have not been applied to cloud reputation systems, our proposed method is implemented using a dataset collected from www.getapp.com using a web crawler (Alkalbani et al. 2016).

As a clique is a new definition in our thesis and our method is also the first that attempts to detect cliques in cloud reputation systems, there is no previous methods with which to compare it. For this reason and the time constraints to complete the thesis, we leave the validation of the method for detecting cliques to our future research.

3.6. The research approach to problem solving

There are several research methodologies, such as design research, case study and action research. In this doctoral study, we use the design methodology which is illustrated in Figure 5. This methodology is the most appropriate to achieve our research objectives. Using this approach, the research starts with an in-depth study of the existing literature to obtain a detailed understanding of the current challenges in the methods to identify four types of incorrect reviews, ballot stuffing, bad mouthing, spammer groups and cliques. After this, a number of research gaps are identified and the problems that need to be addressed are presented. In the next step, the suggested feasible solutions are defined in Chapter 4 which leads to a tentative design for this study. In the next step, the solutions are developed and tested through experimentation. The solution artefacts are then evaluated and assessed against the existing approaches.

The research approaches for all the proposed research questions are presented in the next section.

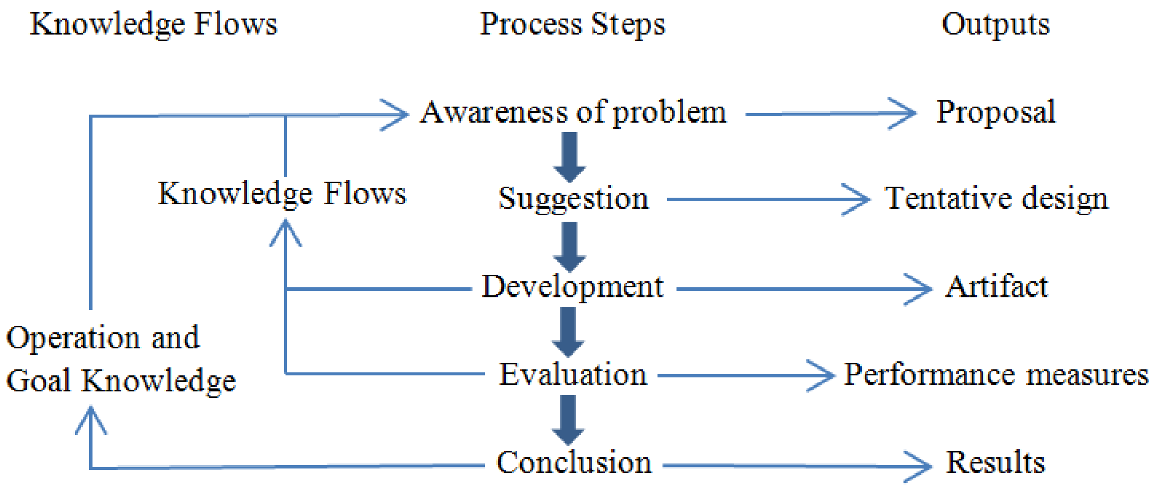


Figure 5: Research methodology (Peffers et al. 2007)

3.7. Conclusion

In this chapter, we defined some key terms and concepts that we use throughout our thesis. Furthermore, we provided the research problem, the research issues and the research questions. Finally, we select the most appropriate approach for our research. In the next chapter, we give an overview of our solution to the research problem presented in this chapter.

3.8. References

[1] Alkalbani, A.M., Ghamry, A.M., Hussain, F.K. and Hussain, O.K., 2016, ‘Harvesting Multiple Resources for Software as a Service Offers: A Big Data Study’, In *International Conference on Neural Information Processing*, pp. 61-71. Springer, Cham.

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I.O.N. & Zaharia, M. 2010, 'A View of Cloud Computing', *Communications of the ACM*, vol. 53, no. 4, pp. 50-8.

[3] Buchegger, S. & Le Boudec, J.Y., 2003. *A robust reputation system for mobile ad-hoc networks*, No. LCA-REPORT-2003-006.

[4] Gill, R. 2011, 'Why Cloud Computing Matters to Finance', *Strategic Finance*, vol. 92, no. 7, pp. 43-7.

[5] Gupta, A. 2010, 'Cloud computing growing interest and related concerns', *Computer Technology and Development (ICCTD), 2010 2nd International Conference on*, pp. 462-5.

[6] Hofmann, P. & Woods, D. 2010, 'Cloud Computing: The Limits of Public Clouds for Business Applications', *Internet Computing, IEEE*, vol. 14, no. 6, pp. 90-3.

[7] Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. 2007, ‘A design science research methodology for information systems research’, *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77.

CHAPTER 4:
SOLUTION OVERVIEW

4.1. Introduction

In Chapter 3, we presented an overview of all the solutions for the five issues listed in the previous chapter. This chapter presents the solution overview for all five issues listed in this thesis.

In section 4.2, we present an overview of the proposed solutions for all five proposed research issues that were discussed in the last chapter. Section 4.3, 4.4, 4.5 and 4.6 present an overview of the solution for identifying ballot stuffing, bad mouthing, spammer groups and cliques respectively. Section 4.7 concludes the chapter.

4.2. Overview of the proposed solutions

In this section, we present an overview of all the solutions for the five issues listed in Chapter 3.

In order to identify ballot stuffing, we calculate the suspicion score of three types of nodes, reviewers, reviews and products, in a review graph and then use naïve Bayes to conclude if a review is ballot stuffing or not.

The same method is used to identify bad mouthing, however, instead of taking into account high rating reviews, we use low rating reviews.

After detecting all the ballot stuffing and bad mouthing reviews, we find all the groups to which these reviews belong using *K-means* or *fuzzy K-means clustering*. By using this approach, we identify spammer groups.

A graph-based method is used to identify cliques. This method also makes use of the ballot stuffing detected previously.

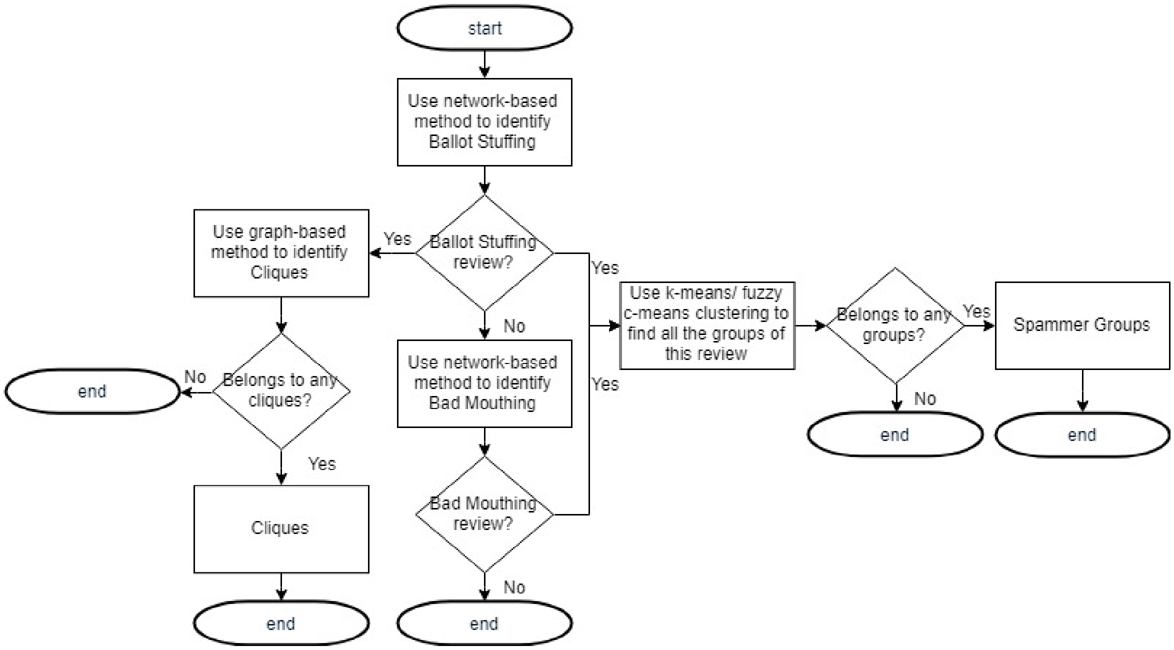

start
Use network-based method to identify Ballot Stuffing
Ballot Stuffing review?
Yes
Use graph-based method to identify Cliques
Belongs to any cliques?
No
end
Yes
Cliques
end
No
Use network-based method to identify Bad Mouthing
Bad Mouthing review?
No
end
Yes
Yes
Use k-means/ fuzzy c-means clustering to find all the groups of this review
Belongs to any groups?
Yes
Spammer Groups
end
No
end

Figure 6: The relationship between the four algorithms

4.3. Overview of the solution for identifying ballot stuffing

To identify ballot stuffing, we use a network-based method. The network-based method makes use of a review graph that consists of three types of nodes: reviewers, reviews, and products, as shown in Figure 7. The ballot stuffing reviewers are identified in four steps. The first three steps calculate a suspicion score for reviewers, reviews and products, respectively. The last step uses naïve Bayes to find the ballot stuffing reviews.

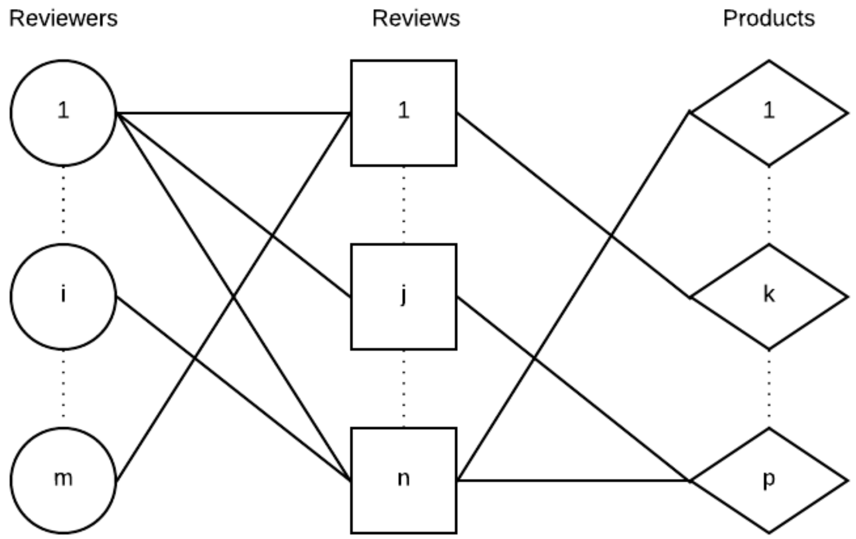


Figure 7: A review graph (ballot stuffing)

Step 1: Calculate the suspicion score of reviewers

To identify whether a review provided by a given reviewer is suspicious or not, we use/calculate the following attributes:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.
- The reviewer is a one-time reviewer or not.
- Ratio of the number of high-rating reviews by reviewer i to the number of reviews by i.
- The number of times review bursting occurs.
- Whether the review is high rating and duplicate.

Step 2: Calculate the suspicion score of reviews

A review is suspicious if it is a high-rating and duplicate or a high-rating and near-duplicate review. Therefore, the suspicion score of a review can be calculated based on this.

Step 3: Calculate the suspicion score of products

To identify whether a product is suspicious or not, we calculate the following attributes:

- The product has received bursting high-rating reviews.
- The product has received high-rating reviews from one-time reviewers.
- A substantial fraction of their high-rating reviews are posted in the shortest average period of time.
- A substantial fraction of their high-rating reviews are posted in the longest average period of time.

Step 4: Apply naïve Bayes to identify ballot stuffing

To determine which reviews are ballot stuffing, we apply naïve Bayes in this step. Details are given in the next chapter.

```
Algorithm to identify ballot stuffing
// Obtain training dataset of ballot stuffing reviews by network-based method
Input:
        A dataset D which consists of products, products' reviews and reviewers
Output:
        A training dataset T of ballot stuffing reviews
Step:
    1.  Calculate the suspicion score $SS_i$ of reviewer $i$ using Equation 6
    2.  Calculate the suspicion score $SS_j$ of review $j$ using Equation 7
    3.  Calculate the suspicion score $SS_k$ of product $k$ using Equation 12
// Apply naïve Bayes to identify all ballot stuffing reviews
Input:
        A training dataset T of ballot stuffing reviews
        $x = (x_1, x_2, x_3, \dots, x_n)$ //value of the predictor variable in testing dataset
Output:
        A class of testing dataset which contains ballot stuffing reviews
Step:
    1.  Read the training dataset T
    2.  Calculate the mean and variance of the predictor variables in each class using Equation 17 and
        Equation 18, respectively
    3.  Repeat
                Calculate the probability of $x_i$ using the Gauss density equation in each class
        Until the probability of all predictor variables $(x_1, x_2, x_3, \dots, x_n)$ has been calculated
    4.  Calculate the likelihood for each class
    5.  Get the greatest likelihood
```

Figure 8: Algorithm to identify ballot stuffing

4.4. Overview of the solution for identifying bad mouthing

To identify bad mouthing, we use a network-based method. The network-based method makes use of a review graph that consists of three types of nodes: reviewers, reviews, and products, as shown in Figure 7. Bad mouthing reviewers are identified in four steps. The first three steps calculate a suspicion score for reviewers, reviews and products, respectively. The last step employs naïve Bayes to identify bad mouthing.

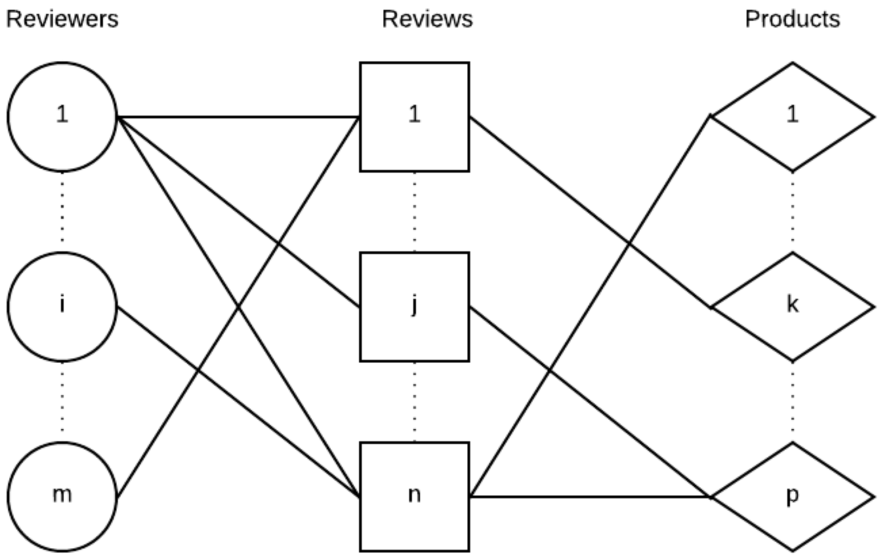


Figure 9: A review graph (bad mouthing)

Step 1: Calculate the suspicion score of reviewers

To identify whether a review provided by a given reviewer is suspicious or not, we calculate the following attributes:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.
- The reviewer is a one-time reviewer or not.
- Ratio of the number of low-rating reviews by reviewer i to the number of reviews by i.
- The number of times review bursting occurs.
- Whether the review is low-rating and duplicate.

Step 2: Calculate the suspicion score of reviews

A review is suspicious if it is a low-rating and duplicate or a low-rating and near-duplicate review. Therefore, the suspicion score of a review can be calculated based on this.

Step 3: Calculate the suspicion score of products

To identify whether a product is suspicious or not, we calculate the following attributes:

- The product has received bursting low-rating reviews.
- The product has received reviews from one-time reviewers.
- A substantial fraction of their low-rating reviews are posted in the shortest average period of time.
- A substantial fraction of their low-rating reviews are posted in the longest average period of time.

Step 4: Apply naïve Bayes to identify bad mouthing.

To determine which reviews are bad mouthing, we apply naïve Bayes in this step. Details are given in the next chapter.

Algorithm to identify bad mouthing

```
// Obtain training dataset of bad mouthing reviews by network-based method
Input:
        A dataset D which consists of products, products' reviews and reviewers
Output:
        A training dataset T of bad mouthing reviews
Step:
    1.  Calculate the suspicion score $SS_i$ of reviewer $i$ using Equation 24
    2.  Calculate the suspicion score $SS_j$ of review $j$ using Equation 25
    3.  Calculate the suspicion score $SS_k$ of product $k$ using Equation 30
// Apply naïve Bayes to identify all bad mouthing reviews
Input:
        A training dataset T of bad mouthing reviews
        $x = (x_1, x_2, x_3, \dots, x_n)$ //value of the predictor variable in testing dataset
Output:
        A class of testing dataset which contains bad mouthing reviews
Step:
    1.  Read the training dataset T
    2.  Calculate the mean and variance of the predictor variables in each class using Equation 35 and
        Equation 36, respectively
    3.  Repeat
                Calculate the probability of $x_i$ using the Gauss density equation in each class
        Until the probability of all predictor variables $(x_1, x_2, x_3, \dots, x_n)$ has been calculated
    4.  Calculate the likelihood for each class
    5.  Get the greatest likelihood
```

Figure 10: Algorithm to identify bad mouthing

4.5. Overview of the solution for identifying spammer groups

To identify spammer groups, we propose the following two steps:

- Stage 1: Identify ballot stuffing and bad mouthing using the method in section 4.3 and 4.4, respectively. At the end of this step, we can determine which reviews are ballot stuffing and which are bad mouthing. This stage includes four steps as mentioned in sections 4.3 and 4.4.
- Stage 2: Find the groups of ballot stuffing and bad mouthing reviews identified in stage 1: In order to achieve this, we use different approaches. We can use the Pearson correlation coefficient, Spearman correlation coefficient, K-mean clustering or fuzzy K-mean clustering to find the group to which the suspicion reviewer can belong. However, in this thesis, we only validate two algorithms to find the spammer groups to which one single incorrect review can belong. These two algorithms are K-means and fuzzy K-means clustering.

Algorithm to identify spammer group

```
// Identify ballot stuffing and bad mouthing
Input:
        A dataset D which consists of products, products' reviews and reviewers
Output:
        A set of all the ballot stuffing and bad mouthing reviews
Step:
    1.  Identify ballot stuffing using the methods in section 4.3
    2.  Identify bad mouthing using the methods in section 4.4
// Find the groups of identified ballot stuffing and bad mouthing reviews using K-means clustering algorithm
Input:
        A set of all ballot stuffing and bad mouthing reviews and a data set that we need to identify spammer groups from $X = (x_1, x_2, x_3, \ldots, x_n)$ // set of n data items
        K //number of desired cluster
Output:
        A set of K spammer groups
Step:
    1.  Select an initial partition with K clusters
    2.  Repeat
                Assign each item $x_i$ to the cluster which has the closest centroid
                Compute new mean for each cluster
        Until cluster membership stabilizes
// Find the groups of identified ballot stuffing and bad mouthing reviews using K-means clustering algorithm
Input:
        A set of initial cluster centres $\mathrm{SC}_0 = \{\mathrm{C}_\mathrm{j}(0)\}$, C: centroid matrix, ε: threshold value used as stopping criteria
Output:
        C: updated centroid matrix
Step:
    1.  Randomly initialize the fuzzy partition matrix $U = [u_{i,j}]$
    2.  Repeat
```

```
            Update the membership using Equation 77
            Calculate the cluster center using Equation 78
        Until $\|C_j(p) - C_j(p-1)\| < \varepsilon \; for \; j = 1 \; to \; k$
    3.  return C
```

Figure 11: Algorithm to identify spammer groups

4.6. Overview of the solution for identifying cliques

To identify cliques, we use a graph-based method comprising the following nine steps:

- Step 1: Check whether a review is ballot stuffing using the method in section 4.3, then calculate the total number of stars in all the reviews.
- Step 2: Find the shortest path distance between two nodes and calculate the influence of each node. In this step, we need to find the shortest path distance between the source and the destination node using the Dijkstra algorithm. The influence of each node is calculated based on Epanechnikov kernel.
- Step 3: Define the influence region and direct region for each node. The direct region of a node is a collection of nodes directly having reviews for that node. The influence region of a node is the extension of the direct region because not only is the direct level considered, but also the nodes with indirect connections to that node are included if the path distance is smaller than h.
- Step 4: Calculate the density of each node. In this context, the density of each node demonstrates how many reviews that provider received from others and how highly that provider is rated.
- Step 5: Identify and select all core nodes from the node list. A core node must have a density value greater than or equal to a minimum density τ, which means that the provider must receive reviews from at least a number of other providers with a minimum star rating.
- Step 6: Cluster the list of core nodes into separated subsets. These clusters are identified based on the direct region of each node. If a provider belongs to a core cluster, all providers reviewing for that provider and all providers being reviewed by it will also belong to that cluster. Due to this characteristic, all of these clusters are separated from each other.

- Step 7: Identify cliques – strong core clusters. In our context, strong core clusters are cliques because in order to become a strong core cluster, it must fulfil the core path property which means that for all pairs of providers (s, d) in the strong core, there is not only a core path from s to d but also another reverse path from d to s. That is, the providers review each other and form a review circle.
- Step 8: Identify semi-strong core clusters. A semi-strong core cluster does not have bi-directional paths between each pair of providers in the cluster, but only requires at least a path between each pair. In our context, semi-strong clusters are similar to any providers with interactions with any of the actors in a clique. Therefore, the providers in a semi-strong cluster will have less involvement in spam review activities than the providers in a strong cluster.
- Step 9: Update providers' reputation. Initially, the reputation mark of each provider is initialized by 100. If a provider is identified as a clique, its reputation is subtracted by 70%. On the other hand, as mentioned in Step 7, providers in semi-strong core clusters have a lower risk of participating in writing incorrect reviews, so the reputation marks of each of these providers is subtracted by 30%. Reputation is one of the criteria in searching service providers, which means that providers with a higher reputation have a higher priority in the search result list.

Algorithm to identify cliques

```
// Identify ballot stuffing and bad mouthing
Input:
        A dataset D which consists of products, products' reviews and reviewers
Output:
        A set of all the ballot stuffing reviews
Step:
        Identify ballot stuffing using the methods in section 4.3
// Apply a graph-based method to identify cliques
Input:
        A set of all ballot stuffing reviews and a dataset that we need to identify cliques from
Output:
        A set of cliques, a set of semi-strong core clusters and updated providers' reputation
Step:
    1.  Calculate the total number of stars in all the reviews using Equation 79
    2.  Find the shortest path distance between two nodes using Equation 80 and calculate the
        influence of each node using Equation 81
    3.  Define the influence region and direct region for each node using Equation 82 and Equation 83,
        respectively
    4.  For each node, calculate the density using Equation 84
    5.  Identify and select all core nodes from the node list using Equation 86
```

```
6. Cluster the list of core nodes into separated subsets using Equation 87
7. Identify cliques (strong core clusters) using Equation 89
8. Identify semi-strong core clusters using Equation 90
9. Update providers' reputation
```

Figure 12: Algorithm to identify cliques

4.7. Overview of the experiment

In order to validate the approach developed to identify ballot stuffing, we compare our method with BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. We conduct the experiment with different percentages of ballot stuffing injected. Moreover, we also compare our method with the other three existing methods using FPR (false positive rate) and FNR (false negative rate) to prove which method is capable of majority (that is, the ability to identify ballot stuffing when the majority is ballot stuffing). Furthermore, in order to test the burstiness capability of our method and other existing methods, we carry out an experiment on one random agent in a period of time (different percentage of ballot stuffing reviews in different periods of time).

In order to validate the approach developed to identify bad mouthing, we compare our method with BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. We conduct the experiment with different percentages of bad mouthing injected. Moreover, we also compare our method with the three existing methods using FPR (false positive rate) and FNR (false negative rate) to prove which method is capable of majority (that is, the ability to identify bad mouthing when the majority is bad mouthing). Furthermore, in order to test the burstiness capability of our method and other existing methods, we conduct an experiment on one random agent in a period of time (different percentage of bad mouthing reviews in different periods of time).

In order to validate the methods developed to identify spammer groups, we compare our two proposed methods with each other: one method uses *K-means clustering* and one uses *fuzzy K-means clustering*. One way by which we validate the methods is to prove that when using K-means clustering, we can find one spammer group for every spam review, however, when using fuzzy K-means clustering, we find more than one group for every spam review. Another way by which we validate our method is that we run our two methods to find all the spammer groups in different periods of time and then compare our two methods using t-test.

4.8. Conclusion

In this chapter, we presented an overview of the solutions for all the issues proposed in Chapter 3. In section 4.2, an overview of the solutions for all the five issues proposed in Chapter 3 was presented. Section 4.3, 4.4 4.5 and 4.6 subsequently presented the overview of the solutions to identify ballot stuffing, bad mouthing, spammer groups and cliques, respectively. In section 4.7, we gave an overview of the experiment.

In the next chapter, we provide more details on how to identify the four types of incorrect reviews, ballot stuffing, bad mouthing, spammer groups and cliques.

4.9. References

[1] Ott, M., Choi, Y., Cardie, C. & Hancock, J. T. 2011, '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, Association for Computational Linguistics*, vol. 1, pp. 309-19.

CHAPTER 5:
NETWORK-BASED METHOD TO IDENTIFY BALLOT STUFFING

5.1. Introduction

In this chapter, we introduce the details of our solution to identify ballot stuffing.

Ballot stuffing reviews are identified using a network-based method. The network-based method makes use of a review graph that consists of three types of nodes, reviewers, reviews and products. To identify an individual spam reviewer, a network method is applied which consists of four steps. In the first three steps, the suspicion score of reviewers, reviews and products are calculated respectively. Then, naïve Bayes is used to detect ballot stuffing reviews based on the relationship of the reviewer with its related reviews and products.

In the first step, we calculate the suspicion score of a reviewer based on five attributes which are rating deviation, one-time reviewer, ratio of the number of positive reviews by reviewer i to the number of reviews by i, bursting review times and duplicate reviews. A reviewer is considered to be ballot stuffing if a rating deviation by a reviewer is much higher compared to the average rating of a reviewed product, or that reviewer only posts one review for one product, or that reviewer only posts all 4-star or 5-star ratings reviews, or that reviewer provides too many good reviews for the same product, or that reviewer posts duplicate reviews. Therefore, the five factors which are rating deviation, one-time reviewer, ratio of the number of positive reviews by reviewer i to the number of reviews by i, bursting review times and duplicate reviews will be taken into account to calculate the suspicion score of a reviewer.

In the second step, we calculate the suspicion score of a review. A review is considered to be ballot stuffing if it is a duplicate or near-duplicate review. Therefore, we calculate the suspicion score of a review based on this factor.

In the third step, we calculate the suspicion score of a product. A product is suspected of receiving review spams if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors.

In the last step, we use a training set obtained by calculating the suspicion score of the reviewer, review, product of the ballot stuffing reviews (these reviews are known to be ballot stuffing), then apply these to the dataset crawled from the website www.getapp.com using naïve Bayes to identify all the ballot stuffing reviews that may be in the dataset crawled.

5.2. Solution to identify ballot stuffing

5.2.1. Step 1: Calculate the suspicion score of reviewers

A reviewer is suspected to be ballot stuffing if

- the reviewer provides high rating deviation compared to the average rating of a reviewed product, or
- the reviewer is a one-time reviewer, or
- the reviewer posts all positive reviews (4-star or 5-star ratings only), or
- the reviewer posts too many good reviews for the same product, or
- the reviewer posts the same review a few times.

Therefore, to identify whether a review provided by a given reviewer is suspicious or not, the following attributes are taken into account:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.

$$dev_{ik} = \frac{|r_{ik} - \overline{r_{ik}}|}{4}$$

Equation 1

where r_{ik} are the ratings of reviewer i for product k and $\overline{r_{ik}}$ is the average rating of product k that reviewer i has reviewed. Since the ratings are integers between 1 and 5, the term is divided by 4 to be normalized between 0 and 1.

- One-time reviewer

$$ot_i = \begin{cases} 1\ if\ i\ is\ a\ one - time\ reviewer \\ 0\ if\ i\ is\ not\ a\ one - time\ reviewer \end{cases}$$

Equation 2

- Ratio of the number of positive reviews by reviewer i to the number of reviews by i.

$$rat_i = \frac{np_i}{n_i}$$

Equation 3

where np_i denotes the total number of positive reviews by reviewer i and n_i denotes the total number of reviews by reviewer i.

- Bursting review times, or $burstiness_i$:

$$brt_i = 1 - \frac{nd_i}{n_i}$$

Equation 4

where brt_i is the bursting review times by i, nd_i is the total number of times that i writes distinct reviews and n_i is the total number of times that i posts reviews.

- Posting duplicate reviews

$$dup_i = \begin{cases} 1\ if\ i\ has\ posted\ duplicate\ reviews \\ 0\ if\ i\ has\ not\ posted\ duplicate\ reviews \end{cases}$$

Equation 5

where dup_i is an attribute to check if reviewer i has posted the same review a few times. The attribute returns a value of 1 if reviewer i has posted duplicate reviews. Otherwise, the attribute returns a value of 0.

Finally, the suspicion score for reviewer i is calculated as follows:

$$SS_i = \frac{1}{5}(dev_{ik} + ot_i + rat_i + brt_i + dup_i)$$

Equation 6

5.2.2. Step 2: Calculate the suspicion score of a review

A review is suspicious if it is a (near-)duplicate review. Therefore, we can use Equation 7 to calculate the suspicion score of a review:

$$SS_j = max_{i \neq j} Cosine'(review_j, review_s)$$

$$s.t. \exists product\ k\ and\ (j,k),(s,k) \in E$$

Equation 7

where E is the edge set of the review graph and cosine' is the cosine linearly scaled between 0 and 1 to be consistent with the other features.

5.2.3. Step 3: Calculate the suspicion score of product k

A product is suspected of receiving review spams if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors. The suspicion score of product k is given by the following attributes:

- It has received bursting high-rating reviews

$$brt_k = 1 - \frac{nd_k}{n_k}$$

Equation 8

where brt_k is the bursting review times for product k, nd_k is the total number of distinct reviews written for k and n_k is the total number of reviews posted for product k.

- A significant fraction of its high ratings are posted by one-time reviewers.

by_ot_k represents what fraction of a high-rating review for product k is from one-time reviewers, which is given in Equation 9.

$$by\_ot_k = \frac{np\_ot_k}{np_k}$$

Equation 9

where np_ot_k is the total number of positive reviews for product k by one-timers, and np_k is the total number of positive reviews for product k.

- A substantial fraction of their high ratings are posted in the shortest average period of time.

The attribute $sapt_k$ is used to present the $burstiness_k$ in its shortest average period of time. To calculate the shortest average period of time, we use Equation 10.

$$sapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} min\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 10

where np_k is the number of positive reviews for k, dd is date distance between each review and r_m is its temporary nearest neighbour.

- A substantial fraction of their high ratings are posted in the longest average period of time.

The attribute $lapt_k$ is used to present the $burstiness_k$ for its longest average period of time. To calculate the longest average period of time, we use Equation 11.

$$lapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} max\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 11

where np_k is the number of positive reviews for k and dd is the date distance between each review and r_m is its temporary nearest neighbour.

As a last step, the suspicion score of product k is calculated as follows:

$$SS_k = \frac{1}{4}(brt_k + by\_ot_k + in\_best\_sapt_k + in\_best\_lapt_k)$$

Equation 12

5.2.4. Step 4: Apply naïve Bayes to identify if the review is ballot stuffing or not

In this step, we apply naïve Bayes to classify which review is a ballot stuffing review and which review is not a ballot stuffing review.

Let a vector $x = (x_1, \ldots, x_n)$ represent some n features (independent variables) which is assigned to the following instance probabilities:

$$p(C|x_1, \ldots, x_n)$$

Equation 13

for each k possible outcomes of class C.

The problem with the above formulation is that if the number of features *n* is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

Equation 14

$$P(C|x) = P(x_1|C) * P(x_2|C) * \ldots * P(x_n|C) * P(C)$$

Equation 15

where $P(C|x)$ is the posterior probability of class C given that the predictor (attribute) is true

$P(C)$ is the prior probability of class C

$P(x|C)$ is the likelihood which is the probability of the predictor, given that class C is true

$P(x)$ is the prior probability of predictor

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by class, and then compute the mean and variance of x in each class. Let μ be the mean of the values in x associated with class C and let σ^2 be the variance of the values in x associated with class C. Suppose we have collected some observation value v. Then, the probability *distribution* of v given a class C, $P(x{=}v|C)$, can be computed by plugging v into the equation for a normal distribution parameterized by μ and σ^2. That is,

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k{}^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k{}^2}}$$

Equation 16

Mean and variance is computed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 17

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Equation 18

5.3. Conclusion

Chapter 5 presents the solution for our research question 1. In this chapter, a network-based method to identify ballot stuffing is proposed. This method makes use of a review graph which comprises three nodes, reviewers, reviews and products. There are four steps in the process of identifying ballot stuffing. Firstly, the suspicion score of the reviewers are calculated using attributes such as rating deviation, one-time reviewer, ratio of the number of positive reviews by reviewer i to the number of reviews by i, bursting review times and posting duplicate reviews. In the second step, the suspicion score of reviews is calculated based on the fact that a review which is a duplicate or near-duplicate is suspected as a spam review. Then, the suspicion score of the product is calculated by checking if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Finally, naïve Bayes is used to detect the ballot stuffing reviews based on the relationship of the reviewer with its related reviews and products.

CHAPTER 6:
NETWORK-BASED APPROACH FOR IDENTIFYING BAD MOUTHING

6.1. Introduction

In this chapter, we introduce a method to identify bad mouthing where the same approach as that to identify ballot stuffing is used but instead of taking into account high-rating reviews, we use low-rating reviews.

Bad mouthing reviews are identified using a network-based method. The network-based method makes use of a review graph that consists of three types of node, reviewers, reviews and products. To identify an individual spam reviewer, a network method is applied which consists of four steps. In the first three steps, the suspicion score of reviewers, reviews and products is calculated, respectively. Then, naïve Bayes is used to detect bad mouthing reviews based on the relationship of the reviewer with its related reviews and products.

In the first step, we calculate the suspicion score of a reviewer based on five attributes which are rating deviation, one-time reviewer, ratio of the number of negative reviews by reviewer i to the number of reviews by i, bursting review times and duplicate reviews. A reviewer is considered to be bad mouthing if a rating deviation by a reviewer is much higher compared to the average rating of a reviewed product, or that reviewer only posts one review for one product, or that reviewer only posts all 1-star or 2-star ratings reviews, or that reviewer provides too many bad reviews for the same product, or that reviewer posts duplicate reviews. Therefore, these five attributes will be taken into account to calculate the suspicion score of a reviewer.

In the second step, we calculate the suspicion score of a review. A review is considered to be bad mouthing if it is a duplicate or near-duplicate review. Therefore, we calculate the suspicion score of a review based on this factor.

In the third step, we calculate the suspicion score of a product. A product is suspected of getting review spams if it has received bursting low-rating reviews, or a significant fraction of its low ratings are posted by one-time reviewers, or most of its low ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors.

In the last step, we use a training set obtained by calculating the suspicion score of the reviewer, review and product of the bad mouthing reviews (these reviews are known to be bad mouthing), then apply it to the dataset crawled from the website www.getapp.com using naïve Bayes to identify all the bad mouthing reviews that may be in the dataset.

6.2. Solution to identify bad mouthing

6.2.1. Step 1: Calculate the suspicion score of reviewers

A reviewer is suspected to be bad mouthing if

- the reviewer has a low rating deviation compared to the average rating of a reviewed product, or
- the reviewer is a one-time reviewer, or
- the reviewer posts all negative reviews (1-star or 2-star ratings only), or
- the reviewer posts too many bad reviews for the same product, or
- the reviewer posts the same review a few times.

Therefore, to identify whether a review provided by a given reviewer is suspicious or not, the following attributes are taken into account:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.

$$dev_{ik} = \frac{|r_{ik} - \overline{r_{ik}}|}{4}$$

Equation 19

where r_{ik} are the ratings of reviewer i for product k and $\overline{r_{ik}}$ is the average rating of product k that reviewer i has reviewed.

- One-time reviewer

$$ot_i = \begin{cases} 1\ if\ i\ is\ a\ one - time\ reviewer \\ 0\ if\ i\ is\ not\ a\ one - time\ reviewer \end{cases}$$

Equation 20

- Ratio of the number of low-rating reviews by reviewer i to the number of reviews by i.

$$rat_i = \frac{nn_i}{n_i}$$

Equation 21

where nn_i denotes the total number of low-rating reviews by reviewer i and n_i denotes the total number of reviews by reviewer i.

- Bursting review times, or $burstiness_i$:

$$brt_i = 1 - \frac{nd_i}{n_i}$$

Equation 22

where brt_i is the bursting review times by i, nd_i is the total number of times that i writes distinct reviews and n_i is the total number of times that i posts reviews.

- Posting duplicate reviews

$$dup_i = \begin{cases} 1\ if\ i\ has\ posted\ duplicate\ reviews \\ 0\ if\ i\ has\ not\ posted\ duplicate\ reviews \end{cases}$$

Equation 23

where dup_i is an attribute to check if reviewer i has posted the same review a few times. The attribute returns a value of 1 if reviewer i has posted duplicate reviews. Otherwise, the attribute returns a value of 0.

Finally, the suspicion score for reviewer i is calculated as follows:

$$SS_i = \frac{1}{5}(dev_{ik} + ot_i + rat_i + brt_i + dup_i)$$

Equation 24

6.2.2. Step 2: Calculate the suspicion score of a review

A review is suspicious if it is a (near-)duplicate review. Therefore, we can use Equation 25 to calculate the suspicion score of a review:

$$SS_j = max_{i \neq j} Cosine'\left(review_j, review_s\right)$$

$$s.t. \exists product\ k\ and\ (j,k),(s,k) \in E$$

Equation 25

where E is the edge set of the review graph and cosine' is the cosine linearly scaled between 0 and 1 to be consistent with the other features.

6.2.3. Step 3: Calculate the suspicion score of product k

A product is suspected of receiving review spams if it has received bursting low-rating reviews, or a significant fraction of its low ratings are posted by one-time reviewers, or most of its low ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors. The suspicion score of product k is given by the following attributes:

- It has received bursting low-rating reviews

$$brt_k = 1 - \frac{nd_k}{n_k}$$

Equation 26

where brt_k is the bursting review times for product k, nd_k is the total number of distinct reviews written for k and n_k is the total number of reviews posted for product k.

- A significant fraction of its low ratings are posted by one-time reviewers.

by_ot_k represents the fraction of a high-rating review for product k from one-time reviewers, which is given in Equation 27.

$$by\_ot_k = \frac{nn\_ot_k}{nn_k}$$

Equation 27

where nn_ot_k is the total number of negative reviews for product k by one-timers, and nn_k is the total number of negative reviews for product k.

- A substantial fraction of their low ratings are posted in the shortest average period of time.

The attribute $sapt_k$ is used to present the $burstiness_k$ in its shortest average period of time. To calculate the shortest average period of time, we use Equation 28.

$$sapt_k = \frac{1}{nn_k - 2} \sum_{m=1}^{np_k} min\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 28

where nn_k is the number of low-rating reviews for k, dd is the date distance between each review and r_m is its temporary nearest neighbour

- A substantial fraction of their low ratings are posted in the longest average period of time.

The attribute $lapt_k$ is used to present the $burstiness_k$ for its longest average period of time. To calculate the longest average period of time, we use Equation 29.

$$lapt_k = \frac{1}{nn_k - 2} \sum_{m=1}^{np_k} max\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 29

where nn_k is the number of low-rating reviews for k and dd is date distance between each review and r_m is its temporary nearest neighbour.

As a last step, the suspicion score of product k is calculated as follows:

$$SS_k = \frac{1}{4}(brt_k + by\_ot_k + in\_best\_sapt_k + in\_best\_lapt_k)$$

Equation 30

6.2.4. Step 4: Apply naïve Bayes to identify bad mouthing

In this step, we apply naïve Bayes to classify which review is a bad mouthing review and which review is not a bad mouthing review.

Let a vector $x \ = (x_1, \ldots, x_n)$ represent some n features (independent variables) which are assigned to the following instance probabilities:

$$p(C|x_1, \ldots, x_n)$$

Equation 31

for each k possible outcomes or class C

The problem with the above formulation is that if the number of features *n* is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

Equation 32

$$P(C|x) = P(x_1|C) * P(x_2|C) * \ldots * P(x_n|C) * P(C)$$

Equation 33

where $P(C|x)$ is the posterior probability of class C given that the predictor (attribute) is true,

$P(C)$ is the prior probability of class C,

$P(x|C)$ is the likelihood which is the probability of the predictor given that class C is true,

$P(x)$ is the prior probability of predictor.

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by class, and then compute the mean and variance of x in each class. Let μ be the mean of the values in x associated with class C and let σ^2 be the variance of the values in x associated with class C. Suppose we have collected some observation value v. Then, the probability *distribution* of v given a class C, $P(x{=}v|C)$, can be computed by plugging v into the equation for a normal distribution parameterized by μ and σ^2. That is,

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k{}^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k{}^2}}$$

Equation 34

Mean μ and variance σ^2 is computed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 35

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Equation 36

6.3. Conclusion

Chapter 6 clearly presented the solution for our research question 2. In this chapter, a network-based method to identify bad mouthing was proposed. This method makes use of a review graph which comprises three nodes, reviewers, reviews and products. There are four steps in the process of identifying bad mouthing. Firstly, the suspicion score of the reviewers is calculated using five attributes, rating deviation, one-time reviewer, ratio of the number of negative reviews by reviewer i to the number of reviews by i, bursting review times and posting duplicate reviews. In the second step, the suspicion score of reviews is calculated based on the fact that a review which is a duplicate or near-duplicate is suspected of being a spam review. Then, the suspicion score of the product is calculated by checking if it has received bursting low-rating reviews, or a significant fraction of its low ratings are posted by one-time reviewers, or most of its low ratings are posted in the period with its highest average ratings. Finally, naïve Bayes is used to detect bad mouthing reviews based on the relationship of the reviewer with its related reviews and products.

CHAPTER 7:
NETWORK-BASED APPROACH FOR IDENTIFYING SPAMMER GROUPS

7.1. Introduction

As discussed in Chapter 2, techniques that have already been employed to detect single spammers, or groups of them, can be divided into three categories: linguistic methods, behaviour methods and network methods. It is easier for spammers to mimic language and behaviour patterns than to mimic network-based patterns. For example, spammers can accommodate their use of certain language that has been detected to be associated with unfair reviews (Ott et al. 2011). On the other hand, it is harder to deceive network-level characteristics for the following reasons. Firstly, spammers often do not have a complete view of the whole review network because of its sheer range. Secondly, due to the limited budget, unfair reviewers would not repeat unimportant structures in the network. Therefore, this thesis focuses on detecting incorrect reviews using a network-based method.

In this chapter, we present two intelligent methods to identify spammer groups. The first method makes use of the ballot stuffing and bad mouthing reviews detected using the methods presented in Chapter 5 and Chapter 6, and then uses K-means clustering to find the group to which these reviews belong. The second method makes use of the ballot stuffing and bad mouthing reviews detected using the methods presented in Chapter 5 and Chapter 6, and then applies fuzzy K-means clustering to find all the groups to which these reviews belong.

7.2. Solution to identify spammer groups

We propose to combine different existing techniques that focus mainly on a network-based approach, with a view to creating a model that outperforms previous methods, enabling the identification of both individual spammers and group spammers with a high level of accuracy. The developed model is further enhanced by adding data mining techniques such as K-means clustering or fuzzy K-means clustering, allowing us to identify candidate groups of spammers. Through this process, it is possible to extend the efficacy of this technique to a group level.

There are two stages in the solution to identify spammer groups. In the first stage, we identify individual spam reviews. The second stage finds all the groups to which these individual spam reviews belong.

7.2.1. A K-means approach to identify spammer groups

7.2.1.1. Stage 1: Identify ballot stuffing/bad mouthing

Step 1: Calculate the suspicion score of reviewers

A reviewer is suspected to be ballot stuffing if

- the reviewer provides a high rating deviation compared to the average rating of a reviewed product, or
- the reviewer is a one-time reviewer, or
- the reviewer posts all positive reviews (4-star or 5-star ratings only), or
- the reviewer posts too many good reviews for the same product, or
- the reviewer posts the same review a few times.

Therefore, to identify whether a review provided by a given reviewer is suspicious or not, the following attributes are taken into account:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.

$$dev_{ik} = \frac{|r_{ik} - \overline{r_{ik}}|}{4}$$

Equation 37

where r_{ik} are the ratings of reviewer i for product k and $\overline{r_{ik}}$ is the average rating of product k that reviewer i has reviewed.

- One-time reviewer

$$ot_i = \begin{cases} 1\ if\ i\ is\ a\ one-time\ reviewer \\ 0\ if\ i\ is\ not\ a\ one-time\ reviewer \end{cases}$$

Equation 38

- Ratio of the number of high-rating reviews by reviewer i to the number of reviews by i.

$$rat_i = \frac{np_i}{n_i}$$

Equation 39

where np_i denotes the total number of high-rating reviews by reviewer i and n_i denotes the total number of reviews by reviewer i.

- Bursting review times, or $burstiness_i$:

$$brt_i = 1 - \frac{nd_i}{n_i}$$

Equation 40

where brt_i is the bursting review times by i, nd_i is the total number of times that i writes distinct reviews and n_i is the total number of times that i posts reviews.

- Posting duplicate reviews

$$dup_i = \begin{cases} 1\ if\ i\ has\ posted\ duplicate\ reviews \\ 0\ if\ i\ has\ not\ posted\ duplicate\ reviews \end{cases}$$

Equation 41

where dup_i is an attribute to check if reviewer i has posted the same review a few times. The attribute returns a value of 1 if reviewer i has posted duplicate reviews. Otherwise, the attribute returns a value of 0.

Finally, the suspicion score for reviewer i is calculated as follows:

$$SS_i = \frac{1}{5}(dev_{ik} + ot_i + rat_i + brt_i + dup_i)$$

Equation 42

Step 2: Calculate the suspicion score of a review

A review is suspicious if it is a (near-)duplicate review. Therefore, we can use Equation 43 to calculate the suspicion score of a review:

$$SS_j = max_{i \neq j} Cosine'\left(review_j, review_s\right)$$

$$s.t. \exists product\ k\ and\ (j,k),(s,k) \in E$$

Equation 43

where E is the edge set of the review graph and cosine' is the cosine linearly scaled between 0 and 1 to be consistent with the other features.

Step 3: Calculate the suspicion score of product k

A product is suspected of getting review spams if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors. The suspicion score of product k is given by the following attributes:

- It has received bursting high-rating reviews

$$brt_k = 1 - \frac{nd_k}{n_k}$$

Equation 44

where brt_k is the bursting review times for product k, nd_k is the total number of distinct reviews written for k and n_k is the total number of reviews posted for product k.

- A significant fraction of its high ratings are posted by one-time reviewers.

by_ot_k represents what fraction of a high-rating review for product k is from one-time reviewers, which is given in Equation 45.

$$by\_ot_k = \frac{np\_ot_k}{np_k}$$

Equation 45

where np_ot_k is the total number of positive reviews for product k by one-timers, and np_k is the total number of positive reviews for product k.

- A substantial fraction of their high ratings are posted in the shortest average period of time.

The attribute $sapt_k$ is used to present the $burstiness_k$ in its shortest average period of time. To calculate the shortest average period of time, we use Equation 46.

$$sapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} min\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 46

where np_k is the number of positive reviews for k, dd is date distance between each review and r_m is its temporary nearest neighbour.

- A substantial fraction of their high ratings are posted in the longest average period of time.

The attribute $lapt_k$ is used to present the $burstiness_k$ for its longest average period of time. To calculate the longest average period of time, we use Equation 47.

$$lapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} max\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 47

where np_k is the number of positive reviews for k and dd is date distance between each review and r_m is its temporary nearest neighbour.

As a last step, the suspicion score of product k is calculated as follows:

$$SS_k = \frac{1}{4}(brt_k + by\_ot_k + in\_best\_sapt_k + in\_best\_lapt_k)$$

Equation 48

Step 4: Apply naïve Bayes to identify if the review is ballot stuffing or not

In this step, we apply naïve Bayes to classify which review is a ballot stuffing review and which review is not a ballot stuffing review.

Let a vector $x = (x_1, \ldots, x_n)$ represent some n features (independent variables) which is assigned to the following instance probabilities:

$$p(C|x_1, \ldots, x_n)$$

Equation 49

for each k possible outcomes or class C.

The problem with the above formulation is that if the number of features *n* is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

Equation 50

$$P(C|x) = P(x_1|C) * P(x_2|C) * \ldots * P(x_n|C) * P(C)$$

Equation 51

where $P(C|x)$ is the posterior probability of class C given that the predictor (attribute) is true

$P(C)$ is the prior probability of class C

$P(x|C)$ is the likelihood which is the probability of the predictor given that class C is true

$P(x)$ is the prior probability of the predictor

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by class, and then compute the mean and variance of x in each class. Let μ be the mean of the values in x associated with class C and let σ^2 be the variance of the values in x associated with class C. Suppose we have collected some observation value v. Then, the probability distribution of v given a class C, $P(x=v|C)$, can be computed by plugging v into the equation for a normal distribution parameterized by μ and σ^2. That is,

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k{}^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k{}^2}}$$

Equation 52

Mean and variance are computed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 53

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Equation 54

7.2.1.2. *Stage 2: Find all spammer groups to which all ballot stuffing/bad mouthing reviews detected in Stage 1 belong using K-means clustering*

Let $X = \{x_i\}, i = \overline{1,n}$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k\}, k = \overline{1,K}$. The K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster are minimized. Let μ_k be the mean of cluster c_k. The squared error between μ_k and the points in cluster c_k is defined as

$$J(c_k) = \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$

Equation 55

The goal of K-means is to minimize the sum of the squared error over all K clusters,

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$

Equation 56

According to Jain and Dubes (1988), there are three main steps the in K-means clustering algorithm as follows:

(1) Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.

(2) Generate a new partition by assigning each pattern to its closest cluster centre.

(3) Compute new cluster centres.

7.2.2. A fuzzy approach to identify spammer groups

The method proposed in this paper comprises two stages. In Stage 1, a network method is applied to identify individual spam reviewers based on a score calculated by the network method. In Stage 2, a fuzzy K-means clustering algorithm is used to find the cluster(s) to which a spam reviewer belongs. As previously mentioned, fuzzy K-means clustering is able to reveal spammer affiliations with more than one cluster.

7.2.2.1. Stage 1: Identify ballot stuffing/bad mouthing

Step 1: Calculate the suspicion score of reviewers

A reviewer is suspected to be ballot stuffing if

- the reviewer provides a high rating deviation compared to the average rating of a reviewed product, or
- the reviewer is a one-time reviewer, or
- the reviewer posts all positive reviews (4-star or 5-star ratings only), or
- the reviewer posts too many good reviews for the same product, or
- the reviewer posts the same review a few times.

Therefore, to identify whether a review provided by a given reviewer is suspicious or not, the following attributes are taken into account:

- Rating deviation: A rating by a reviewer that greatly deviates from the average ratings for that product is significant.

$$dev_{ik} = \frac{|r_{ik} - \overline{r_{\imath k}}|}{4}$$

Equation 57

where r_{ik} are the ratings of reviewer i for product k and $\overline{r_{\imath k}}$ is the average rating of product k that reviewer i has reviewed.

- One-time reviewer

$$ot_i = \begin{cases} 1\ if\ i\ is\ a\ one - time\ reviewer \\ 0\ if\ i\ is\ not\ a\ one - time\ reviewer \end{cases}$$

Equation 58

- Ratio of the number of high-rating reviews by reviewer i to the number of reviews by i.

$$rat_i = \frac{np_i}{n_i}$$

Equation 59

where np_i denotes the total number of high-rating reviews by reviewer i and n_i denotes the total number of reviews by reviewer i.

- Bursting review times, or $burstiness_i$:

$$brt_i = 1 - \frac{nd_i}{n_i}$$

Equation 60

where brt_i is the bursting review times by i, nd_i is the total number of times that i writes distinct reviews and n_i is the total number of times that i posts reviews.

- Posting duplicate reviews

$$dup_i = \begin{cases} 1\ if\ i\ has\ posted\ duplicate\ reviews \\ 0\ if\ i\ has\ not\ posted\ duplicate\ reviews \end{cases}$$

Equation 61

where dup_i is an attribute to check if reviewer i has posted the same review a few times. The attribute returns a value of 1 if reviewer i has posted duplicate reviews. Otherwise, the attribute returns a value of 0.

Finally, the suspicion score for reviewer i is calculated as follows:

$$SS_i = \frac{1}{5}(dev_{ik} + ot_i + rat_i + brt_i + dup_i)$$

Equation 62

Step 2: Calculate the suspicion score of a review

A review is suspicious if it is a (near-)duplicate review. Therefore, we can use Equation 63 to calculate the suspicion score of a review:

$$SS_j = max_{i \neq j} Cosine'\left(review_j, review_s\right)$$

$$s.t. \exists product\ k\ and\ (j, k), (s, k) \in E$$

Equation 63

where E is the edge set of the review graph and cosine' is the cosine linearly scaled between 0 and 1 to be consistent with the other features.

Step 3: Calculate the suspicion score of product k

A product is suspected of receiving review spams if it has received bursting high-rating reviews, or a significant fraction of its high ratings are posted by one-time reviewers, or most of its high ratings are posted in the period with its highest average ratings. Therefore, we calculate the suspicion score of a review based on these factors. The suspicion score of product k is given by the following attributes:

- It has received bursting high-rating reviews

$$brt_k = 1 - \frac{nd_k}{n_k}$$

Equation 64

where brt_k is the bursting review times for product k, nd_k is the total number of distinct reviews written for k and n_k is the total number of reviews posted for product k.

- A significant fraction of its high ratings are posted by one-time reviewers.

by_ot_k represents the fraction of a high-rating review for product k from one-time reviewers, which is given in Equation 65.

$$by\_ot_k = \frac{np\_ot_k}{np_k}$$

Equation 65

where np_ot_k is the total number of positive reviews for product k by one-timers, and np_k is the total number of positive reviews for product k.

- A substantial fraction of their high ratings are posted in the shortest average period of time.

The attribute $sapt_k$ is used to present the $burstiness_k$ in its shortest average period of time. To calculate the shortest average period of time, we use Equation 66.

$$sapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} min\left(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\right)$$

Equation 66

where np_k is the number of positive reviews for k, dd is date distance between each review and r_m is its temporary nearest neighbour.

- A substantial fraction of their high ratings are posted in the longest average period of time.

The attribute $lapt_k$ is used to present the $burstiness_k$ for its longest average period of time. To calculate the longest average period of time, we use Equation 67.

$$lapt_k = \frac{1}{np_k - 2} \sum_{m=1}^{np_k} max\big(dd(r_m, r_{m-1}), dd(r_m, r_{m+1})\big)$$

Equation 67

where np_k is the number of positive reviews for k and dd is date distance between each review and r_m is its temporary nearest neighbour.

As a last step, the suspicion score of product k is calculated as follows:

$$SS_k = \frac{1}{4}(brt_k + by\_ot_k + in\_best\_sapt_k + in\_best\_lapt_k)$$

Equation 68

Step 4: Apply naïve Bayes to identify if the review is ballot stuffing or not

In this step, we apply naïve Bayes to classify which review is ballot stuffing review and which review is not ballot stuffing review.

Let a vector $x = (x_1, \ldots, x_n)$ represent some n features (independent variables) which are assigned to the following instance probabilities:

$$p(C|x_1, \ldots, x_n)$$

Equation 69

for each k possible outcomes or class C.

The problem with the above formulation is that if the number of features *n* is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

Equation 70

$$P(C|x) = P(x_1|C) * P(x_2|C) * \ldots * P(x_n|C) * P(C)$$

Equation 71

where $P(C|x)$ is the posterior probability of class C given that the predictor (attribute) is true

$P(C)$ is the prior probability of class C

$P(x|C)$ is the likelihood which is the probability of the predictor given that the class C is true

$P(x)$ is the prior probability of the predictor

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by class, and then compute the mean and variance of x in each class. Let μ be the mean of the values in x associated with class C and let σ^2 be the variance of the values in x associated with class C. Suppose we have collected some observation value v. Then, the probability *distribution* of v given a class C, $P(x=v|C)$, can be computed by plugging v into the equation for a normal distribution parameterized by μ and σ^2. That is,

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi{\sigma_k}^2}} e^{-\frac{(v-\mu_k)^2}{2{\sigma_k}^2}}$$

Equation 72

Mean and variance is computed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 73

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Equation 74

7.2.2.2. Stage 2: Find all spammer groups to which the ballot stuffing/bad mouthing reviews detected in Stage 1 belong using fuzzy K-means clustering

In this section, we present the fuzzy K-means clustering algorithm (Chang et al. 2011) to detect spamming groups. The algorithm divides the data points into k clusters $S_l(l = 1, 2, \ldots, k)$. Clusters S_l are associated with cluster centre C_l and this kind of relationship is fuzzy. The degree of association between data point X_i and cluster centreC_j is represented by a membership $u_{i,j} \in [0, 1]$. The set of data points is denoted as $S = \{X_i\}$.

The following distortion should be minimized with respect to cluster representation C_j and membership $u_{i,j}$:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{N} u_{i,j}^{m} \, d_{ij}$$

Equation 75

where *N* is the number of data points, *m* is the fuzzified parameter, *k* is the number of clusters and d_{ij} is the squared Euclidean distance between data point X_i and cluster centre C_j. Also, $u_{i,j}$ should satisfy the constraint in Equation 76.

$$\sum_{j=1}^{k} u_{i,j} = 1, for\ i = 1\ to\ N$$

Equation 76

The main process of the fuzzy K-means clustering algorithm is the mapping of a given set of representative vectors into an enhanced set by partitioning the data points. It starts with a set of initial cluster centres. The mapping process is repeated until it satisfies a stopping criterion. It is assumed that two clusters do not coincide. In the case that two cluster centres have the same cluster representative, a cluster centre should be disordered so that there is no coincidence in the iterative process. If $d_{ij} < \eta$, then $u_{i,j} = 1$ and $u_{i,l} = 0$ for $l \neq j$, where η is a very small positive number. The fuzzy K-means clustering algorithm can be illustrated in the following four steps:

Step 1: Input a set of initial cluster centres

Input a set of initial cluster centres $\mathrm{SC}_0 = \{\mathrm{C_j}(0)\}$ and the value of ε. Set $\mathrm{p} = 1$.

Step 2: Update the membership

Given the set of cluster centres SC_p, compute d_{ij} for $i = 1\ to\ N$ and $j = 1\ to\ k$. The memberships $u_{i,j}$ are updated using the following formula:

$$u_{i,j} = \left(\left(d_{ij}\right)^{1/m-1} \sum_{l=1}^{k} \left(\frac{1}{d_{il}}\right)^{1/m-1} \right)^{-1}$$

Equation 77

If $d_{ij} < \eta$, set $u_{i,j} = 1$, where η is a very small positive number.

Step 3: Compute the centre for each cluster

The centre for each cluster is computed using the following equation to arrive at a new set of cluster centres SC_{p+1}.

$$C_j(p) = \frac{\sum_{i=1}^{N} u_{ij}^{m} X_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Equation 78

Step 4: Iteration step

Given $\varepsilon > 0$ is a very small positive number, if $\left\| C_j(p) - C_j(p-1) \right\| < \varepsilon\ for\ j = 1\ to\ k,$ then stop. Otherwise set $p + 1 \rightarrow p$ and go to step 2.

7.3. Conclusion

In this chapter, two algorithms to identify spammer groups were presented. The first method makes use of ballot stuffing and bad mouthing reviews detected using the methods proposed in Chapter 5 and Chapter 6, and then uses K-means clustering to find the group to which these reviews belong. The second method makes use of ballot stuffing and bad mouthing reviews detected using the methods proposed in Chapter 5 and Chapter 6, and then applies fuzzy K-means clustering to find all the groups to which these reviews belong.

7.4. References

[1] Chang, C. T., Lai, J. Z. & Jeng, M. D. 2011, ‘A fuzzy K-means clustering algorithm using cluster center displacement’, *Journal of Information Science and Engineering*, vol. 27, no. 3, pp. 995-1009.

[2] Jain, A.K., Dubes, R.C. 1988, *Algorithms for Clustering Data*, Prentice Hall.

[3] Ott, M., Choi, Y., Cardie, C. & Hancock, J. T. 2011, ‘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, Association for Computational Linguistics*, vol. 1, pp. 309-19.

CHAPTER 8:
GRAPH-BASED METHOD FOR DETECTING CLIQUES

8.1. Introduction

In this chapter, we present the solution to identify cliques, which involves nine steps using a graph-based method. Based on our definition of a clique in the context of incorrect reviews in Chapter 2, a clique is a group of agents who work together to promote their products. Agents in a clique form a review circle as one agent will receive a review from another agent and will also provide a review on a different agent. This means the definition of a clique is similar to a directed and weighted graph. Therefore, a graph-based method is taken into account to identify cliques in the context of cloud reviews.

The clique in the context of incorrect reviews has similar characteristics to a directed and weighted graph where each dense subgraph has a high probability of becoming a clique. Each node in a review graph is a provider and two providers have connections when they write good reviews for each other. If a provider writes a good review for another provider, there is a direction between these two provider nodes. In this context where a directed and weighted graph is used, there are two different connections if two providers write good reviews for each other. In addition to direction, weight is another important factor in this algorithm because it represents the strength of the connection between two nodes. In the context of this research, each provider can write many reviews for the other provider with different star ratings. Therefore, the weight in the review graph will be the medium star rating the providers give each other.

8.2. Solution to identify cliques

Our proposed solution comprises the following nine steps.

8.2.1. Step 1: Check whether a review is ballot stuffing, then calculate the total number of stars in all the reviews

In order to identify cliques in reputation systems, we first check whether a review is ballot stuffing. Then, we calculate the weight in the weighted directed graph using Equation 79. Weight is an important factor in this algorithm because it represents the strength of the connection between two nodes. In the context of this research, each provider can write many reviews for the other provider with different ratings. Therefore, the weight in the review graph will be the medium star rating the providers give each other, which means that the higher the rating, the stronger their connection.

$$weight = \frac{1}{TotalNoOfStars}$$

Equation 79

To identify cliques, we take into account how to find a subgraph in a weighted directed graph. We make use of the method by Günnemann and Seidl (2010) to identify all subgraphs in a weighted directed graph to find the cliques. The number of stars in all the reviews where *s* gives d_1, *s* gives d_2, d_1 gives d_3, d_1 gives d_2, d_2 gives d_4 are 1, 10, 2, 3, 11 respectively. Figure 13 illustrates the number of reviews that one person gives another.

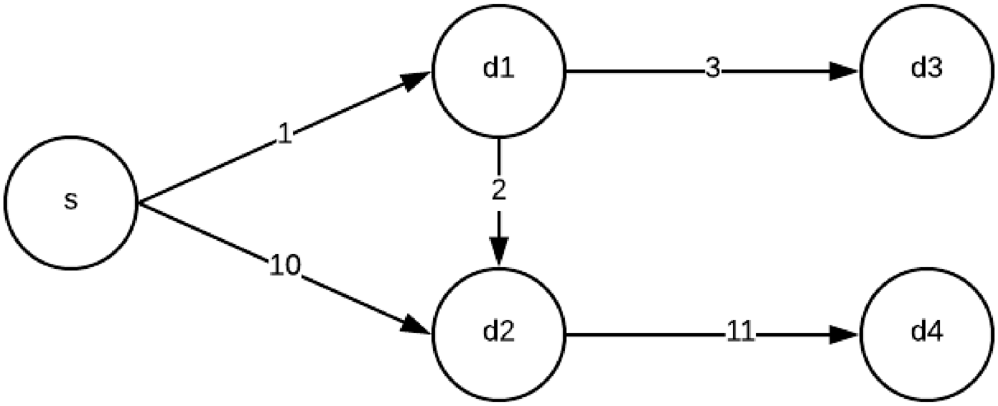


Figure 13: Clique with numbers of reviews

8.2.2. Step 2: Find the shortest path distance between two nodes and calculate the influence of each node

The shortest path distance between two nodes *s* and *d* is used in this approach as a representative for a graph-based distance function. There may be more than one path between the source and the destination. The distance between the source and the destination is calculated based on the sum of weights of all edges in between. In this step, we need to find the shortest path distance between the source and the destination nodes. Given a weighted graph *G = (V, E, w)* in which *G* represents the graph and *V, E* and *w* represent the vertices, edges and weight respectively, using the Dijkstra algorithm, the shortest path distance node s and d is defined as

$$d_{min-path}(s,d) = \underset{path<p_1,\ldots,p_n>}{min}\{\sum_{i=1}^{n-1} w(p_i, p_{i+1} | s \wedge p_n = d\}$$

Equation 80

The influence based on the Epanechnikov kernel is defined by

$$influence(s,d) = W\left(\frac{d_{min-path}(s,d)}{h}\right) \; with \; W(x) = \begin{cases} \frac{3}{4}(1-x^2), & |x| < 0 \\ 0, & else \end{cases}$$

Equation 81

where *s* is the source node,

d is the destination node,

h is the maximum value of the edge weight which is defined by the minimum star rating median one provider gives to the other to be considered as belonging to a clique. In other words, the action of giving a star rating to the other provider which is too low is not considered as creating a fake review to inflate that provider, therefore it cannot be in a clique. The minimum star rating allowed can be defined depending on the context of the dataset and the range of the star rating.

As all graphs are directed, the influence value of *(s, d)* may be different from the influence of *(d, s)*.

For example, we can see in Figure 13:

$$d_{min-path}(s, d_1) = 1$$

$$d_{min-path}(s, d_2) = 3$$

$$d_{min-path}(s, d_3) = 4$$

$$d_{min-path}(s, d_4) = 14$$

$influence(x, y) > 0$ iff $d_{min-path}(x, y) < 5$

8.2.3. Step 3: Define the influence region and direct region for each node

The direct region of a node is a collection of nodes directly giving reviews for that node. The influence region of node *s* is illustrated in Equation 82.

$$d \in direct(s) \Leftrightarrow (s, d) \in E \wedge w(s, d) < h$$

Equation 82

The influence region of a node is the extension of the direct region because not only is the direct level considered, but also the nodes with indirect connections to that node are included if the path's distance is smaller than h.

$$d \in region(s) \Leftrightarrow influence(s, d) > 0$$

Equation 83

In Figure 13, the direct region is: $region(s) = \{s, d_1, d_2, d_3\}, direct(s) = \{d_1\}$

s can reach all nodes in its influence region

8.2.4. Step 4: Calculate the density of each node

In this context, the density of each node demonstrates how many reviews the providers received from others and how highly that provider is rated. The density of a node is calculated in Equation 84.

$$density(d) = \sum_{s \in revRegion(d)} influence(s, d)$$

Equation 84

where $revRegion(d)$ is the reverse influence region of node d containing all nodes whose influence region contains d.

The reverse influence region of node d is a set of nodes that have an effect on node d, in opposition to the nodes that d itself influences. The reverse region is calculated as shown in Equation 85.

$$revRegion(d) = \{s \in V | d \in region(s)\}$$

Equation 85

In Figure 13, the reverse region is: $revRegion(d_2) = \{s, d_1, d_2\}$

8.2.5. Step 5: Identify and select all core nodes from node list

A core node must have density value greater than or equal to a minimum density τ, which means that the provider must receive reviews from at least a number of other providers with a minimum star rating.

The set of core nodes is calculated as shown in Equation 86 below:

$$coreNodes = \{v \in V | density(v) \geq \tau\}$$

Equation 86

8.2.6. Step 6: Cluster the list of core nodes into separated subsets

These clusters are identified based on the direct region of each node. If a provider belongs to a core cluster, all providers reviewing for that provider and all providers being reviewed by it will also belong to that cluster. Due to this characteristic, all of these clusters are separated from each other. This type of cluster is a subset $C \subseteq coreNodes$ defined as shown in Equation 87:

$$C\ is\ a\ subset\ of\ the\ list\ of\ the\ core\ nodes\ iff$$

$$\forall v \in C : \forall s \in coreNodes : s \in direct(v) \vee v \in direct(s) \Rightarrow s \in C$$

$$and\ C\ is\ a\ minimal\ among\ these\ sets$$

Equation 87

8.2.7. Step 7: Identify cliques – strong core clusters

In a core cluster, if there is a path such as $s \rightarrow d_1 \rightarrow d_3$ in which d_1 is in the direct region of s, this path is called a core path from s to d_3. As the graph is directed, this does not mean that if there is a path from s to d_3, there will be a path from d_3 to s. In our context, strong core clusters are cliques because in order to become a strong core cluster, it must fulfil the core path property which means that for all pairs of providers (s, d) in the strong core, there is not only a core path from s to d but also another reverse path from d to s. That is, the providers will review each other and form a review circle.

The core path property is defined as shown in Equation 88.

$$The\ core\ C\ fulfils\ the\ core\ path\ property\ iff\ \forall s, d \in C: corePath_C(s, d) = TRUE$$

$$corePath_C(s, d) = TRUE \Leftrightarrow \exists v_1, \dots, v_n \in C: v_1 = s \wedge v_n = d \wedge v_{i+1} \in direct(v_i) \wedge v_i \in coreNodes$$

Equation 88

A non-empty subset $C \subseteq coreNodes$ is a strong core cluster iff C fulfils the core path property and C is maximal with respect to the core path property (there is no $C' \supset C$ that fulfils also the core path property).

$$A\ subset\ C \subseteq coreNodes\ is\ a\ strong\ core\ cluster\ iff\ \forall s, d \in C: corePath_C(s, d) = TRUE \wedge corePath_C(d, s) = TRUE\ and\ C\ is\ maximal$$

Equation 89

8.2.8. Step 8: Identify semi-strong core clusters

A semi-strong core cluster does not have bi-directional paths between each pair of providers in the cluster, but only requires at least a path between each pair. In our context, semi-strong clusters are similar to any providers with interactions with any of the actors in a clique. Therefore, the providers in a semi-strong cluster will have less involvement in spam review activities than the providers in a strong cluster.

A non-empty subset $C \subseteq coreNodes$ is a strong core cluster iff $iff\ \forall s, d \in C: corePath_C(s, d) = TRUE \vee corePath_C(d, s) = TRUE$ and C is maximal with respect to the core path property (there is no $C' \supset C$ that fulfils also the core path property).

$$A\ subset\ C \subseteq coreNodes\ is\ a\ semi - strong\ core\ cluster\ iff\ \forall s, d \in C{:}\ corePath_C(s,d) = TRUE \vee corePath_C(d,s) = TRUE\ and\ C\ is\ maximal$$

Equation 90

8.2.9. Step 9: Calculate providers' reputation based on detected strong core clusters (cliques) and semi-strong core clusters

Initially, the reputation mark of each provider is initialized by 100. If a provider is identified as a clique, its reputation is subtracted by 70%. On the other hand, as mentioned in Step 7, providers in semi-strong core clusters have a lower risk of participating in writing incorrect reviews, so the reputation marks of each of these providers is subtracted by 30%. Reputation is one of the criteria in searching for service providers, which means that the providers with a higher reputation have a higher priority in the search result list.

8.3. Conclusion

This chapter presented a method to identify cliques using a graph-based method. This algorithm comprises nine steps to identify the strong core clusters and semi-strong core clusters. The strong core clusters are considered to be cliques in our context and semi-strong core clusters are considered as clusters which are less likely to become cliques. Therefore, in nine steps, we present a method to calculate the providers' reputation scores based on the detected strong core clusters and semi-strong core clusters.

8.4. References

[1] G¨unnemann, S. & Seidl, T. 2010, 'Subgraph Mining on Directed and Weighted Graphs', *the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010),* Springer - Heidelberg, Germany, pp. 133-46.

CHAPTER 9:
VALIDATION AND VERIFICATION

9.1. Introduction

In this chapter, the solution for research question 5 is presented. Section 9.2 overviews the solution implementation which includes an overview of the solution implementation of the method to identify ballot stuffing, bad mouthing and spammer groups. Section 9.3 provide more detail on the solution implementation for the methods to identify ballot stuffing, bad mouthing, and spammer groups. Due to the restriction of time to undertake this thesis, we did not validate the intelligent method to identify cliques. Section 9.4 concludes the chapter.

9.2. Overview of Solution Implementation

9.2.1. Overview of solution implementation of the method to identify ballot stuffing

In order to validate the method developed to identify ballot stuffing, we compare our method with BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. We conduct an experiment with different percentages of ballot stuffing injected. Moreover, we also compare our method with the three existing methods using FPR (false positive rate) and FNR (false negative rate) to prove which method is has a majority capability (the ability to identify ballot stuffing when the majority of reviews are ballot stuffing). Furthermore, in order to test the burstiness capability of our method and other existing methods, we conduct an experiment on one random agent in a period of time (different percentages of ballot stuffing reviews in different periods of time).

9.2.2. Overview of solution implementation of the method to identify bad mouthing

In order to validate the method developed to identify bad mouthing, we compare our method with BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. We conduct an experiment with different percentages of bad mouthing injected. Moreover, we also compare our method with the three existing methods using FPR (false positive rate) and FNR (false negative rate) to prove which method has a majority capability (the ability to identify bad mouthing when the majority of reviews are bad mouthing). Furthermore, in order to test the burstiness capability of our method and other existing methods, we conduct an experiment on one random agent in a period of time (different percentages of bad mouthing reviews in different periods of time).

9.2.3. Overview of solution implementation of the method to identify spammer groups

In order to validate the methods developed to identify spammer groups, we compare our two proposed methods: one uses K-means clustering and the other uses fuzzy K-means clustering. One method of validation is to prove that when K-means clustering is used, we can find one spammer group for each spam review, however, when fuzzy K-means clustering is used, more than one group for each spam review is identified. Another method of validation is to run our two methods to find all the spammer groups in different periods of time, then to compare our two methods using the paired t-test algorithm.

9.3. Solution implementation

9.3.1. Solution implementation for the algorithm to identify ballot stuffing

In this section, we conduct two experiments to compare our method and the other existing methods regarding their capability to solve the problem of advisors providing thousands of ratings within a short period of time and their ability to identify ballot stuffing when there is a majority of ballot stuffing in the whole large dataset. We also measure the performance of our method and other methods using the following performance measurement.

Performance measurement

To measure the performance of the approach to detect ballot stuffing, we measure its ability to detect ballot stuffing reviews. A good approach should be able to identify dishonest reviews. This performance can be measured by precision, recall and accuracy (David & Goadrich, 2006).

	Actual positive	Actual negative
Predicted positive	True positive	False positive

Predicted negative	False negative	True negative

Table 2: Confusion matrix

$$Recall = \frac{TP}{TP + FN}$$

Equation 91

$$Precision = \frac{TP}{TP + FP}$$

Equation 92

$$True\ positive\ rate = \frac{TP}{TP + FN}$$

Equation 93

$$False\ negative\ rate = \frac{FP}{FP + TN}$$

Equation 94

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 95

True positives are examples correctly labelled as positive.

False positives are examples incorrectly labelled as positive.

True negatives are examples correctly labelled as negative.

False positives are examples incorrectly labelled as negative.

Recall is the true positive rate, which is calculated in Equation 91.

Precision measures the fraction of examples classified as positive that are truly positive, which is calculated in Equation 92.

The true positive rate measures the fraction of positive examples that are correctly labelled, which is calculated in Equation 93.

The false negative rate measures the fraction of negative examples that are misclassified as positive, which is calculated in Equation 94.

Accuracy measures the accuracy in identifying true positives and true negatives, which is calculated in Equation 95.

Experiment results and analysis

Experiment 1: Overall performance and majority capability

Accuracy, precision and recall are the three metrics that are used to compare the overall performance. To compare the overall performance, we conducted an experiment on a dataset which includes 15,000 reviews, in which the percentage of ballot stuffing reviews is 10%, 20%, 50% and 80% respectively. We calculate the number of identified reviews, the number of true positives, false positives, true negatives, false negatives and then measure the accuracy, precision and recall of BRS, TRAVOS, iCLUB and our method.

Steps:

- Step 1: An admin agent bootstraps the set of all the reviews for all agents. The set of all the reviews in this experiment will contain 15,000 reviews which are crawled from the website www.getapp.com. The admin agent knows that all the ballot stuffing reviews in the dataset have been bootstrapped. It maintains a count of the number of ballot stuffing reviews in the dataset.
- Step 2: The percentage of ballot stuffing in the experiment is 10%, 20%, 50% and 80% respectively. To be specific, we divide them into 4 test cases as follows:
    - Test case 1: there are 15,000 reviews in which 1,500 reviews are ballot stuffing.
    - Test case 2: there are 15,000 reviews in which 3,000 reviews are ballot stuffing.
    - Test case 3: there are 15,000 reviews in which 7,500 reviews are ballot stuffing.
    - Test case 4: there are 15,000 reviews in which 12,000 reviews are ballot stuffing.
- Step 3: BRS, TRAVOS, iCLUB and our method are run to identify all the ballot stuffing reviews.
- Step 4: The number of identified reviews and the number of true positives, false positives, true negatives and false negatives are calculated.

- Step 5.1: The accuracy, precision and recall of BRS, TRAVOS, iCLUB and our method are measured to determine which method is the most accurate.

To check the majority capability, we also undertake the following additional step:

- Step 5.2: The false positive rate and the false negative rate are measured to see which one is less affected when the majority of reviews are ballot stuffing.

Figure 14 shows the result of accuracy for each method when the number of ballot stuffing reviews is 1500, 3000, 7500 and 12000, respectively. As we applied the network-based method which combines the suspicion score of reviewers, reviews and products, it should be much more accurate than other methods which do not consider the relationship between reviewers, reviews and products. As shown in Figure 14, the accuracy of our method is better than BRS, TRAVOS and iCLUB when tested with different percentages of incorrect reviews. When the percentage of ballot stuffing is 10%, 20% and 80%, we can see that the accuracy of our method is the highest while the accuracy of BRS is the lowest. If half of the reviews are ballot stuffing, the accuracy of BRS and iCLUB is slightly higher than TRAVOS. In this case, the accuracy of our method is 0.7094 and it is still the highest compared to the other three methods. When there are 12000 ballot stuffing reviews, the accuracy of our method is noticeably higher than BRS, with a difference of more than 0.11.

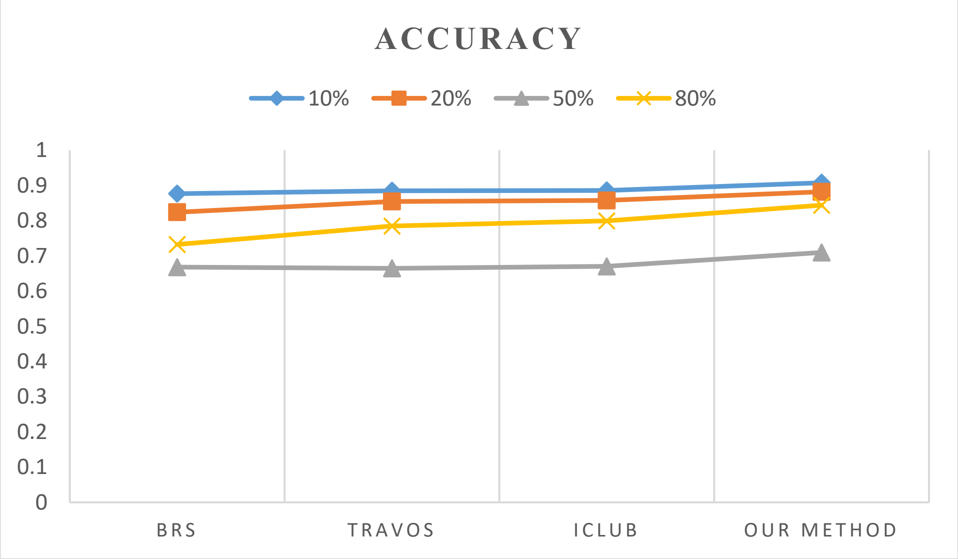


Figure 14: Accuracy of BRS, TRAVOS, iCLUB and our method with different percentages of ballot stuffing

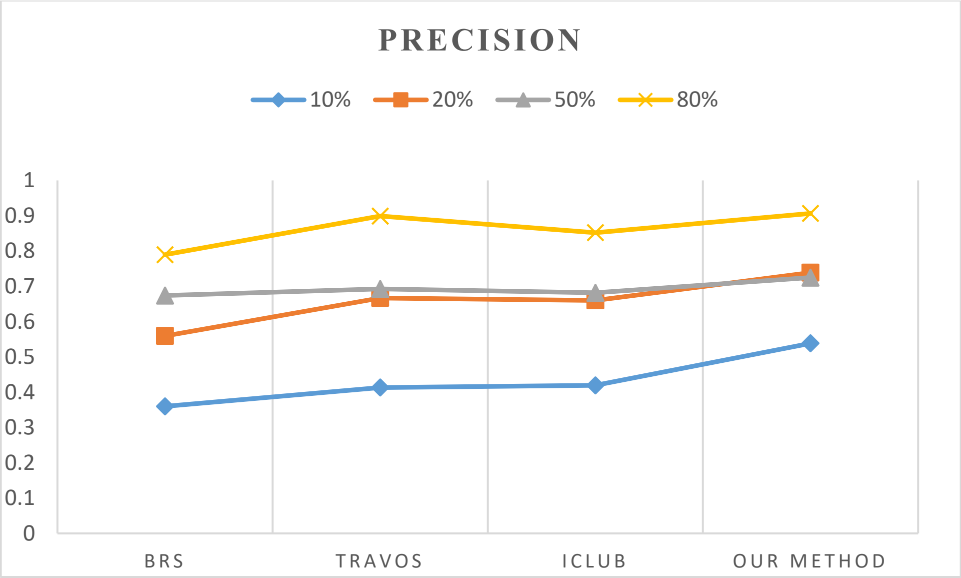


Figure 15: Precision of BRS, TRAVOS, iCLUB and our method with different percentages of ballot stuffing

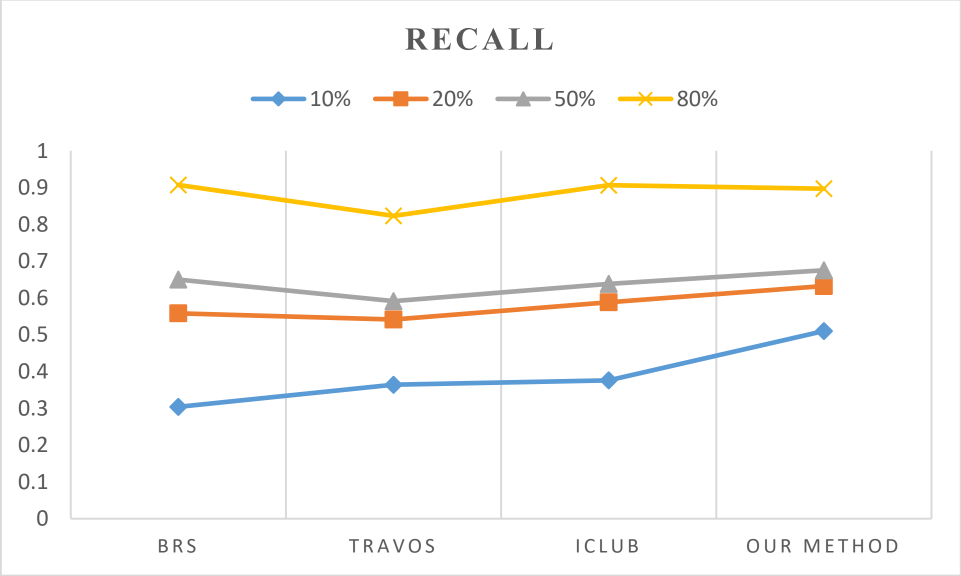


Figure 16: Recall of BRS, TRAVOS, iCLUB and our method with different percentages of ballot stuffing

We also measure the precision and recall of BRS, TRAVOS, iCLUB and our method when the number of ballot stuffing reviews is 10%, 20%, 50% and 80% of the total number of reviews. We can see in Figure 15 and Figure 16 that our method has the highest precision and highest recall compared to the others when the percentage of ballot stuffing is 10%, 20% or 50% which means that our method is very good in most cases, however, it is very picky because it does not think many reviews are ballot stuffing, but despite this, it is still able to accurately identify most of the ballot stuffing reviews. When the percentage of ballot stuffing reviews is 80%, the precision of BRS and iCLUB is higher but the recall of BRS and iCLUB is lower than our method. This means that BRS and iCLUB can detect more ballot stuffing that is actually ballot stuffing and more “not ballot stuffing” that is actually not ballot stuffing. In all cases, BRS has the lowest precision and our method has the highest precision. When 20% and 50% of the reviews are ballot stuffing, the precision of TRAVOS and iCLUB is slightly different. When there is 80% ballot stuffing, our method has the highest precision with nearly 0.91, TRAVOS has the second highest with nearly 0.90, iCLUB’s precision is third with 0.85 and BRS’s precision is the lowest with approximately 0.79. In Figure 16, the recall of our method is the highest in most cases, except when 80% of the reviews are ballot stuffing. When there is 10% or 80% ballot stuffing, the recall of BRS is the lowest, while the recall of TRAVOS is the lowest when 20% or 50% of the reviews are ballot stuffing. However, as shown in Figure 14, our method is still the most accurate compared to BRS, TRAVOS and iCLUB in all situations.

Other results in Figure 17 and Figure 18 show that FPR of BRS and iCLUB are increasing and approaching 1 while the FNR of BRS and iCLUB are decreasing and approaching 0, especially when the majority of the reviews are incorrect. This means if the majority of the reviews are ballot stuffing, BRS and iCLUB tend to identify every review as ballot stuffing. As the FPR of BRS is closer to 1 and the FNR of BRS is closer to 0 than the three other methods, we can conclude that BRS is most affected when the majority of the reviews are ballot stuffing. In Figure 19 and Figure 20, we can see that TRAVOS and our method are not affected much when more than half of the reviews are ballot stuffing. As shown in Figure 21, TRAVOS is slightly less affected by the majority capability than our method. However, our method is still the best in all cases when considering accuracy, as shown in Figure 14.

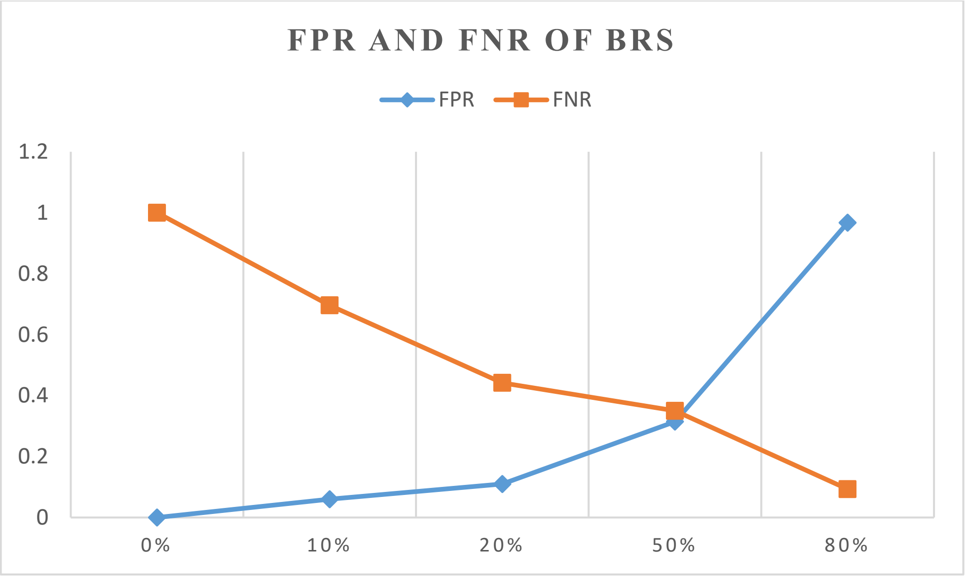


Figure 17: FPR and FNR of BRS with different percentages of ballot stuffing

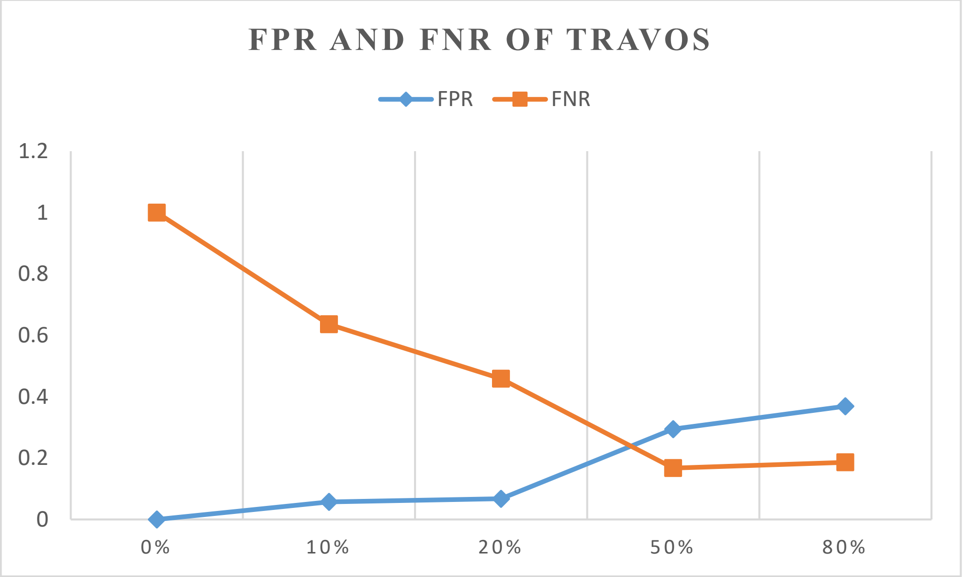


Figure 18: FPR and FNR of TRAVOS with different percentages of ballot stuffing

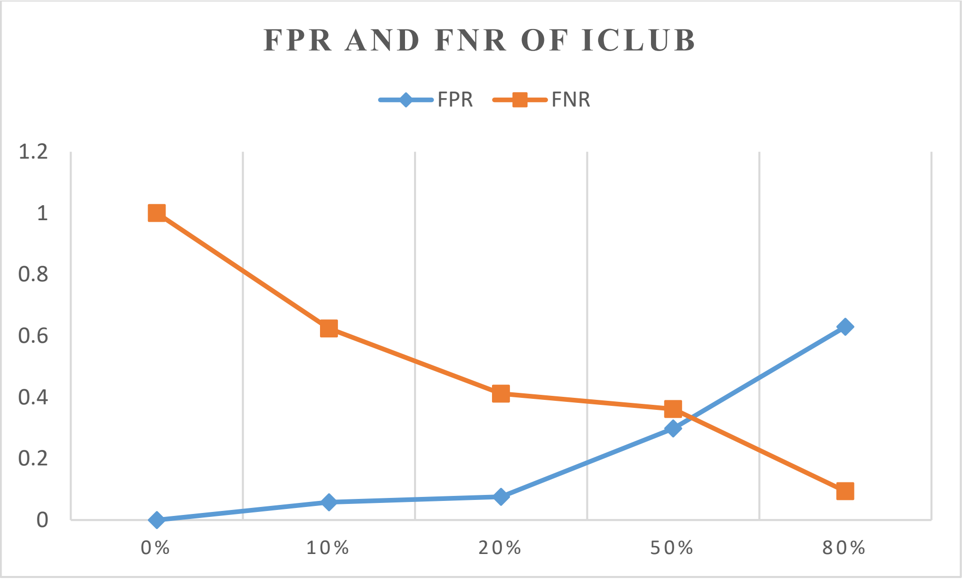


Figure 19: FPR and FNR of iCLUB with different percentages of ballot stuffing

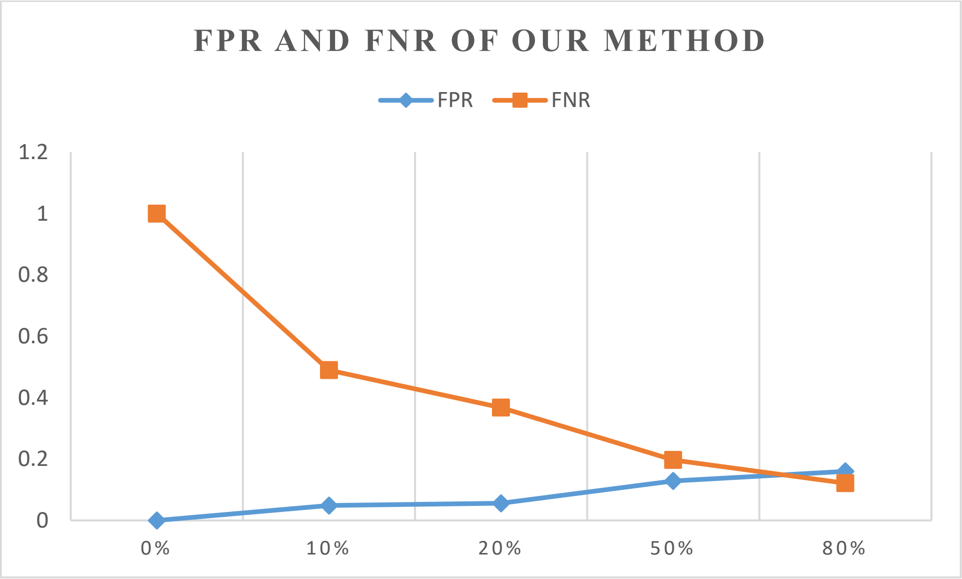


Figure 20: FPR and FNR of our method with different percentages of ballot stuffing

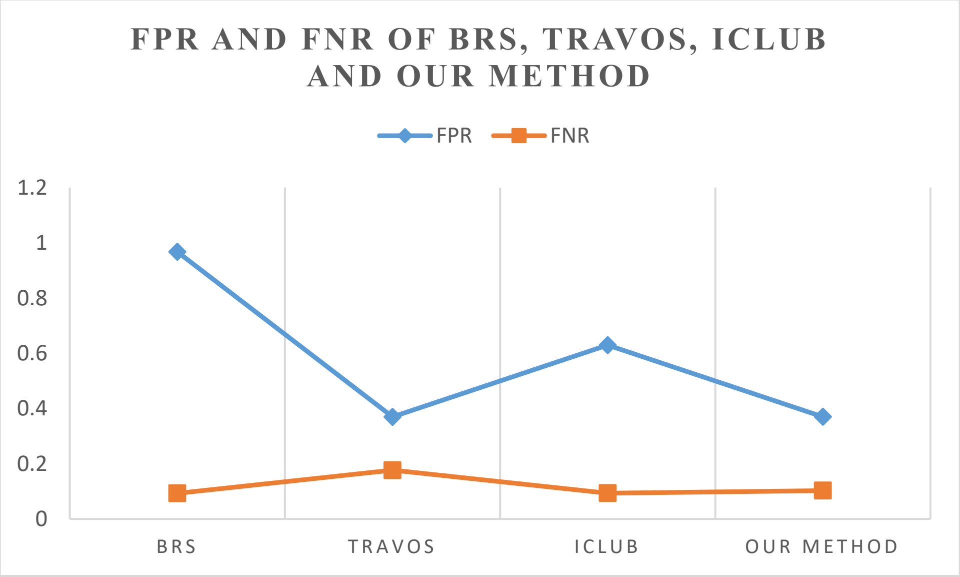


Figure 21: FPR and FNR of BRS, TRAVOS, iCLUB and our method when the majority are ballot stuffing (80%)

Experiment 2: Burstiness capability

In this experiment, we check the burstiness capability and how it affects the reputation of an advisor agent.

Burstiness is the ability of a method to overcome a situation in which an advisor agent provides a large number of ballot stuffing reviews in a short period of time.

In this experiment, there are 2,000 reviews, 5,000 reviews, 10,000 reviews, and 15,000 reviews with a percentage of ballot stuffing reviews being 10%, 20%, 50% and 80% respectively, bursting in a period of 60 days (for example, in the first period of 10 days, there are 2000 reviews with 10% ballot stuffing reviews, in the next period of 20 days, there are 3,000 more reviews with 20% ballot stuffing reviews, in the next period of 20 days, there are 5,000 more reviews with 50% ballot stuffing reviews, in the next period of 10 days, there are 5,000 more reviews with 80% ballot stuffing reviews).

In order to check the burstiness capability, we conduct an experiment which comprises the following twenty one steps:

- Step 1: An admin agent bootstraps the set of all the reviews for all agents in a period of 40 days. The set of all the reviews in our framework contain 15,000 reviews which are crawled from the website https://www.getapp.com. The admin agent knows all the ballot stuffing reviews in the dataset that it bootstrapped. It maintains a count of the number of ballot stuffing reviews in the dataset.

- Step 2: Pick any random agent X. In this experiment, we pick agent Box with 330 reviews. Calculate the total number of reviews for agent Box. Then, the ballot stuffing reviews for agent Box will be injected in four test cases as follows:
    - Test case 1: 10% ballot stuffing reviews in 10 days.
    - Test case 2: 20% ballot stuffing reviews in the next 10 days.
    - Test case 3: 50% ballot stuffing reviews in the next 10 days.
    - Test case 4: 80% ballot stuffing reviews in the next 10 days.
- Step 3: Calculate the reputation of agent Box. The computed value will include ballot stuffing values. The reputation for agent Box at the beginning is set to be A_0=0.
- Step 4: Apply our algorithm to identify the ballot stuffing values.
- Step 5: Calculate the percentage of the ballot stuffing reviews of agent Box that were filtered out and not taken into account during the reputation calculation due to our algorithm. Compute the following:
    - percentage of ballot stuffing reviews filtered out.
    - percentage of ballot stuffing reviews not filtered out.
    - compute true positives, false positives, true negatives, false negatives.
    - Compute accuracy.
- Step 6: Calculate the reputation of agent Box again. Let this value be A_i. This value is calculated using the following formula:

$$A_1 = A_0 + Accuracy_1 * TP_1/NumberOfReviews * 100$$

$$A_2 = A_1 + Accuracy_2 * (TP_2 - TP_1)/NumberOfReviews * 100$$

$$A_3 = A_2 + Accuracy_3 * (TP_3 - TP_2)/NumberOfReviews * 100$$

$$A_4 = A_3 + Accuracy_4 * (TP_4 - TP_3)/NumberOfReviews * 100$$

$$A_i = A_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

where A_1: reputation of agent Box after 10 days,

A_2: reputation of agent Box after 20 days,

A_3: reputation of agent Box after 30 days,

A_4: reputation of agent Box after 40 days.

- Step 7: The admin agent will also calculate the reputation value of agent Box again. Let this value be B_1.

$$B_1 = A_0 + \frac{NumberOfBallotStuffingReviews}{NumberOfReviews} * 100$$

- Step 8: Compute the degree of correlation between A_i and B_1. Let this value be CA_i.

$$CA_i = |(A_i - B_1)|, i = \overline{1,4}$$

- Step 9: Apply the BRS algorithm to identify ballot stuffing values.
- Step 10: Calculate the reputation of agent Box again. Let this value be D_i. The value is calculated using the formula:

$$D_i = D_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 11: The admin agent will also calculate the reputation value of agent Box again. Let this value be B_2.

$$B_2 = A_0 + \frac{NumberOfBallotStuffingReviews}{NumberOfReviews} * 100$$

- Step 12: Compute the degree of correlation between D_i and B_2. Let this value be CD_i.

$$CD_i = |(D_i - B_1)|, i = \overline{1,4}$$

- Step 13: Apply the TRAVOS algorithm to identify the ballot stuffing values.
- Step 14: Calculate the reputation of agent Box again. Let this value be E_i.

$$E_i = E_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 15: The admin agent will also calculate the reputation value of agent Box again. Let this value be B_3.

$$B_3 = A_0 + \frac{NumberOfBallotStuffingReviews}{NumberOfReviews} * 100$$

- Step 16: Compute the degree of correlation between A_3 and B_3. Let this value be CE_i.

$$CE_i = |(E_i - B_1)|, i = \overline{1,4}$$

- Step 17: Apply iCLUB algorithm to identify ballot stuffing values.
- Step 18: Calculate the reputation of agent Box again. Let this value be F_i.

$$F_i = F_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 19: The admin agent will also calculate the reputation value of agent Box again. Let this value be B_4.

$$B_4 = A_0 + \frac{NumberOfBallotStuffingReviews}{NumberOfReviews} * 100$$

- Step 20: Compute the degree of correlation between A_3 and B_3. Let this value be CF_i.

$$CF_i = |(F_i - B_1)|, i = \overline{1,4}$$

- Step 21: Compare CB_i, CD_i, CE_i and CF_i.

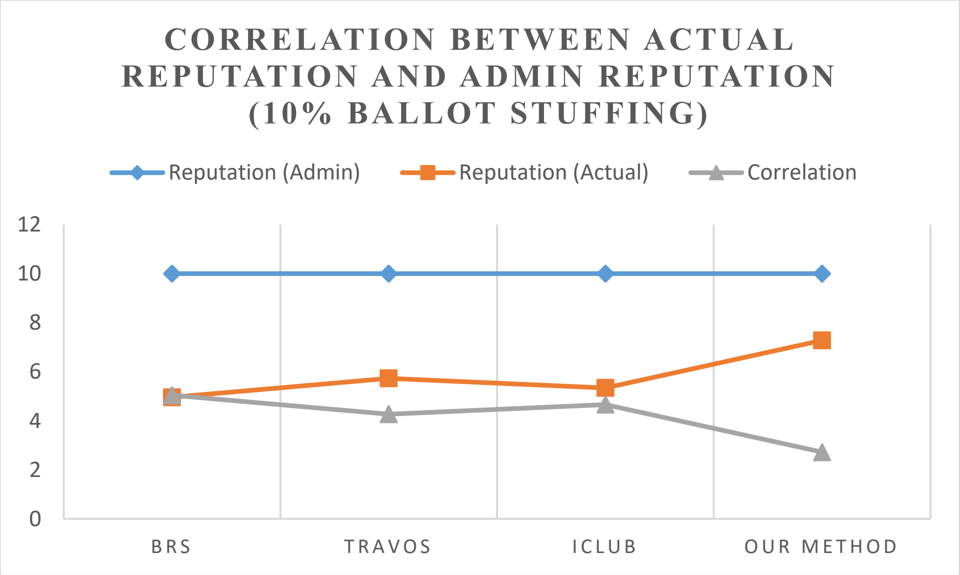


Figure 22: Correlation between actual reputation and admin reputation (10% ballot stuffing)

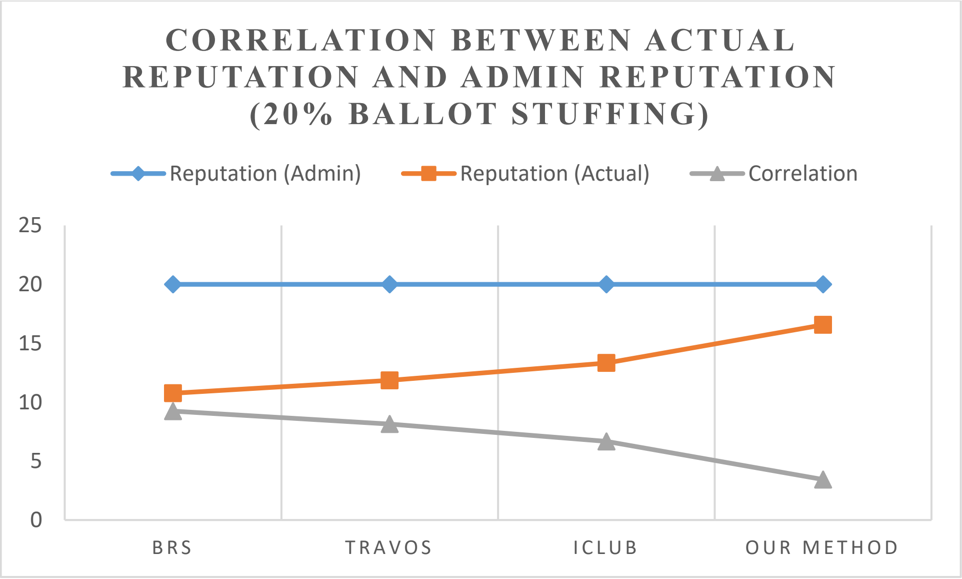


Figure 23: Correlation between actual reputation and admin reputation (20% ballot stuffing)

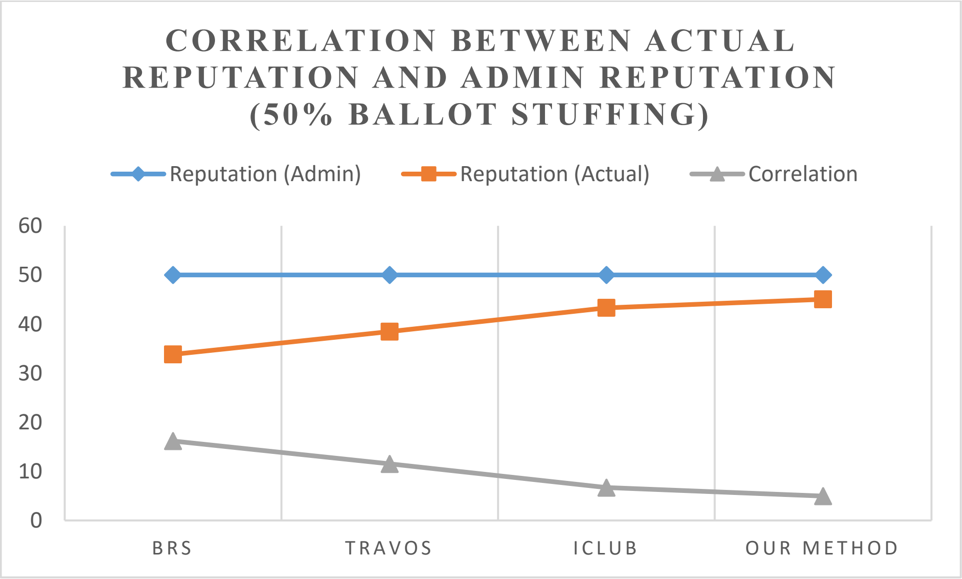


Figure 24: Correlation between actual reputation and admin reputation (50% ballot stuffing)

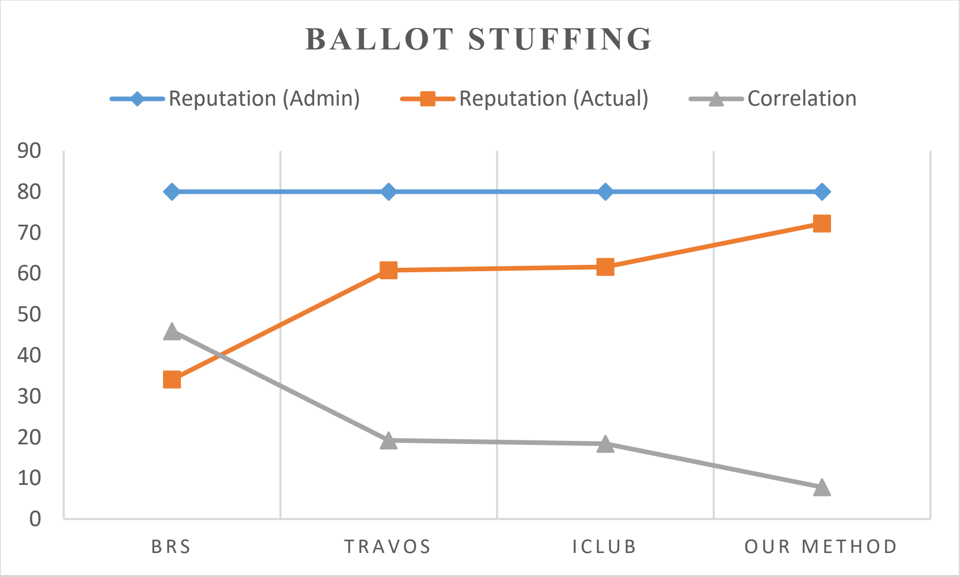


Figure 25: Correlation between actual reputation and admin reputation (80% ballot stuffing)

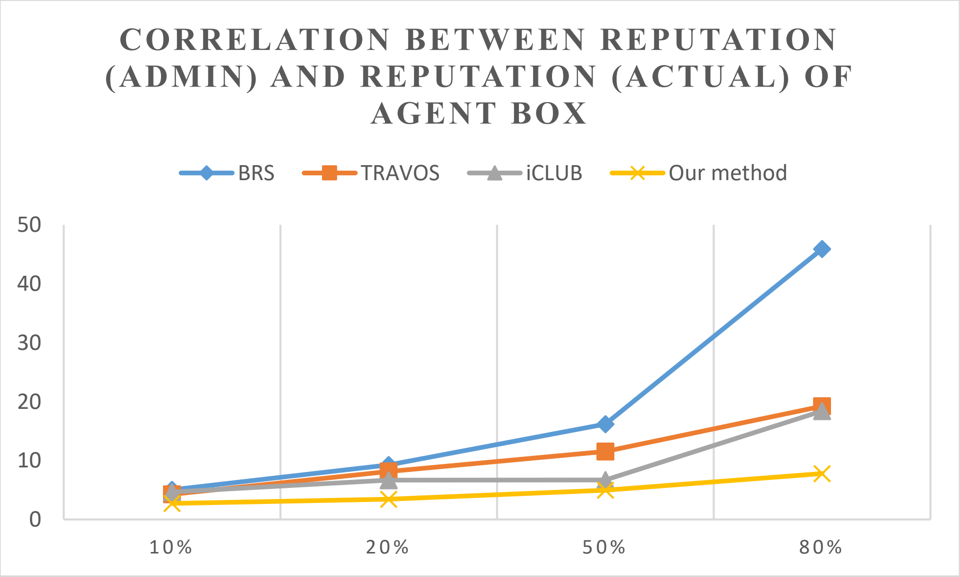


Figure 26: Correlation between reputation (admin) and reputation (actual) of agent Box

There is a slight increase when the percentage of ballot stuffing is 10% in the first 10 days, 20% in the next 10 days and 50% in the next 10 days. However, there is a dramatic change when the percentage of ballot stuffing is 80% in the last 10 days. This means that BRS is strongly affected by the burstiness capability.

The correlation between the reputation of admin and the actual reputation of our method is always the lowest when the percentage of ballot stuffing reviews is 10%, 20%, 50% or even 80%. This means that our method is the best at detecting ballot stuffing.

9.3.2. Solution implementation for the algorithm to identify bad mouthing

In this section, we conduct two experiments to compare our method and the other existing methods regarding their capability to solve the problem of advisors providing thousands of ratings within a short period of time and their ability to identify bad mouthing when there is a majority of bad mouthing in the whole large dataset. We also measure the performance of our method and the other methods using the following performance measurement.

Performance measurement

The performance of an approach to detect bad mouthing reviews can be measured by its ability to detect bad mouthing reviews. A good approach should be able to identify dishonest reviews. This performance can be measured by precision, recall and accuracy (David & Goadrich, 2006).

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Table 3: Confusion matrix

True positives are examples correctly labelled as positive.

False positives are examples incorrectly labelled as positive.

True negatives are examples correctly labelled as negative.

False positives are examples incorrectly labelled as negative.

Recall is the true positive rate, which is calculated in Equation 91.

Precision measures the fraction of examples classified as positive that are truly positive, which is calculated in Equation 92.

The true positive rate measures the fraction of positive examples that are correctly labelled, which is calculated in Equation 93.

The false negative rate measures the fraction of negative examples that are misclassified as positive, which is calculated in Equation 94.

Accuracy measures the accuracy in identifying true positives and true negatives, which is calculated in Equation 95

Experiment results and analysis

Experiment 1: Overall performance and majority capability

Accuracy, precision and recall are the three metrics that are used to compare overall performance by conducting an experiment on a dataset which includes 15,000 reviews, in which the percentage of bad mouthing reviews is 10%, 20%, 50% and 80%, respectively. We calculate the number of identified reviews, the number of true positives, false positives, true negatives and false negatives and then measure the accuracy, precision and recall of BRS, TRAVOS, iCLUB and our method.

Steps:

- Step 1: An admin agent bootstraps the set of all the review for all agents. The set of all the reviews in this experiment contain 15,000 reviews which are crawled from the website https://www.getapp.com. The admin agent knows all the bad mouthing reviews in the dataset that it bootstrapped. It maintains a count of the number of bad mouthing reviews in the dataset.
- Step 2: The percentage of bad mouthing in the experiment will be 10%, 20%, 50% and 80%, respectively. To be specific, we divide them into the four following test cases:
    - Test case 1: there are 15,000 reviews in which 1,500 reviews are bad mouthing.
    - Test case 2: there are 15,000 reviews in which 3,000 reviews are bad mouthing.
    - Test case 3: there are 15,000 reviews in which 7,500 reviews are bad mouthing.
    - Test case 4: there are 15,000 reviews in which 12,000 reviews are bad mouthing.

- Step 3: BRS, TRAVOS, iCLUB and our method are run to identify all the bad mouthing reviews.
- Step 4: Calculate the number of identified reviews, the number of true positives, false positives, true negatives, and false negatives.
- Step 5.1: Measure the accuracy, precision and recall of BRS, TRAVOS, iCLUB and our method to see which method is the most accurate.

To check the majority capability, we implement one more step as follows:

- Step 5.2: Measure the false positive rate and the false negative rate to see which one is less affected when the majority of reviews are bad mouthing.

Figure 27 shows the result for accuracy of each method when the percentage of bad mouthing reviews is 1500, 3000, 7500 and 12000, respectively. As we applied the network-based method which combines the suspicion score of reviewers, reviews and products, it should be much more accurate than the other methods which do not consider the relationship between reviewers, reviews and products. As shown in Figure 27, the accuracy of our method is better than BRS, TRAVOS and iCLUB when being tested with different percentages of incorrect reviews.

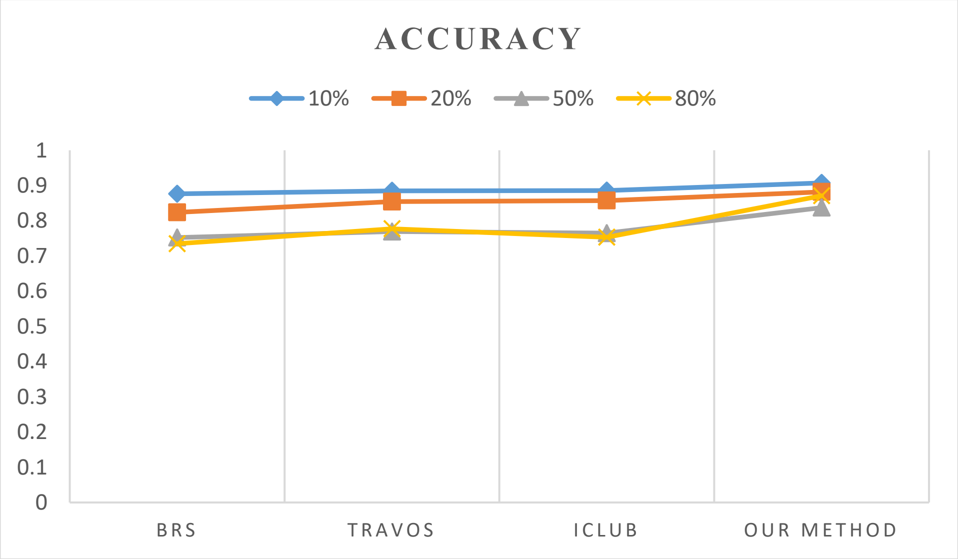


Figure 27: Accuracy of BRS, TRAVOS, iCLUB and our method with different percentages of bad mouthing

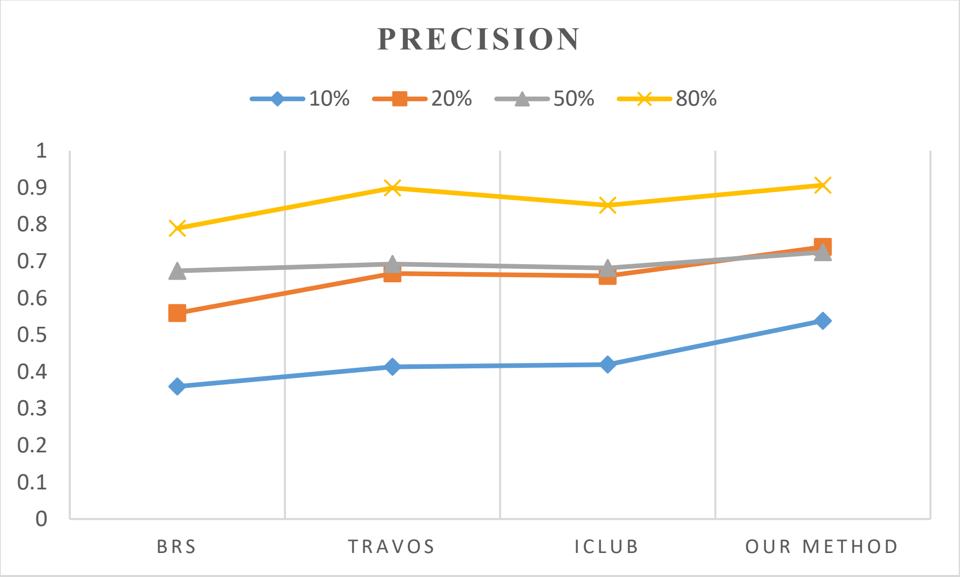


Figure 28: Precision of BRS, TRAVOS, iCLUB and our method with different percentages of bad mouthing

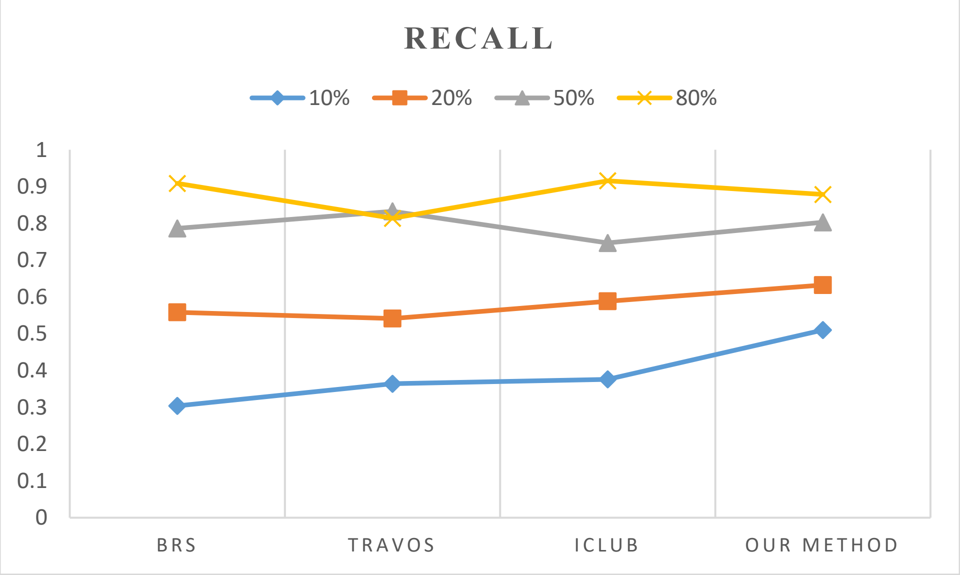


Figure 29: Recall of BRS, TRAVOS, iCLUB and our method with different percentages of bad mouthing

We also measure the precision and recall of BRS, TRAVOS, iCLUB and our method when the number of bad mouthing is 10%, 20%, 50% and 80% of the total number of reviews. We can see that our method has the highest precision and highest recall compared to the others in two cases when the percentage of bad mouthing reviews is 10% and 20% which means that our method is very good, however it is very picky as it does not think many reviews are bad mouthing, but despite this, it is still able to identify most of the bad mouthing reviews which are actually bad mouthing. When the percentage of bad mouthing reviews is 50%, the precision and recall of our method is higher than BRS and iCLUB. However, as shown in Figure 28 and Figure 29, the precision of our method is higher than TRAVOS but the recall of our method is lower than TRAVOS. This means that when comparing our method and TRAVOS, our method does not identify as many bad mouthing reviews as TRAVOS. All the images it identifies as bad mouthing are really bad mouthing. However, it also misses a lot of actual bad mouthing. When the percentage of bad mouthing reviews is 80%, the precision of BRS and iCLUB is lower but the recall is higher than our method. This means that our method identifies many reviews as bad mouthing which are not actually bad mouthing. However, it also identifies many reviews as bad mouthing which are bad mouthing. So, from our set of images, many were classified as bad mouthing, many of which were bad mouthing but many of which were not bad mouthing. However, our method is still the most accurate compared to BRS, TRAVOS and iCLUB in all situations.

Another result shows that FPR of BRS and iCLUB increases and approaches 1 while the FNR of BRS and iCLUB decreases and approaches 0, especially when the majority of the reviews are incorrect. This means if the majority of the reviews are bad mouthing, BRS and iCLUB tend to identify every review as bad mouthing. As the FPR of BRS is closer to 1 and the FNR of BRS is closer to 0 than the three other methods, we can conclude that BRS is most affected when the majority of the reviews are bad mouthing. As shown in Figure 34, TRAVOS is slightly less affected by the majority capability than our method. However, our method is still the best in all cases when considering accuracy, as we can see in Figure 27.

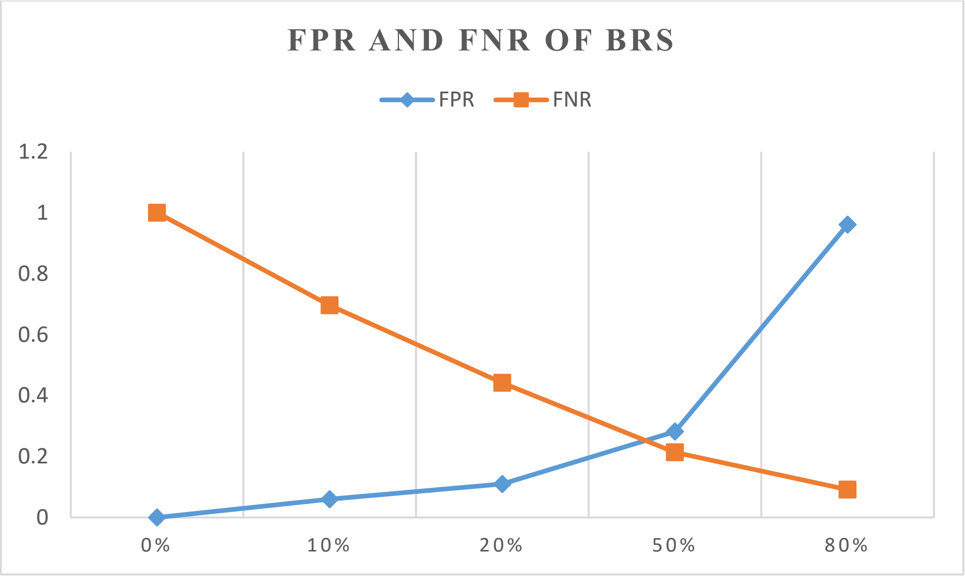


Figure 30: FPR and FNR of BRS with different percentages of bad mouthing

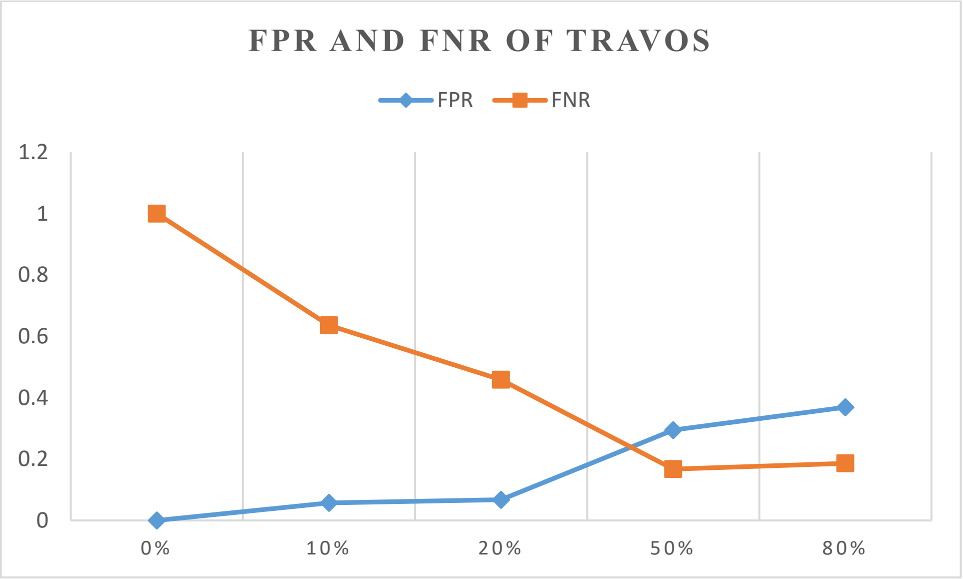


Figure 31: FPR and FNR of TRAVOS with different percentages of bad mouthing

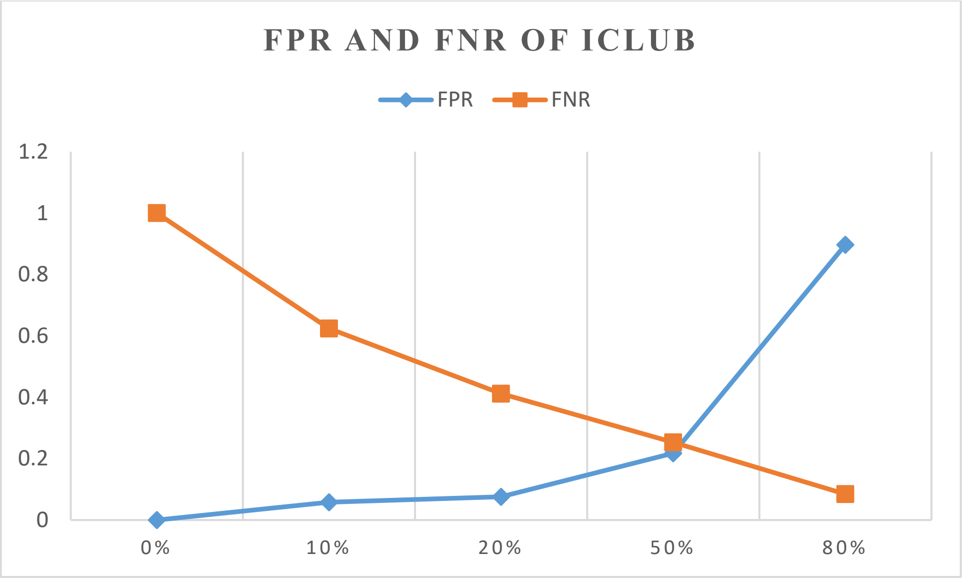


Figure 32: FPR and FNR of iCLUB with different percentages of bad mouthing

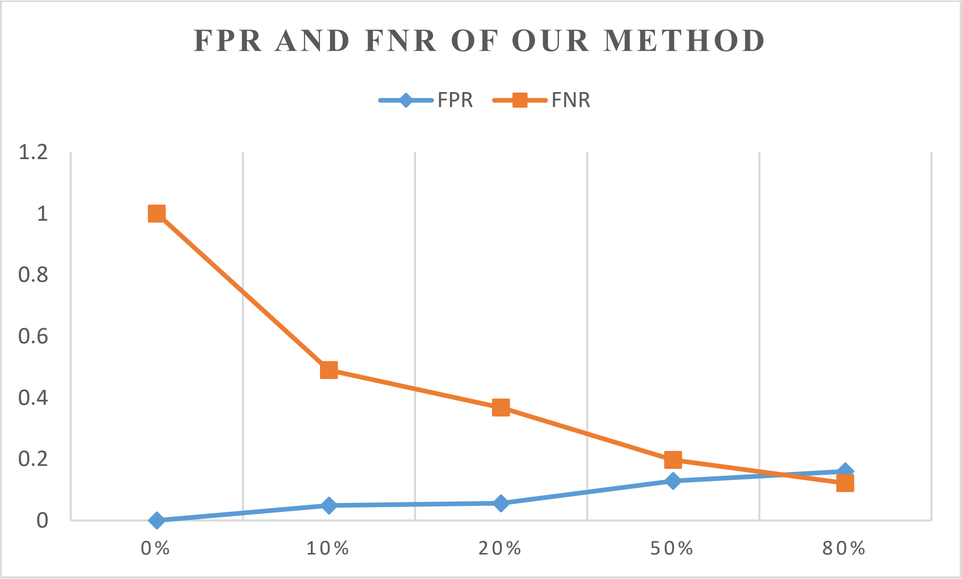


Figure 33: FPR and FNR of our method with different percentages of bad mouthing

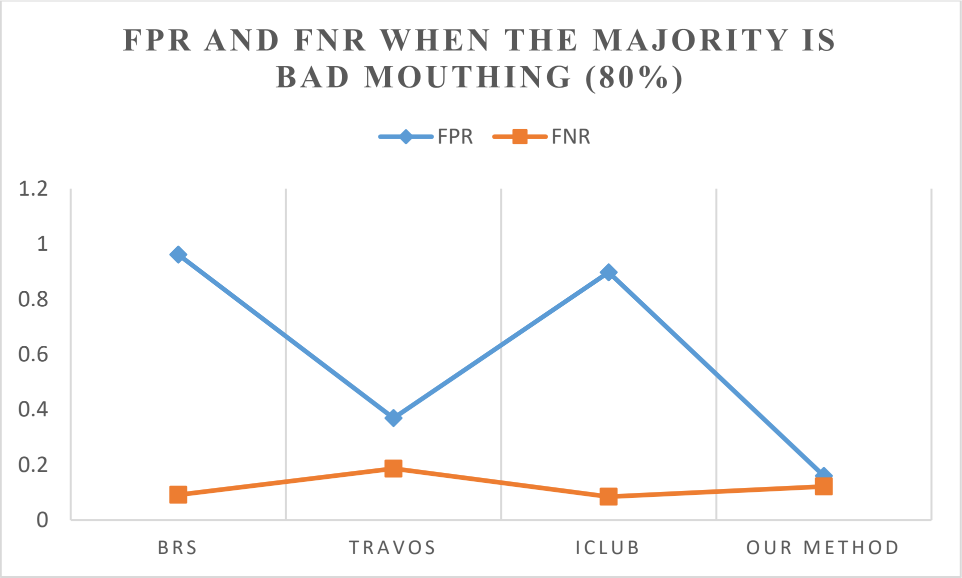


Figure 34: FPR and FNR of BRS, TRAVOS, iCLUB and our method when the majority is bad mouthing (80%)

Experiment 2: Burstiness capability

Burstiness capability and how it affects the reputation of an advisor agent.

Burstiness is the ability of a method to overcome a situation in which an advisor agent provides a large number of bad mouthing reviews in a short period of time.

In this experiment, the number of reviews are 2,000 reviews, 5,000 reviews, 10,000 reviews, and 15,000 reviews with the percentage of bad mouthing being 10%, 20%, 50% and 80%, respectively, bursting in a period of 60 days (for example, in the first period of 10 days, there are 2000 reviews with 10% bad mouthing reviews, in the next period of 20 days, there are 3,000 more reviews with 20% bad mouthing reviews, in the next period of 20 days, there are 5,000 more reviews with 50% bad mouthing reviews, in the next period of 10 days, there are 5,000 more reviews with 80% bad mouthing reviews).

In order to check the burstiness capability, we conduct an experiment which includes the following twenty one steps:

- Step 1: An admin agent bootstraps the set of all the reviews for all agents in a period of 40 days. The set of all the reviews in our framework will contain 15,000 reviews which are crawled from the website https://www.getapp.com. The admin agent knows all the bad mouthing reviews in the dataset that it bootstrapped. It maintains a count of the number of bad mouthing reviews in the dataset.

- Step 2: Pick any random agent X. In this experiment, we pick agent HubSpot Sales. Calculate the total number of reviews for agent HubSpot Sales. Then, the bad mouthing reviews for agent HubSpot Sales will be injected in the following four test cases:
    - Test case 1: 10% bad mouthing reviews in 10 days.
    - Test case 2: 20% bad mouthing reviews in the next 10 days.
    - Test case 3: 50% bad mouthing reviews in the next 10 days.
    - Test case 4: 80% bad mouthing reviews in the next 10 days.
- Step 3: Calculate the reputation of agent HubSpot Sales. The computed value will include bad mouthing values. The reputation for agent HubSpot Sales at the beginning is set to be A_0=0.
- Step 4: Apply our algorithm to identify bad mouthing values.
- Step 5: Calculate the percentage of bad mouthing reviews of agent HubSpot Sales that were filtered out and not taken into account during the reputation calculation due to our algorithm. Compute the following:
    - Percentage of bad mouthing reviews filtered out.
    - Percentage of bad mouthing reviews not filtered out.
    - Compute true positives, false positives, true negatives, and false negatives.
    - Compute Accuracy.
- Step 6: Calculate the reputation of agent HubSpot Sales again. Let this value be A_i. This value is calculated using the formula:

$$A_1 = A_0 + Accuracy_1 * TP_1/NumberOfReviews * 100$$

$$A_2 = A_1 + Accuracy_2 * (TP_2 - TP_1)/NumberOfReviews * 100$$

$$A_3 = A_2 + Accuracy_3 * (TP_3 - TP_2)/NumberOfReviews * 100$$

$$A_4 = A_3 + Accuracy_4 * (TP_4 - TP_3)/NumberOfReviews * 100$$

$$A_i = A_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

where A_1: reputation of agent HubSpot Sales after 10 days,

A_2: reputation of agent HubSpot Sales after 20 days,

A_3: reputation of agent HubSpot Sales after 30 days,

A_4: reputation of agent HubSpot Sales after 40 days.

- Step 7: The admin agent will also calculate the reputation value of agent HubSpot Sales again. Let this value be B_1.

$$B_1 = A_0 + \frac{NumberOfBadMouthingReviews}{NumberOfReviews} * 100$$

- Step 8: Compute the degree of correlation between A_i and B_1. Let this value be CA_i.

$$CA_i = |(A_i - B_1)|, i = \overline{1,4}$$

- Step 9: Apply the BRS algorithm to identify the bad mouthing values.
- Step 10: Calculate the reputation of agent HubSpot Sales again. Let this value be D_i. The value is calculated using the formula:

$$D_i = D_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 11: The admin agent will also calculate the reputation value of agent HubSpot Sales again. Let this value be B_2.

$$B_2 = A_0 + \frac{NumberOfBadMouthingReviews}{NumberOfReviews} * 100$$

- Step 12: Compute the degree of correlation between D_i and B_2. Let this value be CD_i.

$$CD_i = |(D_i - B_1)|, i = \overline{1,4}$$

- Step 13: Apply the TRAVOS algorithm to identify the bad mouthing values.
- Step 14: Calculate the reputation of agent HubSpot Sales again. Let this value be E_i.

$$E_i = E_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 15: The admin agent will also calculate the reputation value of agent HubSpot Sales again. Let this value be B_3.

$$B_3 = A_0 + \frac{NumberOfBadMouthingReviews}{NumberOfReviews} * 100$$

- Step 16: Compute the degree of correlation between A_3 and B_3. Let this value be CE_i.

$$CE_i = |(E_i - B_1)|, i = \overline{1,4}$$

- Step 17: Apply the iCLUB algorithm to identify the bad mouthing values.
- Step 18: Calculate the reputation of agent HubSpot Sales again. Let this value be F_i.

$$F_i = F_{i-1} + Accuracy_i * \frac{TP_i - TP_{i-1}}{NumberOfReviews} * 100$$

- Step 19: The admin agent will also calculate the reputation value of agent HubSpot Sales again. Let this value be B_4.

$$B_4 = A_0 + \frac{NumberOfBadMouthingReviews}{NumberOfReviews} * 100$$

- Step 20: Compute the degree of correlation between A_3 and B_3. Let this value be CF_i.

$$CF_i = |(F_i - B_1)|, i = \overline{1,4}$$

- Step 21: Compare CB_i, CD_i, CE_i and CF_i.

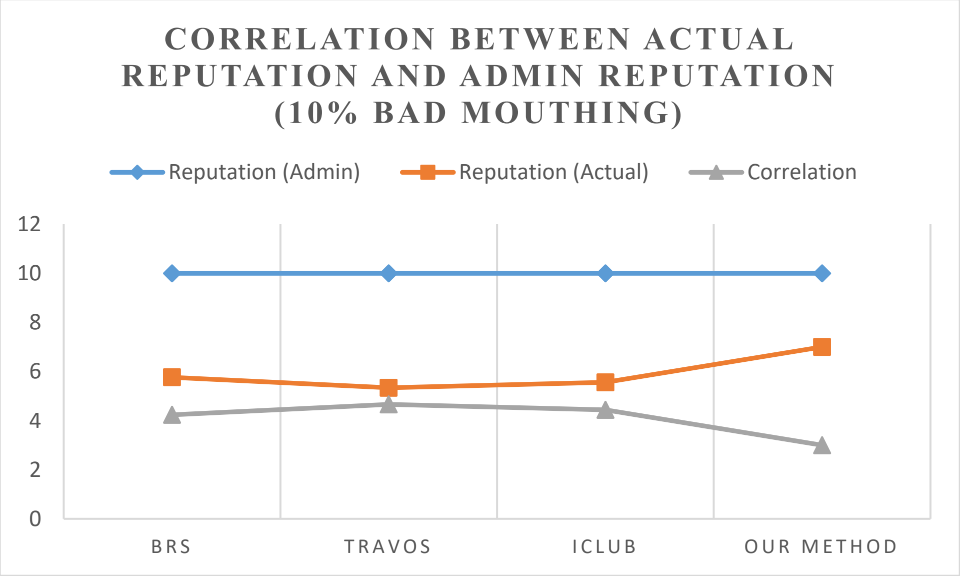


Figure 35: Correlation between actual reputation and admin reputation (10% bad mouthing)

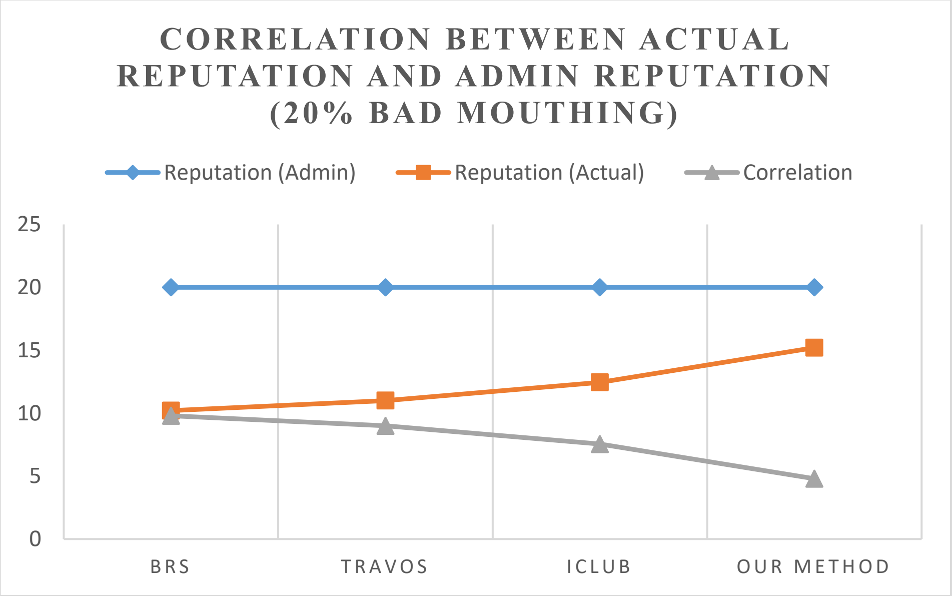


Figure 36: Correlation between actual reputation and admin reputation (20% bad mouthing)

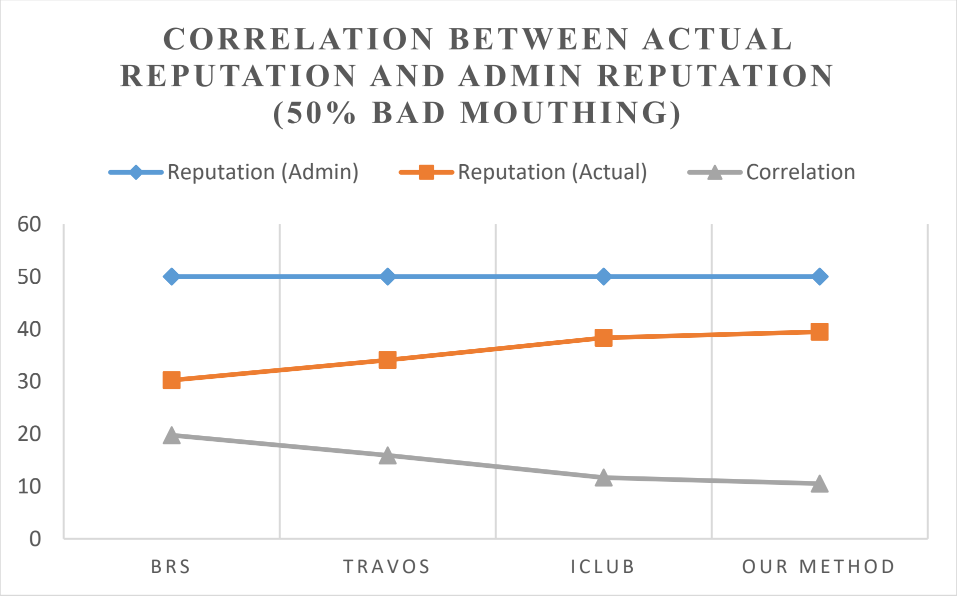


Figure 37: Correlation between actual reputation and admin reputation (50% bad mouthing)

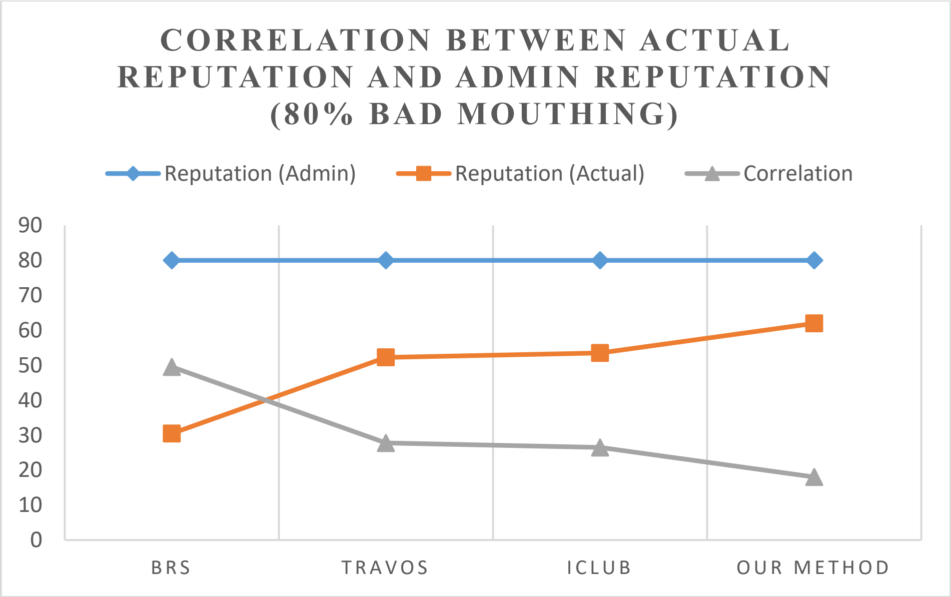


Figure 38: Correlation between actual reputation and admin reputation (80% bad mouthing)

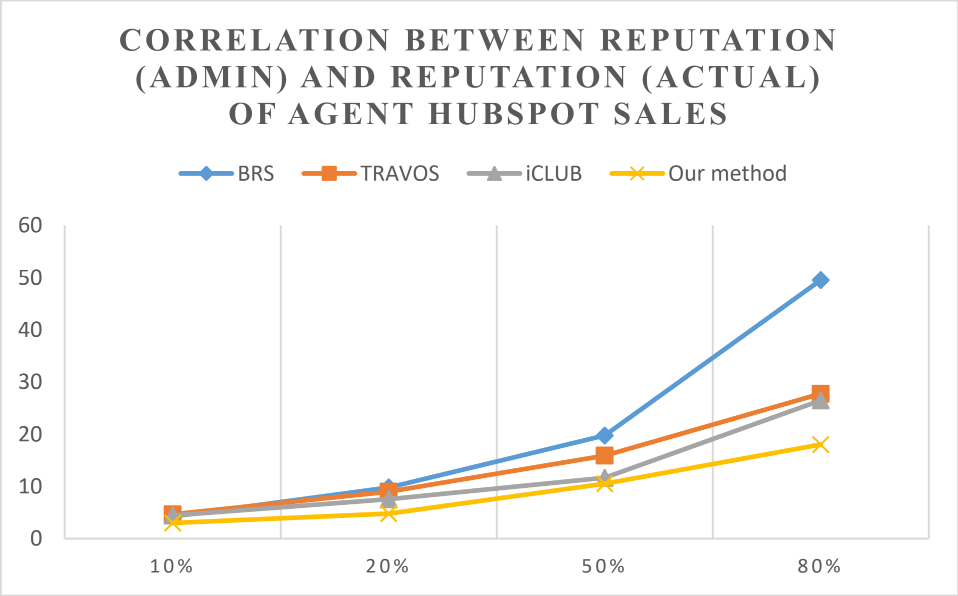


Figure 39: Correlation between actual reputation and admin reputation of agent Box with different percentage of bad mouthing

As shown in Figure 39, there is a slight increase when the percentage of bad mouthing is 10% in the first 10 days, 20% in the next 10 days and 50% in the next 10 days. However, there is a dramatic change when the percentage of bad mouthing is 80% in the last 10 days. This means that BRS is strongly affected by the burstiness capability.

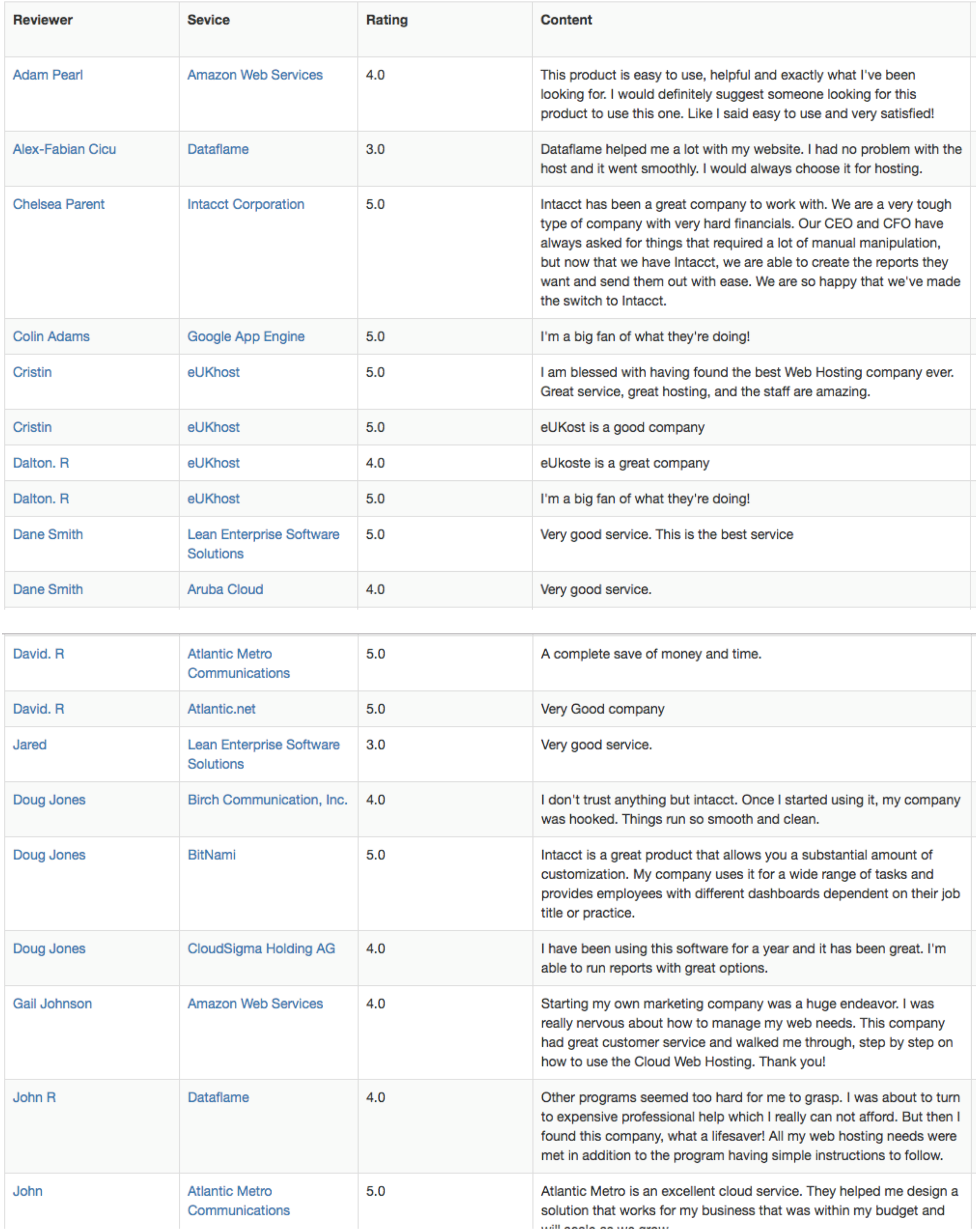
The correlation between the reputation of admin and the actual reputation of our method is always the lowest when the percentage of bad mouthing reviews is 10%, 20%, 50% or even 80%. This means that our method can work best in detecting bad mouthing.

9.3.3. Solution implementation for the algorithm to identify spammer groups

In this section, we perform two experiments in order to compare our two proposed methods. In the first experiment, we prove that the method which uses fuzzy K-means clustering can find more than one group to which a single spam review belongs while the method which uses K-means clustering can find only one group to which a single spam review belongs. In the second experiment, we calculate the number of groups that each method can detect and then we use a paired t-test algorithm to check which method is better.

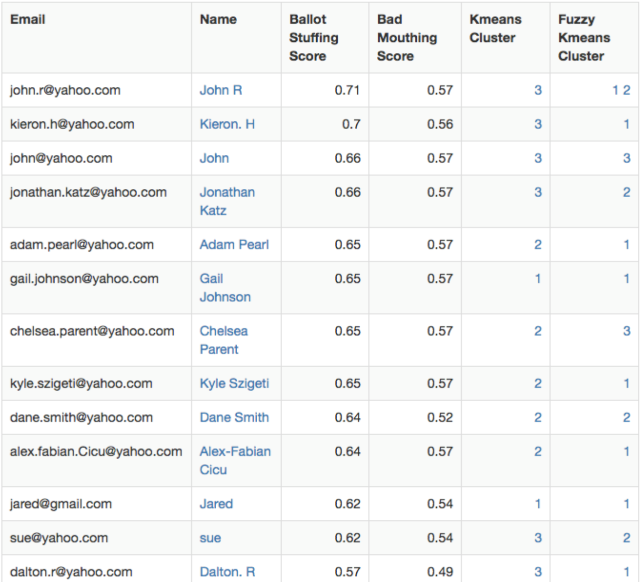
Experiment 1: Check the number of spammer groups that the two proposed methods can detect

The following case study uses a test dataset to evaluate whether our algorithm is able to identify spammer groups. Figure 40 shows a sample from the test set, which includes the reviewer, service, rating and content. We used the network method to calculate a suspicion score for all the reviewers and display them in an ascending order in the third column in Figure 41, hence the first row shows the most suspicious reviewer. To assess these rankings, we ran both the K-means clustering algorithm and the fuzzy K-means clustering algorithm on this dataset. Figure 41 illustrates that the K-means clustering method only finds one cluster for each reviewer, while the fuzzy K-means clustering method found more than one for some reviewers. For example, John R belongs to cluster 3 when using K-means clustering but belongs to both clusters 1 and 2 when using fuzzy K-means clustering. Figure 42 shows the two different clusters to which John R belongs using the fuzzy K-means clustering method.



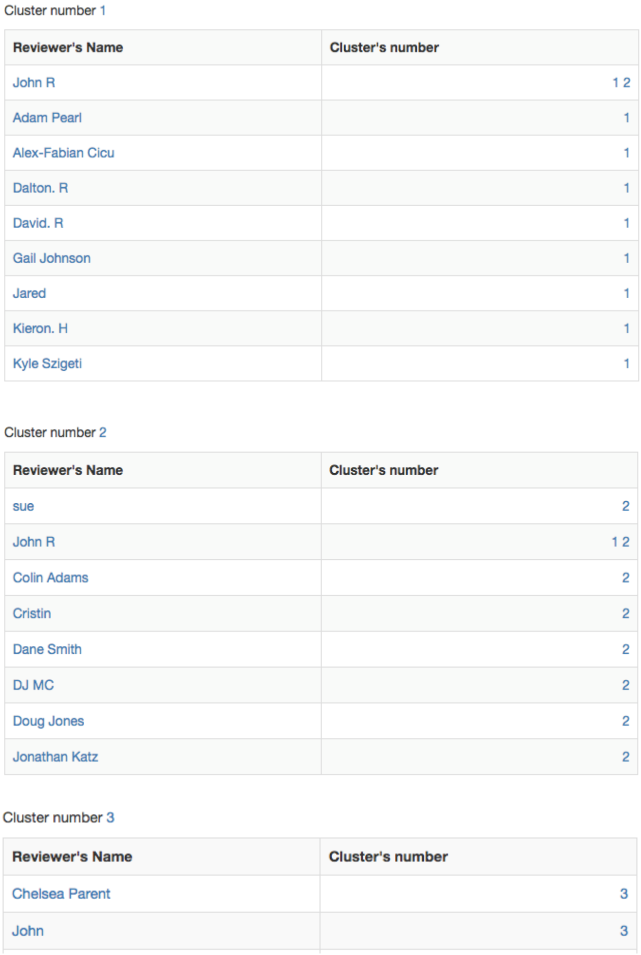
Reviewer	Sevice	Rating	Content
Adam Pearl	Amazon Web Services	4.0	This product is easy to use, helpful and exactly what I've been looking for. I would definitely suggest someone looking for this product to use this one. Like I said easy to use and very satisfied!
Alex-Fabian Cicu	Dataflame	3.0	Dataflame helped me a lot with my website. I had no problem with the host and it went smoothly. I would always choose it for hosting.
Chelsea Parent	Intacct Corporation	5.0	Intacct has been a great company to work with. We are a very tough type of company with very hard financials. Our CEO and CFO have always asked for things that required a lot of manual manipulation, but now that we have Intacct, we are able to create the reports they want and send them out with ease. We are so happy that we've made the switch to Intacct.
Colin Adams	Google App Engine	5.0	I'm a big fan of what they're doing!
Cristin	eUKhost	5.0	I am blessed with having found the best Web Hosting company ever. Great service, great hosting, and the staff are amazing.
Cristin	eUKhost	5.0	eUKost is a good company
Dalton. R	eUKhost	4.0	eUkoste is a great company
Dalton. R	eUKhost	5.0	I'm a big fan of what they're doing!
Dane Smith	Lean Enterprise Software Solutions	5.0	Very good service. This is the best service
Dane Smith	Aruba Cloud	4.0	Very good service.
David. R	Atlantic Metro Communications	5.0	A complete save of money and time.
David. R	Atlantic.net	5.0	Very Good company
Jared	Lean Enterprise Software Solutions	3.0	Very good service.
Doug Jones	Birch Communication, Inc.	4.0	I don't trust anything but intacct. Once I started using it, my company was hooked. Things run so smooth and clean.
Doug Jones	BitNami	5.0	Intacct is a great product that allows you a substantial amount of customization. My company uses it for a wide range of tasks and provides employees with different dashboards dependent on their job title or practice.
Doug Jones	CloudSigma Holding AG	4.0	I have been using this software for a year and it has been great. I'm able to run reports with great options.
Gail Johnson	Amazon Web Services	4.0	Starting my own marketing company was a huge endeavor. I was really nervous about how to manage my web needs. This company had great customer service and walked me through, step by step on how to use the Cloud Web Hosting. Thank you!
John R	Dataflame	4.0	Other programs seemed too hard for me to grasp. I was about to turn to expensive professional help which I really can not afford. But then I found this company, what a lifesaver! All my web hosting needs were met in addition to the program having simple instructions to follow.
John	Atlantic Metro Communications	5.0	Atlantic Metro is an excellent cloud service. They helped me design a solution that works for my business that was within my budget and will scale as we grow.

Figure 40: Test dataset



Email	Name	Ballot Stuffing Score	Bad Mouthing Score	Kmeans Cluster	Fuzzy Kmeans Cluster
john.r@yahoo.com	John R	0.71	0.57	3	1 2
kieron.h@yahoo.com	Kieron. H	0.7	0.56	3	1
john@yahoo.com	John	0.66	0.57	3	3
jonathan.katz@yahoo.com	Jonathan Katz	0.66	0.57	3	2
adam.pearl@yahoo.com	Adam Pearl	0.65	0.57	2	1
gail.johnson@yahoo.com	Gail Johnson	0.65	0.57	1	1
chelsea.parent@yahoo.com	Chelsea Parent	0.65	0.57	2	3
kyle.szigeti@yahoo.com	Kyle Szigeti	0.65	0.57	2	1
dane.smith@yahoo.com	Dane Smith	0.64	0.52	2	2
alex.fabian.Cicu@yahoo.com	Alex-Fabian Cicu	0.64	0.57	2	1
jared@gmail.com	Jared	0.62	0.54	1	1
sue@yahoo.com	sue	0.62	0.54	3	2
dalton.r@yahoo.com	Dalton. R	0.57	0.49	3	1

Figure 41: Clusters of reviewers



Cluster number 1

Reviewer's Name	Cluster's number
John R	1 2
Adam Pearl	1
Alex-Fabian Cicu	1
Dalton. R	1
David. R	1
Gail Johnson	1
Jared	1
Kieron. H	1
Kyle Szigeti	1

Cluster number 2

Reviewer's Name	Cluster's number
sue	2
John R	1 2
Colin Adams	2
Cristin	2
Dane Smith	2
DJ MC	2
Doug Jones	2
Jonathan Katz	2

Cluster number 3

Reviewer's Name	Cluster's number
Chelsea Parent	3
John	3

Figure 42: Clusters using fuzzy K-means clustering

Experiment 2: Comparison of the two proposed methods using a paired t-test

In this validation, we use t-test to compare our two proposed methods

A paired t-test is used to compare two population means when there are two samples in which observations in one sample can be paired with observations in the other sample. Examples of where this might occur are:

- Before-and-after observations on the same subjects (e.g. students' diagnostic test results before and after a particular module or course).
- A comparison of two different methods of measurement or two different treatments where the measurements/treatments are applied to the same subjects (e.g. blood pressure measurements using a stethoscope and a dynamap).

Procedure for carrying out a paired t-test:

Suppose we run the K-means approach and the fuzzy approach to detect spammer groups in 30 days, 45 days and 60 days. Then, we calculate how many groups that these algorithms detect in these different periods of time to determine whether the fuzzy approach leads to improvements in detecting spammer groups. We can use the results from the number of spammer groups detected to draw conclusions about the impact of the fuzzy approach in general.

Let x be the number of spammer groups detected by the K-means approach and y is the number of spammer groups detected by the fuzzy approach. The procedure to test if the fuzzy approach outperforms the K-means approach is as follows:

- Step 1: Calculate the difference ($d_i = y_i - x_i$) between the two observations on each pair, making sure to distinguish between positive and negative differences.
- Step 2: Calculate the mean difference, $\bar{d}$.
- Step 3: Calculate the standard deviation of the differences, s_d, and use this to calculate the standard error of the mean difference, $SE(\bar{d}) = \frac{s_d}{\sqrt{n}}$.
- Calculate the t-statistic, which is given by $T = \frac{\bar{d}}{SE(\bar{d})}$. Under the null hypothesis, this statistic follows a t-distribution with n − 1 degrees of freedom.
- Use tables of the t-distribution to compare the value for T to the t_{n-1} distribution. This will give the p-value for the paired t-test.

We run our two algorithm to identify spammer groups in 30 days, 45 days and 60 days respectively and the results are shown in Table 4 and Table 5.

	30 days	45 days	60 days
Number of groups	1005	1154	1208
Number of reviews	3164	3446	3603

Table 4: Number of groups and number of reviewers detected using the K-means approach

Fuzzy K-means clustering

	30 days	45 days	60 days
Number of groups	1147	1257	1296
Number of reviewers	3367	3685	3798

Table 5: Number of groups and number of reviewers detected using the fuzzy approach

The number of spammer groups detected using the K-means approach and the fuzzy approach is shown in Table 6.

	K-means	Fuzzy K-means	Difference
30 days	1005	1147	142
45 days	1154	1257	103
60 days	1208	1296	88

Table 6: Number of spammer groups detected using the K-means approach and the fuzzy approach

Calculating the mean and standard deviation of the differences gives: $\bar{d} = 111$ and $s_d = 27.87472$. Therefore, $SE(\bar{d}) = \frac{s_d}{\sqrt{n}} = \frac{27.87472}{\sqrt{3}} = 16.09348$

So, we have:

$T = \frac{\bar{d}}{SE(d)} = 6.897204$ on 2df

This gives p = 0.01. Therefore, there is strong evidence that, on average, the fuzzy approach leads to improvements compared to the K-means approach.

9.4. Conclusion

In this chapter, we introduce the solution implementation for three of the four methods to identify ballot stuffing, bad mouthing and spammer groups.

In the implementation of the methods to identify ballot stuffing, we compare our method with three existing methods, BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. In the experiment to compare the overall performance, we conducted an experiment on a dataset which includes 15,000 reviews, in which the percentage of ballot stuffing reviews is 10%, 20%, 50% and 80%, respectively. The results show that when the percentage of ballot stuffing is 10%, 20% and 80%, the accuracy of our method is the highest while the accuracy of BRS is the lowest. If the percentage of ballot stuffing reviews is 50%, the accuracy of BRS and iCLUB is slightly higher than TRAVOS. In this case, the accuracy of our method is still the highest compared to the other three methods. When there are 12000 ballot stuffing reviews, the accuracy of our method is significantly higher than BRS.

We also measure the precision and recall of BRS, TRAVOS, iCLUB and our method when the percentage of ballot stuffing is 10%, 20%, 50% and 80% of the total number of reviews. Our method has the highest precision and highest recall compared to the others when the percentage of ballot stuffing is 10%, 20% or 50% which means that our method is very good in most cases, however it is very picky because it does not think many reviews are ballot stuffing, but despite this, it still identifies most of the ballot stuffing reviews which are actually ballot stuffing. When the percentage of ballot stuffing reviews is 80%, the precision of BRS and iCLUB is higher but the recall of BRS and iCLUB is lower than our method. This means that BRS and iCLUB can detect more ballot stuffing that is actually ballot stuffing and more "not ballot stuffing" that is actually not ballot stuffing. In all cases, BRS has the lowest precision and our method has the highest precision. When 20% and 50% of the reviews are ballot stuffing, the precision of TRAVOS and iCLUB is only slightly different. When 80% of the reviews are ballot stuffing, our method has the highest precision, TRAVOS has the second, iCLUB's precision is third and BRS's precision is the lowest. The recall of our method is the highest in most cases, except when 80% of the reviews are ballot stuffing. When 10% or 80% of the reviews are ballot stuffing, the recall of BRS is the lowest while the recall of TRAVOS is the lowest when 20% or 50% of the reviews are ballot stuffing. However, our method is still the most accurate compared to BRS, TRAVOS and iCLUB in all situations.

Moreover, we also use two metrics, FPR and FNR, to check the majority capability of our method and the three existing methods. The results show that when the majority of the reviews are incorrect, the FPR of BRS and iCLUB increase and approach 1 while the FNR of BRS and iCLUB decrease and approach 0. This means if the majority of the reviews are ballot stuffing, BRS and iCLUB tend to identify every review as ballot stuffing. As the FPR of BRS is closer to 1 and the FNR of BRS is closer to 0 than the three other methods, we can conclude that BRS is most affected when the majority of the reviews are ballot stuffing. TRAVOS and our method is not affected much when more than half of the reviews are ballot stuffing. TRAVOS is a slightly less affected by the majority capability than our method. However, our method is still the best in all cases when considering accuracy.

Furthermore, the burstiness capability of our method and the three previous methods are also reviewed in this validation. Based on the results of our experiment, there is a slight increase when the percentage of ballot stuffing is 10% in the first 10 days, 20% in the next 10 days and 50% in the next 10 days. However, there is a dramatic change when the percentage of ballot stuffing is 80% in the last 10 days. This means that BRS is strongly affected by the burstiness capability. The correlation between the reputation of admin and the actual reputation of our method is always the lowest when the percentage of ballot stuffing reviews is 10%, 20%, 50% or even 80%. This means that our method is the best at detecting ballot stuffing.

In the implementation of the methods to identify bad mouthing, we compare our method with three existing methods, BRS, TRAVOS and iCLUB using three metrics, accuracy, precision and recall. In the experiment to compare the overall performance, we conducted an experiment on a dataset which includes 15,000 reviews, in which the percentage of bad mouthing reviews is 10%, 20%, 50% and 80%, respectively. The results show that the accuracy of our method is better than BRS, TRAVOS and iCLUB when being tested with different percentages of incorrect reviews. We also measured the precision and recall of BRS, TRAVOS, iCLUB and our method when the percentage of bad mouthing is 10%, 20%, 50% and 80% of the total number of reviews. The results show that our method has the highest precision and highest recall compared to the others in two cases when the percentage of bad mouthing reviews is 10% and 20% which means that our method is very good, however it is very picky because it does not think many reviews are bad mouthing, but despite this, it still identifies most of the bad mouthing reviews which are actually bad mouthing. When the percentage of bad mouthing reviews is 50%, the precision and recall of our method is higher than BRS and iCLUB. However, when compared to TRAVOS, the precision of our method is higher than TRAVOS but the recall of our method is lower than TRAVOS. This means that when comparing between our method and TRAVOS, our method does not identify as many bad mouthing as TRAVOS does. All the images it identifies as bad mouthing are actually bad mouthing. However, it also misses a lot of actual bad mouthing reviews. When the percentage of bad mouthing reviews is 80%, the precision of BRS and iCLUB is lower but the recall is higher than our method. This means that our method identifies many reviews as bad mouthing. However, it also identifies that a lot of bad mouthing reviews are actually bad mouthing. So, from our set of images we have many images classified as bad mouthing, many of these actually being bad mouthing but many of them were not bad mouthing. However, our method is still the most accurate compared to BRS, TRAVOS and iCLUB in all situations.

Moreover, we also used two metrics, FPR and FNR, to check the majority capability of our method and the three existing methods. The results show that FPR of BRS and iCLUB increase and approach 1 while the FNR of BRS and iCLUB decrease and approach 0, especially when the majority of the reviews are incorrect. This means if the majority of the reviews are bad mouthing, BRS and iCLUB tend to identify every review as bad mouthing. As the FPR of BRS is closer to 1 and the FNR of BRS is closer to 0 than the three other methods, we can conclude that BRS is most affected when the majority of the reviews are bad mouthing. TRAVOS is slightly less affected by the majority capability than our method. However, our method is still the best in all cases when considering accuracy.

Furthermore, the burstiness capability of our method and the three other methods is also reviewed in this validation. In this experiment, there are 2,000 reviews, 5,000 reviews, 10,000 reviews and 15,000 reviews with the percentage of bad mouthing reviews are 10%, 20%, 50% and 80%, respectively, bursting in the period of 60 days (for example, in the first period of 10 days, there are 2000 reviews with 10% bad mouthing reviews, in the next period of 20 days, there are 3,000 more reviews with 20% bad mouthing reviews, in the next period of 20 days, there are 5,000 more reviews with 50% bad mouthing reviews, in the next period of 10 days, there are 5,000 more reviews with 80% bad mouthing reviews). As shown in the results, there is a slight increase when the percentage of bad mouthing is 10% in the first 10 days, 20% in the next 10 days and 50% in the next 10 days. However, there is a dramatic change when the percentage of bad mouthing is 80% in the last 10 days. This means that BRS is strongly affected by the burstiness capability. The correlation between the reputation of admin and the actual reputation of our method is always the lowest when the percentage of bad mouthing reviews is 10%, 20%, 50% or even 80%. This means that our method is the best at detecting bad mouthing.

In the implementation of the methods to identify spammer groups, we compare our two proposed methods with each other by conducting two experiments. The first experiment proves that the K-means approach is able to find one and only one group to which a single spam review belongs while the fuzzy approach is able to find more than one group to which a single spam review belongs. The second experiment uses a paired t-test algorithm to compare our two proposed methods to determine which one is better. Based on the result, the fuzzy approach leads to an improvement compared to the K-means approach in the identification of spammer groups.

Due to the time restriction of this thesis, the implementation of the method to identify cliques has not been proposed. This will be carried out in future research.

9.5. References

[1] Whitby, A., Jøsang, A. & Indulska, J., 2004, 'Filtering out unfair ratings in Bayesian reputation systems', In *Proc. 7th Int. Workshop on Trust in Agent Societies*, vol. 6, pp. 106-17.

[2] David, J. & Goadrich, M. 2006, The relationship between precision-recall and ROC curves, Technical report, viewed 20 October 2018, http://research.cs.wisc.edu/techreports/2006/TR1551.pdf.

CHAPTER 10:
RECAPITULATION AND FUTURE WORK

10.1. Introduction

An increasing number of companies are looking for ways to reduce their information technology development costs. In order to select a cloud provider, companies will check the cloud reviews of previous customers. Therefore, the cloud reviews that are published on the providers' websites and third-party websites will affect the decision of future customers. For these reasons, it is critical that incorrect reviews on cloud reputation systems are identified. As is evident from the state-of-the-art described in Chapter 2, researchers have proposed several methods to identify incorrect reviews. However, there are several shortcomings to these methods to identify the four types of incorrect reviews which are discussed in this thesis.

To propose a solution to identify the four types of incorrect reviews on cloud reputation systems, this thesis identified five research directions and addressed them.

In the next section, we discuss the problems related to identifying the incorrect reviews on cloud reputation systems that were addressed in this thesis. In Section 10.3, we detail the five contributions of this thesis to the existing literature. Section 10.4 concludes the thesis and sets the stage for future work.

10.2. Problems addressed in this thesis

This thesis focuses on methodologies to identify four types of incorrect reviews, ballot stuffing, bad mouthing, spammer groups and cliques. Of the four types, ballot stuffing and bad mouthing are the two most commonly used types of incorrect reviews which has attracted a large amount of research focus. Although there are many studies on detecting ballot stuffing and bad mouthing, there are still several drawbacks in the methodologies to identify ballot stuffing and bad mouthing, these being: (1) there is no holistic research that identifies all four types of incorrect reviews in cloud reputation systems, (2) most of the existing research to identify ballot stuffing or bad mouthing is not capable of solving the problem of advisors providing thousands of ratings within a short period of time, (3) most of the existing research to identify ballot stuffing or bad mouthing is not effective when more than half of the reviews are ballot stuffing or bad mouthing, (4) most of the existing research to identify ballot stuffing or bad mouthing does not consider the relationship between review, reviewer and product, (5) most of the methods to identify spammer groups find all the groups first and then check which groups are spammer groups later which can be more time consuming to find all the groups for a large number of reviews, (6) none of the existing studies examine cloud reputation systems.

10.3. Contribution of this thesis to the existing literature

10.3.1. Contribution 1 – Holistic research that identify all four types of incorrect reviews

Even though there is a large volume of research on developing intelligent methods to identify spam in reviews, there is no holistic research that identifies all four types of incorrect reviews in cloud reputation systems. This thesis is the first to identify all four types of incorrect reviews in cloud reputation systems, these being ballot stuffing, bad mouthing, spammer groups and cliques.

In order to identify ballot stuffing, the suspicion scores of three types of nodes in a review graph, reviewers, reviews and products are calculated and then naïve Bayes is used to conclude if a review is ballot stuffing or not.

In order to identify bad mouthing, the same method to that utilised to identify ballot stuffing is used, however, instead of calculating suspicion scores based on high-rating reviews, we use low-rating reviews.

After identifying all the ballot stuffing and bad mouthing reviews, all the groups to which these reviews belong are detected using *K-means* or *fuzzy K-means clustering*. By using this approach, we identify the spammer groups.

In order to identify cliques, we use a graph-based method. This method also makes use of the ballot stuffing detected previously.

10.3.2. Contribution 2 – Intelligent method to identify ballot stuffing

This thesis proposes an intelligent method to identify ballot stuffing in cloud reputation systems. In order to identify ballot stuffing, a networked-based method is used. This method makes use of a review graph that comprises three types of nodes, reviewers, reviews, and products. There are four steps in the process to identify ballot stuffing reviews. In the first three steps, the suspicion scores of reviewers, reviews and products are calculated, respectively. In the last step, naïve Bayes is used to find all the ballot stuffing reviews.

10.3.3. Contribution 3 – Intelligent method to identify bad mouthing

This thesis proposes an intelligent method to identify bad mouthing in cloud reputation systems. In order to identify bad mouthing, a networked-based method is used. This method makes use of a review graph that comprises three types of nodes, reviewers, reviews, and products. There are four steps in the process to identify bad mouthing reviews. In the first three steps, the suspicion scores of reviewers, reviews and products are calculated, respectively. In the last step, naïve Bayes is employed to find all the bad mouthing reviews. This method is similar to the method to identify ballot stuffing discussed in section 10.3.2, however, instead of using high-rating reviews to calculate the suspicion scores, low-rating reviews are used.

10.3.4. Contribution 4 – Intelligent methods to identify spammer groups

This thesis proposes two intelligent methods to identify spammer groups in cloud reputation systems. There are two stages in the process of identifying spammer groups. In the first stage, ballot stuffing and bad mouthing reviews are identified using a network-based method. This stage comprises four steps that are discussed in section 10.3.2 and section 10.3.3. In the second stage, the groups of ballot stuffing/bad mouthing reviews identified in the first stage are identified using two different types of approaches. The first approach to find all the groups to which the identified ballot stuffing/bad mouthing reviews belong is to use K-means clustering. The second approach to find all the groups to which the identified ballot stuffing/bad mouthing belong is to use fuzzy K-means clustering.

10.3.5. Contribution 5 – Intelligent method to identify cliques

To the best of the researcher's knowledge, this research is the first to define cliques in the area of cloud reviews and is the first to propose an intelligent method to identify cliques in cloud-based reputation systems. In order to identify cliques, we use a graph-based method which comprises nine steps. In the first step, we find all the ballot stuffing reviews using the method discussed in section 10.3.2, then we calculate the total number of stars of all the reviews. In the second step, the shortest path distance between two nodes and the influence of each node are calculated using the Dijkstra algorithm and

Epanechnikov kernel, respectively. In the third step, the influence region and direct region for each node are defined. In the fourth step, the density of each node is calculated which demonstrates the total number of reviews that providers received from others and how highly that provider is rated. In the fifth step, all the core nodes from the node list are identified and selected. In the sixth step, the list of core nodes is clustered into separate subsets based on the direct region of each node. In the next step, all cliques which are strong core clusters are identified. We also identify all the semi-strong core clusters which might easily become cliques in step 8. In the last step, we update the providers' reputation based on all the strong core clusters (cliques) and semi-strong core clusters identified previously. Reputation is one of the most important criteria in searching service providers, which means that the providers with a higher reputation have a higher priority in the search result list.

10.4. Conclusion and future work

In conclusion, this thesis defines four types of incorrect reviews, ballot stuffing, bad mouthing, spammer groups and cliques and proposes intelligent methods to detect these four types of incorrect reviews. In the validation part for the intelligent methods to identify ballot stuffing and bad mouthing, we compare our method with other proposed methods in identifying ballot stuffing and bad mouthing using metrics such as accuracy, precision and recall. In the validation part for intelligent methods to identify spammer groups, we compare our two proposed methods to identify spammer groups using the paired t-test. We also prove that our second method using fuzzy K-means clustering is able to find more groups for each one single incorrect review than the first method which uses K-means clustering. In all the validation sections, we use the dataset which was crawled from the website www.getapp.com.

Although we have undertaken a lot of research on the topic of this study, there is still plenty of scope for future work. Therefore, we intend to continue working on this topic, primarily along, but not limited to, the following lines:

(1) Validation of our methods to identify spammer groups compared with other methods: Due to the restriction of time to undertake this thesis, we did not validate our methods to identify spammer groups by comparing them with other existing methods, nor did we check the time to run each method. Therefore, in future work,

we will conduct experiments to compare our methods to identify spammer groups with other existing algorithms. The time to run each method to identify spammer groups will also be measured in future work. Moreover, as discussed in Section 4.5, there are many other existing methods to find all the groups to which a ballot stuffing or a bad mouthing review belongs, such as Pearson correlation coefficient and Spearman correlation coefficient. We can apply these methods in order to identify spammer groups. Then, we can compare these methods with the methods proposed in this thesis to determine which is the best. We can also compare the running time between all the methods.

(2) Validation of the method to identify cliques in cloud reputation systems: Due to the restriction of time to undertake this thesis, we did not validate the intelligent method to identify cliques. Therefore, in future work, we will present a way to validate the intelligent method to detect cliques in cloud reputation systems. In theory, although our algorithm seems to be an appropriate solution, we are still not sure if it will work properly in a real dataset. Therefore, in future research, an application will be developed to apply this algorithm to a real dataset to determine if it is an efficient solution in this case.

(3) As we have only validated our method using a dataset collected from one website only, there is a need to apply our algorithms to different datasets on different types of reviews such as travel reviews, restaurant reviews, cloud reviews and compare our algorithms with other existing algorithms in different datasets. A stress test is also needed to check the performance of the algorithms.

(4) To find a cloud provider that best meets their requirements, a customer has to check many different websites. This can be very time consuming for customers. Therefore, our aim in the future is to find more types of incorrect reviews and find the best methods to identify them so that we can create a one-stop shop for cloud providers using the best methods available. We might also personalize the needs of each customer and help them choose the most appropriate cloud provider based on their needs.

BIBLIOGRAPHY

[1] Alkalbani, A.M., Ghamry, A.M., Hussain, F.K. and Hussain, O.K., 2016, ‘Harvesting Multiple Resources for Software as a Service Offers: A Big Data Study’, *In International Conference on Neural Information Processing*, pp. 61-71. Springer, Cham.

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I.O.N. & Zaharia, M. 2010, 'A View of Cloud Computing', *Communications of the ACM*, vol. 53, no. 4, pp. 50-8.

[3] Buchegger, S. & Le Boudec, J.Y., 2003, *A robust reputation system for mobile ad-hoc networks*, No. LCA-REPORT-2003-006.

[4] Chang, C. T., Lai, J. Z. & Jeng, M. D. 2011, ‘A fuzzy K-means clustering algorithm using cluster center displacement’, *Journal of Information Science and Engineering*, vol. 27, no. 3, pp. 995-1009.

[5] Chen, M. & Singh, J.P., 2001, ‘Computing and using reputations for internet ratings’, *In Proceedings of the 3rd ACM Conference on Electronic Commerce, ACM*, pp. 154-62.

[6] David, J. & Goadrich, M. 2006, The relationship between precision-recall and ROC curves, Technical report, viewed 20 October 2018, http://research.cs.wisc.edu/techreports/2006/TR1551.pdf.

[7] Dellarocas, C., 2000, ‘Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior’, *In Proceedings of the 2nd ACM Conference on Electronic Commerce, ACM*, pp. 150-57.

[8] Dellarocas, C., 2000, ‘Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems’, *In Proceedings of the Twenty first International Conference on Information Systems, Association for Information Systems*, pp. 520-5.

[9] Duh, A., Štiglic, G. & Korošak, D., 2013, ‘Enhancing identification of opinion spammer groups’, *In Proceedings of International Conference on Making Sense of Converging Media, ACM*, p. 326.

[10] Farooq, S. & Khanday, H.A., 2016, ‘Opinion Spam Detection: A Review’, *International Journal of Engineering Research and Development*, vol. 12, no. 4, pp. 1-8.

[11] Fayazbakhsh, S.K. & Sinha, J., 2012, Review spam detection: a network-based approach. Final Project Report: CSE, 590.

[12] G¨unnemann, S. & Seidl, T. 2010, ‘Subgraph Mining on Directed and Weighted Graphs’, *the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010)*, Springer - Heidelberg, Germany, pp. 133-46.

[13] Gill, R. 2011, 'Why Cloud Computing Matters to Finance', *Strategic Finance*, vol. 92, no. 7, pp. 43-7.

[14] Gupta, A. 2010, 'Cloud computing growing interest and related concerns', *2010 2nd International Conference on Computer Technology and Development (ICCTD),* pp. 462-5.

[15] Hofmann, P. & Woods, D. 2010, 'Cloud Computing: The Limits of Public Clouds for Business Applications', Internet Computing, *IEEE*, vol. 14, no. 6, pp. 90-3.

[16] Ilakiya, K.S. & Felciah, M.M.L.P., 2015, ‘Challenges and techniques for Sentiment Analysis: a survey’, *IJCSMC*.

[17] Jain, A.K., Dubes, R.C. 1988, Algorithms for Clustering Data, Prentice Hall.

[18] Jamali, M. & Abolhassani, H. 2006, ‘Different aspects of social network analysis’, *IEEE/WIC/ACM International Conference on Web Intelligence,* pp. 66-72.

[19] Jiang, S., Zhang, J. & Ong, Y.S., 2013, ‘An evolutionary model for constructing robust trust networks’, *In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, International Foundation for Autonomous Agents and Multiagent Systems,* pp. 813-20.

[20] Jindal, N. & Liu, B., 2008, ‘Opinion spam and analysis’, *In Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM*, pp. 219-30.

[21] Jindal, N., Liu, B. & Lim, E.P., 2010, ‘Finding unusual review patterns using unexpected rules’, *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM*, pp. 1549-52.

[22] Josang, A. & Ismail, R., 2002, ‘The beta reputation system’, *In Proceedings of the 15th Bled Electronic Commerce Conference*, vol. 5, pp. 2502-11.

[23] Jøsang, A., Gray, E. & Kinateder, M. 2003, ‘Analysing Topologies of Transitive Trust’, *Proceedings of the Workshop of Formal Aspects of Security and Trust (FAST 2003)*, Pisa, Italy, pp. 9-22.

[24] Kolhe, N.M., Joshi, M.M., Jadhav, A.B. & Abhang, P.D., 2014, ‘Fake reviewer groups' detection system’, *Journal of Computer Engineering (IOSR-JCE)*, vol. 16, no. 1, pp. 6-9.

[25] Li, F., Huang, M., Yang, Y. & Zhu, X., 2011, ‘Learning to identify review spam’, *In IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, p. 2488.

[26] Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B. & Lauw, H.W., 2010, ‘Detecting product review spammers using rating behaviors’, *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM*, pp. 939-48.

[27] Liu, G. & Wong, L. 2008, ‘Effective pruning techniques for mining quasi-cliques’, *In:ECML/PKDD*, vol. 2, pp. 33–49.

[28] Liu, S., Zhang, J., Miao, C., Theng, Y.L. & Kot, A.C., 2014, ‘An integrated clustering-based approach to filtering unfair multi-nominal testimonies’, *Computational Intelligence*, vol. 30, no. 2, pp. 316-41.

[29] Mukherjee, A., Liu, B. & Glance, N., 2012, ‘Spotting fake reviewer groups in consumer reviews’, *In Proceedings of the 21st International Conference on World Wide Web, ACM,* pp. 191-200.

[30] Mukherjee, A., Liu, B., Wang, J., Glance, N. & Jindal, N., 2011, ‘Detecting group review spam’, *In Proceedings of the 20th International Conference Companion on World Wide Web, ACM,* pp. 93-4.

[31] Ott, M., Choi, Y., Cardie, C. & Hancock, J.T., 2011, ‘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, Association for Computational Linguistics*, vol. 1, pp. 309-19.

[32] Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. 2007, ‘A design science research methodology for information systems research’, *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77.

[33] Qi, L., Ni, J., Yan, C., Xia, X. & Ma, C., 2014, ‘Why are Reputation Systems Absent from Cloud Services: Reason and Solution’, *The Sixth International Conferences on Advanced Service Computing*, pp. 9-14.

[34] Regineri, M. 2007, ‘Finding All Cliques of an Undirected Graph’, “Current Trends in IE" WS.

[35] Teacy, W.L., Patel, J., Jennings, N.R. & Luck, M., 2006, ‘Travos: Trust and reputation in the context of inaccurate information sources’, *Autonomous Agents and Multi-Agent Systems*, vol. 12, no. 2, pp.183-98.

[36] Vassilevska, V. 2009, ‘Efficient algorithms for clique problems’, *Information Processing Letters*, vol. 109, no. 4, pp. 254-57.

[37] Wang, G., Xie, S., Liu, B. & Yu, P.S., 2012, ‘Identify online store review spammers via social review graph’, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 61.

[38] Whitby, A., Jøsang, A. & Indulska, J., 2004, ‘Filtering out unfair ratings in Bayesian reputation systems’, *In Proc. 7th Int. Workshop on Trust in Agent Societies*, vol. 6, pp. 106-17.

[39] Ye, J. & Akoglu, L., 2015, ‘Discovering opinion spammer groups by network footprints’, *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Cham, pp. 267-82.

[40] Yolum, P. & Singh, M. 2003, ‘Dynamic Communities in Referral Networks’, *Web Intelligence and Agent Systems (WIAS)*, vol. 1. no. 2, pp. 105-16.

[41] Zhang, J., Sensoy, M. & Cohen, R., 2008, ‘A detailed comparison of probabilistic approaches for coping with unfair ratings in trust and reputation systems’, *In Privacy, Security and Trust, 2008. PST'08. Sixth Annual Conference on, IEEE*, pp. 189-200.