



**DETECTION OF SHILLING ATTACK ON
RECOMMENDER SYSTEMS AND FAKE TEXT REVIEWS:
A STUDY TO EVALUATE THE CORRELATION OF THE TWO
APPROACHES**

Submitted by

Nguyen The Tien Dat

20634994

Thesis submitted for

CSE5TSB - Thesis B

Supervisor Name

Dr. Eric Pardede

Co-Supervisor Name

Mr. Syed Mahbub

Department of Computer Science and Information Technology

LATROBE UNIVERSITY

June 2022

Abstract

The growth of the internet and e-commerce has brought many opportunities for both businesses and consumers. Due to the inability to see and experience products beforehand, current online shoppers rely heavily on reviews and ratings of past customers to make decisions. This can be very risky, as this information might not be genuine. There have been many studies on this phenomenon, as well as how to detect them. However, these studies are often in one of two distinct branches. The first branch is Shilling attacks, which focuses on numerical ratings and assumes a number of fake profiles are inserted into the system. The second one is fake textual reviews, focusing on review content and information extracted from users and products. Although e-commerce websites now allow users to post both numeric and text reviews, the detection models for the two types of attacks have been developed separately. This raises the question of the relevance of the two fields of study, as well as the possibility of combining them. In this thesis, the way each type of detection model works will be explored. A simple fake review detection model is built with good detection results. Then an experiment applying a fake text reviews dataset to the shilling attack detection models is conducted, which gives relatively good results. This confirms the relationship between the two types of attacks, opening the prospect of combining the two research directions in the future.

Table of Contents

1. Introduction.....	1
1.1. Background/motivation.....	1
1.2. Research problem.....	2
1.3. Research methodology.....	2
2. Literature reviews.....	4
2.1. Fake reviews detection.....	4
2.1.1. Fake reviews definition.....	4
2.1.2. Detection models.....	4
2.2. Shilling attacks.....	7
2.2.1. Shilling attacks definition.....	7
2.2.2. Shilling attack detection models.....	10
2.3. Datasets.....	12
3. Fake review Detection Models.....	15
3.1. Dataset Exploration.....	15
3.2. Features Extraction.....	17
3.3. Preprocessing.....	20
3.4. Prediction Models.....	21
3.5. Result and Evaluation.....	22
4. Shilling attack detection models.....	24
4.1. Testing SDLib library with the dataset provided in the library.....	24
4.1.1 Data exploration.....	24
4.1.2 Testing shilling attack detection models in SDLib with 2 datasets.....	27
4.2 Testing SDLib library with YelpZip dataset.....	29
5. Analysis and comparison.....	33
6. Conclusion.....	35
References.....	36

List of Tables

Table 2.1 Fake review detection methods.....	6
Table 2.2 Typical Profiles of a Normal User (a) and a Fake User (b)	9
Table 2.3 Six Popular Shilling Attack Strategies.....	10
Table 2.4 The Datasets used in RS Attack Detection papers.....	13
Table 2.5 The Datasets used in Fake Reviews Detection papers.....	13
Table 3.1 Information of 3 files in YelpZip dataset.....	15
Table 3.2 Features used in this thesis and in four reference fake review detection models	19
Table 3.3 Confusion matrix (a) and the related evaluation metrics (b)	22
Table 3.4 Results of review detection models using 5 different classifiers.....	23
Table 3.5 Comparison between the fake review detection built in the thesis and 4 reference models	24
Table 4.1 Statistics of Amazon dataset and AverageAttack dataset	27
Table 4.2 Results of SDLib algorithms using Amazon dataset	28
Table 4.3 Results of SDLib algorithms using AverageAttack dataset.....	28
Table 4.4 Results of SDLib algorithms using YelpZip dataset (Scenario 1)	29
Table 4.5 Results of SDLib algorithms using YelpZip dataset (Scenario 2).....	30
Table 4.6 Results of SDLib algorithms using YelpZip dataset (Scenario 3).....	31
Table 4.7 Results of SDLib algorithms using YelpZip dataset (Scenario 4).....	32
Table 4.8 Dataset statistics and prediction results in 4 testing scenarios using YelpZip.....	32

List of Figures

Figure 1.1 Research structure	3
Figure 2.1 Categorized Algorithms used in Shilling Attacks Detection Models.....	11
Figure 3.1 Screenshot of the merged YelpZip dataset.....	16
Figure 3.2 Distribution of ratings group by user types in YelpZip.....	16
Figure 3.3 Wordclouds for fake reviews.....	17
Figure 3.4 Screenshot of dataset with all the features extracted in Python notebook	20
Figure 4.1 Distribution of ratings for fake profiles (1) and genuine profiles (0) in Amazon dataset	25
Figure 4.2 Distribution of ratings for fake profiles (1) and genuine profiles (0) in AverageAttack dataset	26

1. Introduction

1.1. Background/motivation

Crowd-based decision-making has a long history, going hand-in-hand with the evolution of how past customer reviews are collected and published. Before the internet appeared nearly a century ago, customers were able to choose restaurants and hotels for themselves based on the number of stars rated by reputable organizations [1] [2]. Today, with the development of the internet and e-commerce platforms like Amazon and Ebay, reading ratings and reviews from other users has become an integral part of the buying process. The importance of online reviews is so significant that it has become central to the business models of companies like Yelp and TripAdvisor. Some statistics show that 93% shopper read reviews before buying products [3], and review interaction has increased by 50% since pre-pandemic [4]. While the usefulness of this information to consumers is undisputed, its authenticity remains questionable.

The seriousness of the fraudulent-reviews problem is partly reflected by the media and search engines. Google returns over a billion results for keywords like "fake reviews" and "fake ratings". Services and Facebook groups that provide fake reviews to businesses have been discovered and reported [5]. Online commerce platforms have taken various measures to show that they are dealing with the problem effectively. Amazon announced on Twitter [6] that it had detected over 200 million suspicious reviews before they were shown to end users in 2020. They also banned 600 Chinese brands due to review abuse issue in 2021 [7] and have taken legal actions against some fake review brokers in 2022 [8]. Yelp has published the results of a survey on the trustworthiness of different types of reviews to users [9]. Meanwhile, consumers can help themselves somewhat overcome this difficulty, by looking for tips to avoid suspicious reviews or using online fake review detection tools [10].

While the phrase "fake reviews" is used in the media today to refer to both text and numeric reviews, from an academic perspective, these are two relatively different matters. Studies of fake numerical ratings are often associated with recommendation systems, along with so-called shilling attacks. Recommender systems are systems that predict products or content that users may be interested in. Shilling attacks are understood as when fake user profiles are created to insert bogus numerical ratings into the system, with the goal of influencing recommendation results in the favor of the attackers. Meanwhile, fake reviews are understood as text reviews for with a purpose other than sharing the user's own personal experience.

There is a large amount of research on both these types of attacks. Over the years, many shilling attack detection models and fake reviews models have also been introduced and improved. Despite the fact that numeric ratings and text reviews exist together in modern online commerce websites, the detection models for these two types of attacks are researched and developed relatively independently of each other. The most advanced fake review detection models use numerical rating information, but theories of shilling attack are not included. This offers the potential to combine two types of attack detection models, to address the problem of suspicious review more comprehensively and effectively.

1.2. Research problem

In the framework of this thesis, in order to explore the potential of combining the fake review detection approaches and shilling attack detection approaches mentioned above, I want to find the answer to the question: *when users post both numeric ratings and text reviews on the same system, will the two approaches to detecting suspicious users achieve similar results?*

During the implementation of the experiment to answer the main question, some related issues will also be mentioned and studied:

- Is it possible to build detection models that simulate advanced models? If not, are there any sources that provide these models for research use?
- Are the available datasets diverse and reliable?

1.3. Research methodology

To answer the above research questions, I will conduct a synthesis experiment consisting of 3 main steps. This experiment is summarized in Figure 1.1. Details of the steps are as follows:

In the first section, I will build a simple fake reviews detection model, based on the structure of the most advanced models. This model will then be tested with the YelpZip dataset, which is a dataset used in many studies on detecting fake reviews. The results obtained will be compared with previously published results of the reference models.

In the next section, instead of building a completely new shilling attack model, I use a Python library with 5 detection algorithms in it. Two datasets available in the library will be used to test the algorithms. Then, the YelpZip dataset used in the previous section will be modified with different assumptions and conditions. These versions YelpZip dataset will then be used in the shilling attack detection library as input, in order to find creators of fake reviews.

Finally, the results obtained from the models will be analyzed and compared. The results obtained from the experiment between the YelpZip dataset and the shilling attack models will be used to determine the relationship between the two types of detection.

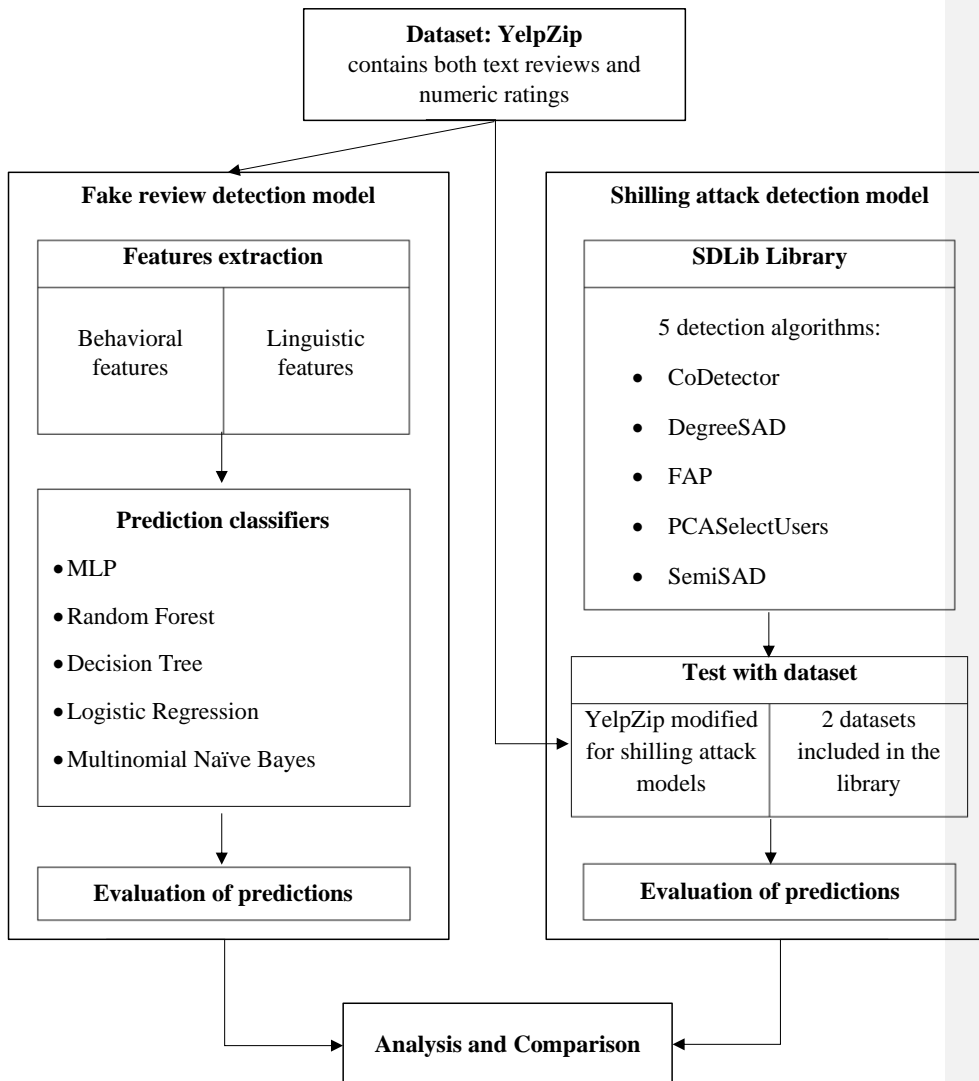


Figure 1.1 Research structure

2. Literature reviews

2.1. Fake reviews detection

2.1.1. Fake reviews definition

The essence of writing and reading reviews is to store and share experiences in the community of users. Later users will make decisions based on previous user experiences, expressed in the form of written language. So, in its most general form, fake reviews are reviews created with no such purpose in mind.

Jidal and Liu with their study published in 2007 [11] are considered to be the first founders of fake review detection studies. They classified fake reviews into 3 types as follows:

- Untruthful opinions describe users who post negative reviews to damage a product/business's reputation or post positive reviews to promote a product / business. These reviews are called fake or deceptive reviews, and they are difficult to detect simply by reading, as real and fake reviews are similar to each other.
- Reviews of a brand only describe those who are commenting on the brand of the products.
- Non-reviews that are irrelevant and offer no genuine opinion or are simply advertisements.

The last two types are called disruptive spam opinions. They can be identified easily by readers and do not cause serious threat. The first type receives more attention in research on fake reviews, because these reviews are written in natural language with the purpose of misleading the reader, so they are also more difficult to identify.

According to Heydari et al. [12], a fake review is a review posted by a user who has never owned or experienced the item. In fact, the motives for creating a suspicious review can sometimes be more complicated than that. For example, a user has personally used the product and had a relatively good impression but decided to write a review only because of the incentives they receive from the business. However, the methods of identifying and classifying fake reviews by researchers are more meaningful for building detection model.

2.1.2. Detection models

Modern fake review detection models make heavy use of machine learning [13]. Researchers often extract various features from the original dataset before including it in the detection

model. These features are usually divided into two main categories: behavioral features and textual features [14].

Behavioral features are parameters that represent the behavior of users in the past, regardless of the specific content of each textual review. Some of the behaviors that are easy to understand and commonly found in research studies include: number of reviews written, percentage of positive/negative reviews, first review ratio... These features are often tied to basic assumptions about the behavior of a fake user or spammer. For example, maybe they only post positive reviews, write a lot of reviews in a short time, or are the first to write a review for a newly launched product.

Text features, or structural features [15], linguistic features [16] are features extracted directly from each specific review. Some common text features are: bag of word, word embedding and semantic features. It can be seen that these features are determined to help the computer describe and understand information in language form, such as length, phrases, semantics, and even emotion.

Although most researchers divide features into the above two categories, there are a few specific features that are classified differently across studies. For example, *review length* belongs to the category of behavioral features in the detection model of N. Hussain et al [17]. However, a similar feature called *average review length* is classified as textual in [18].

Combining both types of features in fake reviews detection models has been shown to have better results than models using only one type of features [14]. In addition, the research by A. Rastogi et al. [18] found that behavioral features play a more important role than linguistic features.

In terms of methods to detect fake reviews, the survey [14] classified them into two main groups: Traditional statistical machine learning and Neural network. The details have been summarized in table 2.1.

Table 2.1 Fake review detection methods

Class	Sub-class	Examples of algorithm used	Result and comments
Traditional statistical machine learning	Supervised ML	Ensemble machine learning, Decision tree, Naïve Bayes, KNN, SVM...	Accuracy gained from 68% to 92% based on the algorithm and the dataset. Some methods have problems with the reasonableness of their assumptions and are effective when applied to other domains.
	Unsupervised ML	Unsupervised multi-iterative graph-based, unsupervised learning...	The experiments got similar results as supervised models, but the methods can be enhanced by combining different features. The assumption that duplicated reviews are fake are unreliable.
	Semi-supervised traditional statistical ML	Semi-supervised (SVM, NB, RF, LR, KNN, LDA, DT), multi-task method...	Difficulty of detecting fake reviews in cross-domain. Some methods only work with long or short text
Neural network (NN)	Convolutional neural network (CNN)	sentence weight NN, word-order preserving CNN, ...	CNN is proven to perform better than LSTM in a mixed domain, but it is not efficient for long text. Different behavioral and linguistic features can be used to increase the affectiveness.
	Recurrent neural network (RNN)	Long short term memory (LSTM), Bi – GTU with attention, Hierarchical CNN, ...	This method is highly efficient, but often requires high computational resources. One research found out that fake reviews expressed stronger emotions than real reviews.
	Generative adversarial network (GAN)	GAN, fakeGAN, Behaviour features generative adversarial network	They did not outperform the state-of the art methods. Working with cross-domain is still a challenge.
	Other neural network models	Pattern recognition, multi-dimensional time series, hierarchical fusion attention network	The results from the papers suggest that it is useful to use additional information like metadata or features related to emotional aspects.

Commented [SM1]: Try fitting the tables in one page, if possible.

The fake reviews detection model that I develop, as well as features extracted will be based mainly on the following 4 papers and models:

- The model introduced by Noekhah et al [15]. It uses both behavioral and structural features, as well as inter- and intra-relationship between user profiles. The list of

features has 24 behavioral features and 9 structural features in total. The author used the Amazon Review Dataset for the experiment and the algorithm was written in C#. They also tested with different sets of features and found out that the combinations of both types of features brought the best result, while using only group behavioral features gave the worst one.

- The model SRD-BM introduced by Hussain et al [17]. It uses behavioral features to label reviews in the Amazon dataset as spam and not spam. They use their own threshold values for each feature in this model, instead of feeding the features into machine learning models like in other research. Some of their behavioral features are classified as linguistic features in other papers (*review length* and *ratio of capital letters* for example). They then built a model called SRD-LM, using the n-gram tokenizing method to find spam reviews in the dataset that they labeled in the previous step.
- The research of Rastogi et al. [18]. They tested the effectiveness of two groups of features: metadata-based (behavioral) and content-based (textual). In addition to the features taken from previous models, they also introduced some new features like *maximum rating deviation* and *early rating deviation*. They tested the models with three settings: review-centric (fake reviews detection), reviewer-centric (fake users detection) and product centric (spam-targeted product detection). Two datasets used are YelpZip and YelpNYC. The result is that behavioral features are more effective than textual features at identifying spam.
- The fake reviews detection models developed by Gupta et. al, available on Github [19]. The model uses both behavioral and linguistic features. Different classifiers are used in their model include: multinomial Naive Bayes, K-Nearest Neighbour (KNN), Random Forest, Convolutional Neural Network (CNN) and CNN with Long short term memory (LSTM).

2.2. Shilling attacks

2.2.1. Shilling attacks definition

A shilling attack is an attack against recommendation systems, with the goal of causing these systems to produce biased results in favor of the attacker. An attacker does this by injecting fake ratings and profiles into the system, usually to promote or demote an item or a brand.

To understand more about shilling attack, the concept of recommender system (RS) also needs to be introduced and clarified. RS is a system that predicts content and items that its users might be interested in. Internet users can now experience the benefits of RS through video

recommendations on youtube, auto-generated playlists on spotify, or recommended items in addition to what they search for on Amazon. Campana et al. [20] divides RS into four categories, based on the underlying assumptions and their mechanisms:

- Collaborative filtering. This is the most popular RS, and also used a lot in research about shilling attacks. The assumptions behind it is that users with similar interests will give similar ratings for the same item.
- Content-based. This kind of RS suggests items that are similar to the items that a user liked on selected in the past.
- Graph-based. Relationships between users and items are represented by nodes. Nodes are then connected by edges. PageRank from Google is the most popular example of this RS.
- Context-Aware. A type of RS that use information related to specific situation when a user is interacting with the system.

From that information about RS, shilling attacks are then evaluated and classified by different criteria. The three most common criteria are required knowledge, attack intent, and cost [21]:

- Required knowledge. It indicates the level of knowledge about a particular RS needed for an attack. When an attacker needs to understand the process and how ratings are distributed in a RS, it is a high-knowledge attack. In contrast, a low-knowledge attack requires only basic independent knowledge and can be obtained from public information sources.
- Attack intent. A push attack is when an attacker enters the system with fake profiles and ratings to promote a certain item. Nuke attacks, on the other hand, aim to demote an item.
- Cost. The attackers need to determine whether the benefits are enough to offset the necessary costs of the attacks. Factors that affect costs include size of attack (the more fake accounts created, the higher the cost), the degree of difficulty encountered in interacting with RS (a system that requires users to verify information or identity before rating items will make the attacks more costly), the amount of knowledge required, and any other resources for the attack.

In the studies about shilling attacks, the basic components of an actual user and a fake user in RS have the following form:

Table 2.2 Typical Profiles of a Normal User (a) and a Fake User (b)

Normal user's profile					
Item	Item ₁	Item ₁	...	Item _{m-1}	Item _m
Rating	R ₁	R ₂		-	-

Fake user's profile							
	Selected items		Filler items			Unrated items	Target items
Item	I _{s1}	I _{s2}	I _{f1}	I _{f2}	...	I _{m-1}	I _m
Rating	$\delta(I_{s1})$	$\delta(I_{s2})$	$\sigma(I_{f1})$	$\sigma(I_{f2})$...	Null	$\gamma(I_m)$

In Table 2.2, Selected items are the items that are voted to make the profile similar to some genuine users; Filler items are those that are rated to disguise the profile as a genuine user; Unrated items are the items that have not received ratings from the profile and Target items are the items that the attacker wants to promote or demote.

The most common RS attack strategies differ based on how profile sections are filled. Below is information on the six most common attack strategies that have been appeared in the latest surveys and research about shilling attacks [21] [22] [23] [24].

- *Random Attack*: the ratings for filler items are random in this attack strategy. If it is a push attack, maximum ratings are used for target items. In contrast, minimum ratings are used if it is nuke attacks. This is the simplest attack strategy and requires the least knowledge of the system, so its effectiveness is lower than other attacks.
- *Average Attack*: filler items are rated by the average value. Although this model is more efficient than Random Attack, it requires cost to compute mean and some extra insights.
- *Bandwagon Attack*: the attacker makes fake profiles resemble real profiles by giving maximum ratings for very common items, and random ratings for a subset of items.
- *Segment Attack*: items are targeted to a certain group of users, who are likely to be interested in the items being pushed. For example, a seller of microwave ovens might want their product recommended to people who love to cook and often buy kitchen products.
- *Love/Hate Attack*: minimum ratings are given to the target items and maximum rating values are for the filler items.

- *Reverse Bandwagon Attack*: in contrast to Bandwagon Attack, items that are rated low by the majority of users are used as Selected items, and the attacker will give low ratings to target items in order to cause the system to make a low prediction for them.

The summary of how the most common types of attacks work has been presented in the research by Rezaimehr [22], summarized in Table 2.3:

Table 2.3 Six Popular Shilling Attack Strategies

Attack type	Rating strategy			
	Selected items	Filler items	Unrated items	Rated items
Random	ϕ	Normal distribution around system mean	Determined by filler size	$i_t = r_{\min}/r_{\max}$
Average	ϕ	Normal distribution around mean rating value for $i \in I_F$	Determined by filler size	$i_t = r_{\min}/r_{\max}$
Bandwagon	Max rating for popular items	Normal distribution around mean rating value across the whole database	Determined by filler size	$i_t = r_{\max}$
Segment	Max rating for popular items	r_{\min}	Determined by filler size	$i_t = r_{\max}$
Love/hate	ϕ	r_{\max}	Determined by filler size	$i_t = r_{\min}$
Revers Bandwagon	Least popular items rated r_{\max}	Random ratings with a normal distribution around the mean rating I in I_F	ϕ	$i_t = r_{\min}$

It should be noted that, modern RSs use both numeric ratings and text reviews as input, both shilling attack and fake written reviews negatively affect RS. However, the assumption in RS studies is that fake profiles with text ratings are inserted in large numbers, while this is difficult to do with text reviews due to the high cost.

2.2.2. Shilling attack detection models

Over the years, many methods of detecting attacks on RS have been proposed and studied.

Si et al. [25] classified ways to detect shilling attacks based on whether the algorithm is supervised or not. For supervised classification, the main algorithms used are kNN, C4.5 and

SVM, combined with the above classification attributes to detect attack profiles. For unsupervised methods, attack detection models use algorithms such as k-means clustering or principal component analysis.

In the research conducted by Rezaimehr et al. [22], 25 models for RS attack detection are analyzed and classified into four categories, based on the types of algorithms used in the methods. The categories are: clustering, feature extraction, probabilistic and classification as shown in the Figure 2.1. Clustering is a commonly used method, assuming that the fake profiles have similar characteristics and can be put in a cluster, separated from the real profiles. Clustering algorithms used include K-nearest neighborhood (K-NN), K-means, K-medoid, and Hierarchical. In Feature extraction method, features like WDA, LengthVar, DegSim [26] are extracted and classified to find attack profiles. Probabilistic methods use algorithms such as Bayes classifier or Bayesian network model to build detection models. For the classification method, researchers use classification algorithms, combined with the techniques like similarity analysis and target item analysis [27] to find fake profiles.

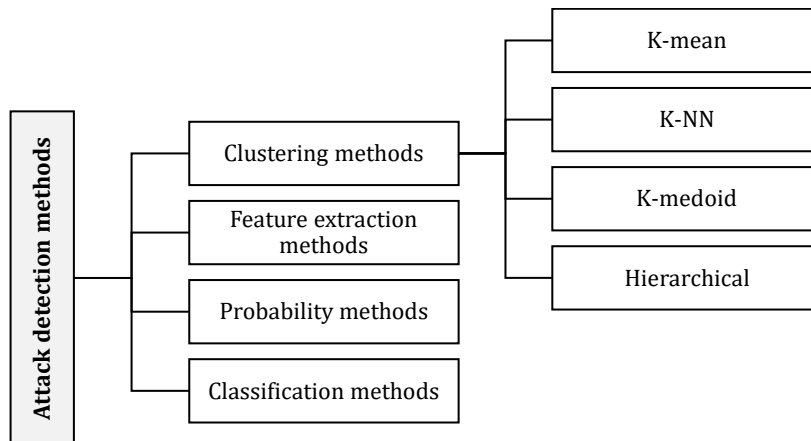


Figure 2.1 Categorized Algorithms used in Shilling Attacks Detection Models

Besides understanding RS attack detection models, it is also important to study how these models are evaluated. These contents will be mentioned next, with brief information about the datasets and the evaluation measures used by the researchers. This information is extracted from the survey [22].

In 2018, Dou et al. introduced a supervised algorithm to detect shilling attack [29]. Along with the research, they also developed a Python library to test and compare their algorithm with four

other algorithms. This library is published by Yu, member of the research team on Github [28]. The details of the 5 models used in this library are summarized as follow:

- CoDetector: Developed by Dou et al. [29], with the idea that fake profiles work in groups when trying to promote an item to increase the impact of an attack. The model uses basic Matrix factorization to find out the relationship between users and items, then combines with the word embedding model to explore the user context structure. The results from their experiment show that this model provides much better accuracy compared to contemporaneous shilling attacks detection models.
- SemiSAD: or Semi-supervised learning based Shilling Attack Detection, introduced by Cao et al. [30]. The researchers' assumption for this model is that, the number of labeled data is much smaller than that of unlabeled data in real life. This is considered the first shilling attack detection model with the first semi-supervised algorithm when they announced it. It learns from both labeled and unlabeled user profiles, using naïve Bayes classifier and EM- λ . EM- λ is used to increase the weight of labeled data in the dataset.
- PCASelectUsers: Developed by Mehta [31]. PCA stands for Principal Component Analysis. It is a multivariate analysis technique used to find the intrinsic size of high dimensional data by projecting it onto a low dimensional space where the selected axes capture the greatest variation in the data [32].
- FAP: a label propagation algorithm for spam user detection, proposed by Zhang et al. [33]. The spam probability of a user is calculated by combining of the spam probability of its adjacent products and the connections (edges) between the user and the corresponding products. The spam probability for each product is also calculated by considering all its adjacent users. This procedure is then conducted back and forth and recursively.
- DegreeSAD: Li et al. [34] developed this algorithm by deriving different features from properties of item popularity in user profiles, then exploit machine learning classifiers to detect and remove attackers.

2.3. Datasets

The survey [3] has listed the most popular datasets used in papers about shilling attacks detection. The details are presented in Table 2.4

Table 2.4 The Datasets used in RS Attack Detection papers

	User	Item	Rate	Type of rating
Movielens	943	1682	100,000	1 – 5
Jester	73,421	100	4.1 million	- 10 to 10
Amazon review	645,072	136,785	1,205,125	1 – 5
Netflix	4334	3558	3558	1 – 5
Each movie	2000	1623	137,425	1 – 5
Book crossing	77,805	185,973	433,671	1 - 10

The survey conducted by Mohawesh et al. [14] has listed the most popular public datasets used by researchers to test different fake review detection models. The top 6 datasets from this survey are summarized in Table 2.5.

Table 2.5 The Datasets used in Fake Reviews Detection papers

Dataset	Construction method	Number of reviews	Number of reviewers	Domain	Text review	Numeric Rating
Yelp CHI	Filtering algorithm	67,365	38,063	Restaurants and hotels	✓	✓
Yelp NYC	Filtering algorithm	359,052	160,25	Restaurants	✓	✓
Yelp ZIP	Filtering algorithm	608,598	260.277	Restaurants	✓	✓
Yelp Consumer Electronic	Rule-based Technique	18,912		Consumer electronic	✓	✓
Amazon	Rule-based technique	5.8 milluon	2.5 million	Products, DVD, music, Books...	✓	✓
TripAdvisor	Amazon Mechanical Turk	3,032		Hotels, restaurants, doctors	✓	

For two experiments with both fake reviews detection models and shilling attack detection models, I need to find a dataset that has both numeric ratings and text reviews. Therefore, two datasets Amazon and Yelp have been studied more closely.

First, Amazon datasets were investigated. In a pioneering study on fake review detection in 2007 [11], researchers Jindal and Liu labeled three types of fake reviews in the Amazon dataset. This dataset contains 5,838,032 reviews and 55,319 reviews have been identified as fake. However, this dataset couldn't be found online when this thesis is written. Ni publishes "Amazon Review Data (2018)" [35] with more than 233 million reviews. However, this is an unlabeled dataset. Hussain made the dataset "Amazon Product Review (Spam and Non Spam)" available on Kaggle.com [36]. Although this dataset is labeled by researchers [17], I have found that all fake reviews are rated 4-5 stars. The paper by Rout et al. [37] also pointed out that this dataset is more suitable for big data approaches than traditional machine learning models.

Yelp datasets are the next options for my experiment. I chose YelpZip [38] as it has more reviews than YelpChi and YelpNYC. Besides the original source on ODDS website, this dataset is also shared on Kaggle.com [39]. This dataset is first used by Rayana and Akoglu in their research in 2015 [40]. It contains 608,598 reviews for restaurants in the US. There is information about products and users, as well as timestamp, ratings, and the text reviews. The dataset contains reviews that have been filtered through a Yelp algorithm, dividing them into two groups: recommended and unrecommended. Although Yelp's algorithm is secret, and possibly imperfect, it has been proven to produce accurate results [38]. These two review groups are respectively considered genuine and fake in the dataset.

For the above reasons, I choose the YelpZip dataset for my experiments on fake reviews detection models and shilling attack detection models in the next part of this thesis.

3. Fake review Detection Models

Through my research, I found that although there are many studies introducing fake review detection models, the programming part of these models is almost impossible to find. The only shared model that is relevant to my study is that of Gupta et al [19]. However, this model is difficult to manage and manipulate. Therefore, I decided to build a brand-new fake review detection machine learning model. This model is simpler to the ones from the reference studies, but it contains most of the features found in modern models. The of process of analyzing the dataset, extracting features, building and testing the model are presented in this chapter 3.

3.1. Dataset Exploration

The YelpZip dataset has 5 psv files, namely: userIdMapping, productIdMapping, reviewContent, reviewGraph and metadata. 3 files are used to create a final combined dataset to be used in this experiment. The details of their structures are shown in Table 3.1.

Table 3.1 Information of 3 files in YelpZip dataset

File name	Field	Description	Datatype
metadata	user_id	ID of users	int
metadata	prod_id	ID of restaurants	int
metadata	rating	rating from 1 to 5	factor
metadata	label	-1 if fake review 1 if real review	factor
metadata	date	format: YYYY-MM-DD	date
reviewContent	user_id	ID of users	int
reviewContent	prod_id	ID of restaurants	int
reviewContent	date	format: YYYY-MM-DD	date
reviewContent	review	text review	string
productIdMapping	pro_name	Restaurant name	string
productIdMapping	prod_id	ID of restaurants	int

The final merged dataset looks like this in the Figure 3.1:

```
[ ] df = pd.merge(metadata, reviewContent, on=['user_id', 'prod_id', 'date']).merge(productIdMapping, on=['prod_id'])
df.head()
```

	user_id	prod_id	rating	label	date	review	prod_name
0	5044	0	1.0	-1	2014-11-16	Drinks were bad, the hot chocolate was watered...	Toast
1	5045	0	1.0	-1	2014-09-08	This was the worst experience I've ever had a ...	Toast
2	5046	0	3.0	-1	2013-10-06	This is located on the site of the old Spruce ...	Toast
3	5047	0	5.0	-1	2014-11-30	I enjoyed coffee and breakfast twice at Toast ...	Toast
4	5048	0	5.0	-1	2014-08-28	I love Toast! The food choices are fantastic -...	Toast

Figure 3.1 Screenshot of the merged YelpZip dataset

This dataset contains 608,458 reviews, with 80,439 reviews labeled with -1 (fake reviews), representing 13% of the total reviews.

The number of fake reviews with a rating greater than 3 is 56344, accounting for 70% of the total number of fake reviews. The distribution of of real and fake reviews by rating is shown in Figure 3.2, with most of ratings are positive with 4 and 5-star.

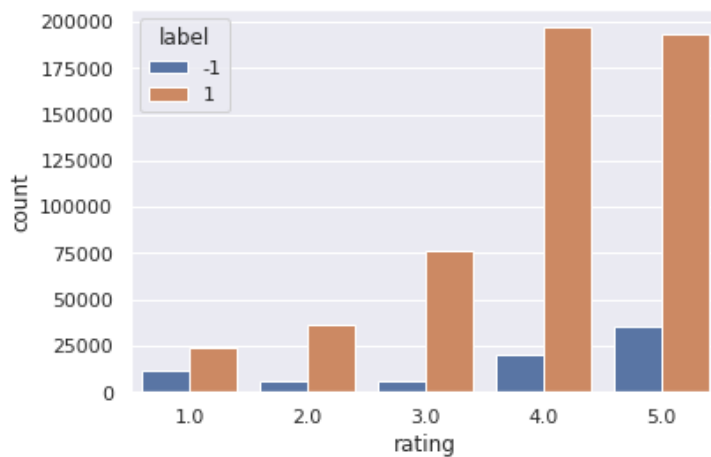


Figure 3.2 Distribution of ratings group by user types in YelpZip

The total number of reviewers in the dataset is 260,239, 24% of which are authors of fake reviews. It should be noted that not all users write only fake reviews or real reviews. The number of people believed to be the author of both fake and genuine reviews in the dataset is 2121.

There are a total of 5,044 restaurants, 4,336 of which received fake reviews. From there, the number of restaurants that only receive genuine reviews is 708.

Wordcloud is also used to visualize commonly used words in real and fake reviews. The two python libraries used for this part are PIL and wordcloud. Figure 3.3(a) is wordcloud generated from 100,000 real reviews, figure 3.3(b) is the wordcloud generated from all 80,439 fake reviews in the dataset. It can be seen that the difference between the words appearing in the 2 obtained wordclouds is not clear. This proves that distinguishing fake reviews from real reviews requires more sophisticated and advanced methods.



Figure 3.3 Wordclouds for fake reviews (a - dark background) and genuine reviews (b - bright background)

3.2. Features Extraction

As described in the literature review, modern fake review detection models use both behavioral and linguistic features. Therefore, I will build a simple model with these 2 types of features extracted from the dataset. The features are based on the studies [15], [17], [18], [19]. Some features are determined the same or similar to the referenced models, some are completely new in my model. Details are presented in the following section.

Six behavioral features have been extracted:

- *Review Count (RW)*: Calculate the number of reviews written by each user. The assumption here is that the number of user reviews can influence the user's "reliability" (e.g. a person who writes a lot of reviews is likely to be a spammer).

- *Activity Window (AW)*: calculate the difference between the timestamps of the first and last reviews of an author. The assumption here is that new users are more likely to be the author of fake reviews, especially when they post a lot of reviews in a short period of time. Trusted users, on the other hand, have been on the platform for a certain amount of time and post reviews relatively regularly.
- *Number of Positive Ratings (PR)*: Calculate the number of ratings with a value greater than 3 for each user. A similar version of this feature is “The ratio of Positive review”, used in the study [18], derived from the research of Rayana and Akoglu [40]
- *Number of Negative Ratings (NR)*: Calculate the number of ratings with a value lower than 3 for each user.
- *Mean product rating (MPR)*: Calculate the mean of all numeric ratings for each user.
- *Order of reviews*: Find out the order of a review among all the reviews for a product/restaurant. The assumption is that early reviews are more likely to be a spam one. This feature is altered from the feature “First position review” in research [15] and “The ratio of first review” in [17], which ignores the reviews in the 2nd or 3rd place etc. of a product.

Four linguistic features are extracted:

- *Word Count (WC)*: calculates the number of words in each review. The assumption is that a review is more likely to be fake if it is way too short, since the attackers has limited time. A variation of this feature is the review length, which is used in [17]. Accordingly, the number of characters is counted instead of the number of words.
- *The Ratio of Capital Letters (RCL)*: Calculate the ratio of uppercase letters to total letters in a review. The assumption that reviews are written entirely in capital letters, or that use more uppercase letters than usual, is more suspicious.
- *The ratio of digits (RD)*: Calculate the ratio of digits/numbers to total letters in a review.
- *Sentiment analysis features extracted using VADER*: include Positive score, Negative score, Neutral Score and Compound.

VADER stands for Valence Aware Dictionary and sEntiment Reasoner. It is a lexicon and rule-based sentiment analysis tool and it has been proved to be very suitable for the analysis of opinions on social media [41]. VADER is often used for sentiment analysis data in linguistic form, because it can quantitatively calculate the level of positive or negative emotions of a text or paragraph.

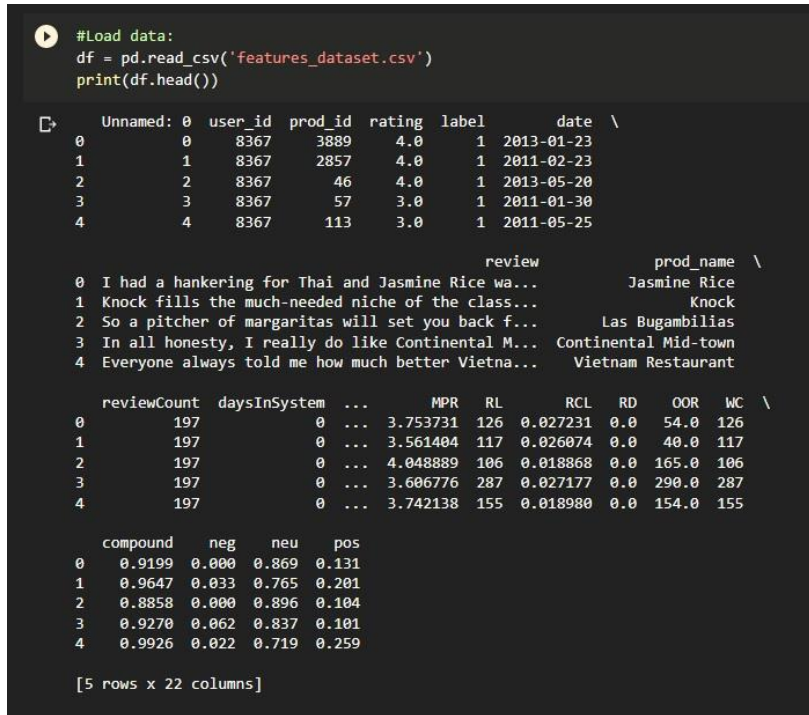
VADERS produces four scores: Postive, Negative, Neutral and Compound. Compound is the sum of the first 3 scores after being normalized between -1 and 1. Accordingly, 1 means extremely positive and -1 means extremely negative.

Table 3.2 gives information about the number of referenced studies or models that my model's features appear in. Note that there are some features that may not be exactly the same that are still considered present in different papers (e.g., number of uppercase letters and ratio of uppercase words, review length and word count).

Table 3.2 Features used in this thesis and in four reference fake review detection models

Features in the thesis	Noekhah et al. [15]	Rastogi et al. [18]	Hussain et al. [17]	Gupta et. al. [19]
Review Count (RW)	✓		✓	✓
Activity Window (AW)			✓	
Number of Positive Ratings (PR)		✓	✓	
Number of Negative Ratings (NR)		✓	✓	
Mean product rating (MPR)	✓			✓
Order of Reviews (OOR)				
Word Count (WC)		✓		✓
The Ratio of Capital Letters (RCL)			✓	✓
The ratio of digits (RD)				✓
VADER scores (negative, positive, neutral, compound)	✓	✓		✓

The dataset that includes all the features now looks like this in Figure 3.4:



```
#Load data:
df = pd.read_csv('features_dataset.csv')
print(df.head())
```

	Unnamed: 0	user_id	prod_id	rating	label	date	\
0	0	8367	3889	4.0	1	2013-01-23	
1	1	8367	2857	4.0	1	2011-02-23	
2	2	8367	46	4.0	1	2013-05-20	
3	3	8367	57	3.0	1	2011-01-30	
4	4	8367	113	3.0	1	2011-05-25	

	review	prod_name	\
0	I had a hankering for Thai and Jasmine Rice wa...	Jasmine Rice	
1	Knock fills the much-needed niche of the class...	Knock	
2	So a pitcher of margaritas will set you back f...	Las Bugambillas	
3	In all honesty, I really do like Continental M...	Continental Mid-town	
4	Everyone always told me how much better Vietna...	Vietnam Restaurant	

	reviewCount	daysInSystem	...	MPR	RL	RCL	RD	OOD	WC	\
0	197	0	...	3.753731	126	0.027231	0.0	54.0	126	
1	197	0	...	3.561404	117	0.026074	0.0	40.0	117	
2	197	0	...	4.048889	106	0.018868	0.0	165.0	106	
3	197	0	...	3.606776	287	0.027177	0.0	290.0	287	
4	197	0	...	3.742138	155	0.018980	0.0	154.0	155	

	compound	neg	neu	pos
0	0.9199	0.000	0.869	0.131
1	0.9647	0.033	0.765	0.201
2	0.8858	0.000	0.896	0.104
3	0.9270	0.062	0.837	0.101
4	0.9926	0.022	0.719	0.259

[5 rows x 22 columns]

Figure 3.4 Screenshot of dataset with all the features extracted in Python notebook

3.3. Preprocessing

First, missing values are checked and removed. Missing data, if present, will reduce the effectiveness of the analysis and may cause machine learning models to fail. In this experiment, there were missing data in the NR (Number of Negative Ratings) feature, because the programming for this feature returned NULL when no negative ratings were counted. These missing values have been replaced with the value 0.

Next, the features and label values are separated. 25% of the data is used for the test set. Train set and test set are created from dataset by `train_test_split()` function. Then, The features values are normalized using `MinMaxScaler()`. Normalization is a necessary step because it helps data in different fields appear similar, without eliminating differences between values in the same field. Thus, it increases the quality of the dataset and makes machine learning models more efficient. The range 0 - 1 is selected so that the Multinomial Naïve Bayes classifier can run properly, since it does not accept negative input values.

Finally, the unbalanced data problem is handled using the Random Over-sampling technique. It functions by generating duplicates of the data in the minority group (specifically the fake reviews) randomly in the training set. This is reasonable, assuming that many of the fake reviews are duplicates of each other. After using `RandomOverSampler()`, the number of samples in the training set increased from 456,343 to 492,116.

3.4. Prediction Models

Five different classifiers were chosen to build a fake review detection model. These include: MLP, Decision Tree, Random Forest, Multinomial Naïve Bayes and Logistic Regression. Details of these classifiers are as follows:

- *MLPClassifier*: The Multi-layer Perceptron (MLP) is a feed forward artificial neural network model. It relies on an underlying Neural Network to map input data to a set of output labels (classification task). An MLP includes 3 types of layers: input layers, hidden layers and output layers, with each layer is fully connected to the following one. The nodes of the layers are neurons with nonlinear activation functions, except for the nodes of the input layer. The neurons in the MLP are trained with the back propagation learning algorithm [42]. In my model, *MLPClassifier* was used with 100 hidden layers, activation function "relu" and learning rate = 0.002.
- *Decision Tree*: is a non-parametric supervised learning method. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data feature, and it can be used for both classification and regression task. A decision tree is built with different nodes and branches. Each node specifies a test on an attribute, each branch represents different possible values for that attribute from a node [43].
- *Random Forest*: is also a supervised machine learning algorithm. In the "forest", a number of decision trees are built on different samples. An additional randomness is also added. In *RandomForest*, Instead of splitting a node to find the most important feature, it searches for the best feature among a random subset of features. This results in a wide diversity and a better model [44].
- *Multinomial Naïve Bayes*: it is a Bayesian learning approach popular in Natural Language Processing (NLP). It brings an alternative to the heavy AI-based semantic analysis and effectively simplifies textual data classification [45].
- *Logistic Regression*: Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. In our experiment the outcomes are binary

outcome: fake or real. Logistic regression has been proven to be a useful analysis method for classification problems [46].

All the above 5 classifiers were run to find the best model. The results and analysis are presented in the next section.

3.5. Result and Evaluation

I choose four indicators: Accuracy, AUC, Specificity and Sensitivity to evaluate the effectiveness of fake review detection models. The three indicators Accuracy, Specificity and Sensitivity are calculated from the confusion matrix, with the formula presented in the table 3.3.

Table 3.3 Confusion matrix (a) and the related evaluation metrics (b)

CONFUSION MATRIX			
Actual class	Predicted class		
		Class = Yes	Class = No
	Class = Yes	True positive (a)	False negative (b)
	Class = No	False positive (c)	True negative (d)

Measure	Formula
Accuracy	$a = \frac{a + d}{a + b + c + d}$
Specificity	$p = \frac{d}{b + d}$
Recall/Sensitivity	$r = \frac{a}{a + b}$

AUC stands for "Area under the ROC Curve." An ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate. AUC gives an aggregate measure for the performance among all possible classification thresholds [47]. In this case, AUC gives the probability that the model predicts a review as real instead of fake.

When considering the meaning of the three metrics calculated from the confusion matrix, Accuracy measures the ability of predicting correctly both types of reviews, Specificity measures the ability of predicting correctly genuine reviews, and Sensitivity measures the ability of correctly predicting fake reviews.

Table 3.4 Results of review detection models using 5 different classifiers

Classifier	Accuracy	AUC	Specificity	Sensitivity /Recall
MLP	0.7041	0.7384	0.6916	0.7852
RandomForest	0.8603	0.5987	0.9547	0.2426
Decision Tree	0.8109	0.5846	0.8927	0.2766
Logistic Regression	0.6102	0.7150	0.5725	0.8574
Multinomial Naïve Bayes	0.5148	0.6842	0.4536	0.9146

From Table 3.4, it can be seen that all classifiers gave relatively good results. However, as accuracy increases, Sensitivity tends to decrease. Accuracy of Random Forest is very good at 86%, however, its sensitivity is very low at only 24.2%. Multinomial Naive Bayes gives the lowest accuracy with 51.5%, but has the highest sensitivity with 91%. MLP is the classifier that gives the most balanced results between accuracy and sensitivity, with these two figures being 70.4% and 78.5% respectively.

Since the number of genuine reviews is much larger than the number of fake reviews in the test set, Sensitivity becomes a more meaningful score. Sensitivity therefore is chosen as the score to find the best model, as it shows the ability of detecting fake reviews. The classifier that gave the highest Sensitivity is Multinomial Naïve Bayes, therefore is considered the best one.

Table 3.5 compares the model in my experiment with the best models in the referenced papers. Although it can be seen that research groups choose different figures to measure the performance of their models, this table shows that my model is relatively efficient.

Table 3.5 Comparison between the fake review detection built in the thesis and 4 reference models

	My model (Multinomial Naïve Bayes)	S. Noekhah et al [15]	A. Rastogi et al. [18]	Naveed Hussain et al [17]	A. Gupta et. al [19] (Logistic Regression)
Accuracy	0.5148	0.89	-	0.885	0.6056
AUC	0.6842	-	-	0.895	-
Specificity	0.4536	-	-	-	-
Sensitivity	0.9146	-	0.9004	0.873	0.76

Commented [EP2]: Try to put this on the next page.

4. Shilling attack detection models

In the previous sections, we have seen that it is possible to create a machine learning model to detect fake reviews in the YelpZip dataset. In this section, I will test whether users who write fake reviews are considered attackers by detection models of shilling attacks. Instead of building a completely new model, this time I will use Yu's SDLib library with 5 models of shilling attacks. First, the two datasets available in this library will be analyzed and applied to the models. Then different versions of YelpZip will be created and included in the models. The results will then be analyzed and compared.

4.1. Testing SDLib library with the dataset provided in the library

There are 3 datasets available in the SDLib library: *amazon*, *averageattack* and *filmtrust*. The first two datasets both contain two files: one with user ratings for items, the other with labels for the users (whether they are real users or attack accounts). The *filmtrust* dataset contains two files: "ratings.txt" contains the ratings and the file "trust.txt" contains information about the relationships between the users. Since this dataset is different from the YelpZip dataset, which has no information about the interactions between accounts, I will test the shilling attack models with only two datasets *amazon* and *averageattack*.

4.1.1 Data exploration

First, I created a python notebook to perform statistical analysis on these two datasets. The results obtained are as follows:

Amazon Dataset:

There are 5,055 users, or profiles, in the amazon dataset. In "labels.txt", the attacking profiles are labeled with the value 1, the genuine ones are labeled the value 0. The total number of attack profiles is 1,973, accounting for 38% of the profiles. The second file, "profiles.txt", contains a total of 51,346 ratings from all the profiles for different items. Ratings have a value of 1 to 5 stars. The majority of ratings are positive ratings, with 61% of 5-star ratings and 25% of 4-star ratings. Meanwhile, the percentage of both 1 and 2 stars are only 4%.

Regarding the number of ratings of each profile in the Amazon dataset, there are only 5 profiles with the number of ratings greater than 100. The maximum number of ratings from a profile is 239 and the minimum is 1. Regarding the frequency of the rating numbers, the number of profiles that posted 4 ratings is the largest with 556 profiles. The numbers of profiles that have posted 5 and 6 ratings are also large, with values of 508 and 463 profiles respectively. The number of profiles that give from 3 to 10 ratings was 3353, accounting for 63.5% of the total number of profiles.

The ratings of the attack profiles are also analyzed and compared with the genuine profiles. Figure 4.1 shows the distribution of ratings for two groups of fake profiles (label 1) and genuine profiles (label 0). Whereby, the total number of ratings generated by attack profiles is 17,989, representing 35% of the total number of ratings. The percentage of 4- and 5-star ratings from fake profiles is extremely high, at 99.2% the total number of ratings generated by fake profiles, while this number for genuine profiles is only 79.6%. The average number of ratings from a fake profile is 9.1, just slightly below the same statistic from genuine profile of 10.8. However, the average number of 5-star ratings from a fake profile is 7.4, much higher than this index for a genuine profile at 5.4.

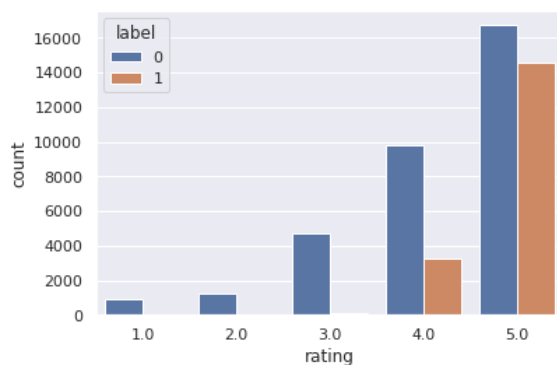


Figure 4.1 Distribution of ratings for fake profiles (1) and genuine profiles (0) in Amazon dataset

AverageAttack dataset

The same method is also applied to the AverageAttack analysis. The dataset includes 2 files, "labels.txt" and "ratings.txt". The first file contains the labels of 1,658 profiles, with 150 profiles identified as attackers, accounting for 9% of the profiles. The second file consists of 44,825 ratings from profiles for different items. There is one difference from the amazon dataset, which is that the lowest value of a rating in this dataset is 0.5. A rating score is higher than the previous one by 0.5 instead of 1. So, from 0.5 to 5, there are a total of 9 values that a profile can choose to rate an item. The number of ratings with points 3 and 4 accounts for the largest proportion the total ratings, at 54%. The percentage of absolute ratings 5/5, however, only accounts for 0.3%.

In this dataset, there are 18 profiles that created more than 100 ratings, of which profile with ID 272 has the largest number of ratings at 244. The number of profiles with 50 ratings is the largest at 197. The percentage of profiles that generated from 1 to 10 ratings is 34%.

The two groups of profiles and fake and genuine are then also separated for comparison. The distribution of ratings for these two groups of profiles is shown in Figure 4.2. It can be seen that 3-star ratings account for the largest proportion of ratings coming from fake profiles. This is expected in a dataset that simulates an average shilling attack. The average number of ratings for a fake profile is 62.2, while this number for a genuine profile is only 23.5.

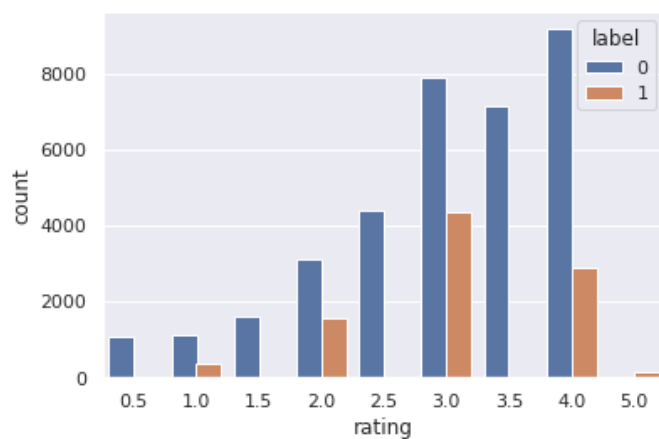


Figure 4.2 Distribution of ratings for fake profiles (1) and genuine profiles (0) in AverageAttack dataset

Commented [EP3]: Put this on the next page. Don't separate heading only at the end of a page.

Comparison

Commented [EP4]: See my previous comment about location of the heading on a page.

The basic statistics of both datasets are listed in the Table 4.1. It is obvious that, the numbers of ratings in the two datasets are similar (more than 30,000). However, the percentage of attack profiles in Amazon is much larger than in the AverageAttack dataset (38% vs. 9%). The average numbers of ratings are also quite large, from 9.1 review per profile. This figure in AverageAttack is up to 62 for the attacking profiles. This leads to the question about the importance of the average rating numbers in detecting shilling attacks. This will be clarified somewhat, after these two datasets are fed into SDLib's algorithms.

Table 4.1 Statistics of Amazon dataset and AverageAttack dataset

Statistics	Amazon Dataset	AverageAttack Dataset
Number of profiles	5,055	1,658
Number of genuine profiles	3,118	1,508
Number of attack profiles	1,937	150
Proportion of attack profiles	38%	9%
Number of ratings	51,346	44,825
Number of ratings by genuine profiles	33,357	35,494
Number of ratings by attack profiles	17,989	9,331
Average number of ratings by a genuine profile	10.8	23.5
Average number of ratings by an attack profile	9.1	62.2
Highest number of ratings by one profile	239	244
Highest number of profiles with the same number of ratings	556 (4 ratings)	197 (50 ratings)

4.1.2 Testing shilling attack detection models in SDLib with 2 datasets

First, all 5 algorithms in SDLib are tested with the Amazon dataset. The results are shown in Table 4.2. In this test, only 3/5 algorithms worked. The two algorithms FAP and

PCASelectUser returned a memory error. All three other algorithms including CoDetector, DegreeSAD and SemiSAD have all shown to be effective in identifying shilling attacks. However, CoDetector is the best one with Precision and Recall for both classes above 80%. In the SDLib library, these two figures are calculated separately for each class, but their meaning remains the same. Accordingly, precision is the ratio between the correct predictions to the total number of predictions for a class, while recall is the ratio between the correct predictions and the actual number of samples in that class.

Table 4.2 Results of SDLib algorithms using Amazon dataset

Amazon Dataset					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	0.8962	0.8666	0.8812	877
	1	0.8063	0.8470	0.8261	575
DegreeSAD	0	0.7936	0.8490	0.8204	616
	1	0.7103	0.6264	0.6657	364
FAP	0	-	-	-	-
	1	-	-	-	-
PCASelectUser	0	-	-	-	-
	1	-	-	-	-
SemiSAD	0	0.7042	0.8045	0.7510	583
	1	0.6426	0.5100	0.5687	402

The AverageAttack dataset was then used to test those 5 algorithms, with the results listed in the Table 4.3. This time, all 5 algorithms ran without errors. However, there is a difference in the efficiency between these 5 algorithms. FAP and SemiSAD barely detect attack profiles. Meanwhile, PCASelectUser gives extremely accurate results. Specifically, recall for class 0 (genuine profiles) is approximately 99%, and recall for class 1 (attack profiles) is up to 100%.

Table 4.3 Results of SDLib algorithms using AverageAttack dataset

AverageAttack Dataset					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	0.9713	0.9667	0.9690	420
	1	0.6889	0.7209	0.7045	43
DegreeSAD	0	0.9804	0.9836	0.9820	305

	1	0.8000	0.7692	0.7843	26
FAP	0	0.9183	0.5280	0.6705	1447
	1	0.0905	0.5000	0.1533	136
PCASelectUser	0	1.0000	0.9894	0.9947	1508
	1	0.9036	1.0000	0.9494	150
SemiSAD	0	0.9140	1.0000	0.9551	287
	1	0.0000	0.0000	0.0000	27

When comparing the two tests above, it is easy to see that the datasets have a significant influence on the effectiveness of the shilling attack detection models in the SDLib library. Only two algorithms CoDetector and DegreeSAD were stable in both tests.

4.2 Testing SDLib library with YelpZip dataset

Now, I will conduct an experiment to see whether the users who generate fake reviews in YelpZip dataset are considered attack profiles in shilling attack detection models. Since YelpZip only has labels for reviews, users need to be labeled before they can be loaded in the SDLib library.

In the first scenario, I used the entire YelpZip dataset for the test. All users who create at least 1 fake review are labeled as fake users, regardless of the number of genuine reviews they have posted. The number of users and ratings therefore remained unchanged, at 260,277 and 608,598, respectively. There are 62,228 users labeled as fake, accounting for 24%.

The results returned from 5 algorithms to detect shilling attacks are shown in the Table 4.4. Out of the 5 algorithms, only one DegreeSad algorithm works with the YelpZip dataset. The remaining 4 algorithms all encountered errors or froze and did not produce results. This is because the YelpZip dataset is much larger than the Amazon and AverageAttack dataset that we tested in the previous section. The results obtained by the DegreeSAD algorithm are also very low, with Precision and Recall for class 1 (fake users or attack profiles) being 0.22 and 0.05 respectively.

Table 4.4 Results of SDLib algorithms using YelpZip dataset (Scenario 1)

YelpZip Dataset: Scenario 1					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	-	-	-	-
	1	-	-	-	-

DegreeSAD	0	0.7602	0.9478	0.8437	39620
	1	0.2211	0.0472	0.0778	12435
FAP	0	-	-	-	-
	1	-	-	-	-
PCASelectUsers	0	-	-	-	-
	1	-	-	-	-
SemiSAD	0	-	-	-	-
	1	-	-	-	-

In the second test, I tried to reduce the size of the dataset by excluding users with less than 10 ratings. This is also to eliminate fake users with low numbers of ratings. From Table 2.2 in the literature review section, it is shown that typical profile in the shilling attack model needs a certain number of ratings in the "baskets": Selected items, Filler items and Target items. Thus, fake users with not many ratings might be difficult to detect. The good results from the tests with Amazon and the AverageAttack dataset also support this hypothesis. The method of labeling fake users remains the same in this scenario. The dataset now has 8,453 users, with 329 fake users, representing 4% of the total users. The rating numbers is 173,448, which is only about 1/3 of the original YelpZip dataset.

Table 4.5 shows the results obtained in the 2nd Scenario. This time, all 5 algorithms worked. The similarity between them is that all 5 algorithms return very high Precision and Recall for class 0 (genuine profiles), from over 80% to over 90%. These two figures for class 1 (fake profiles) are very low, only about 2% to 7%. When comparing the results of DegreeSAD algorithm in both scenarios, there is a small improvement in Recall for class 1, but it is not significant and is only about 3%.

Table 4.5 Results of SDLib algorithms using YelpZip dataset (Scenario 2)

YelpZip Dataset: Scenario 2					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	0.9627	0.9440	0.9533	2430
	1	0.0490	0.0729	0.0586	96
DegreeSAD	0	0.9639	0.9527	0.9583	1627
	1	0.0610	0.0794	0.0690	63
FAP	0	0.9620	0.8163	0.8832	7970

Commented [EP5]: Put the reference number

	1	0.0406	0.1944	0.0672	319
PCASelectUsers	0	0.9600	0.8989	0.9285	8123
	1	0.0296	0.0760	0.0426	329
SemiSAD	0	0.9647	0.8219	0.8876	1662
	1	0.0358	0.1803	0.0598	61

I found that the percentage of fake users of the scenario 2 dataset was much lower than that of Amazon and the AverageAttack dataset (4% vs 38% and 9%). So, in the 3rd test, I randomly eliminated a number of genuine users in the 2nd YelpZip dataset, leaving the number of fake users at 20% of total users number. The dataset now has 329 fake users and 1,300 genuine users, with a total of 33,682 reviews. The results obtained in Table 4.6 show a marked improvement for Precision and Recall of class 1 compared to scenario 2.

Table 4.6 Results of SDLib algorithms using YelpZip dataset (Scenario 3)

YelpZip Dataset: Scenario 3					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	0.8147	0.7647	0.7889	391
	1	0.2640	0.3267	0.2920	101
DegreeSAD	0	0.8115	0.7416	0.7750	267
	1	0.1481	0.2069	0.1727	58
FAP	0	0.7285	0.4987	0.5921	1173
	1	0.1118	0.2534	0.1551	292
PCASelectUsers	0	0.7981	0.9000	0.8460	1300
	1	0.2025	0.1003	0.1341	329
SemiSAD	0	0.8119	0.9250	0.8648	280
	1	0.1600	0.0625	0.0899	64

From what I learned in the previous 3 scenarios, I decided to create a 4th dataset from the original YelpZip dataset with the following rules: Retain users with at least 10 ratings, all users with fake reviews are considered as fake users, a number of genuine users are randomly eliminated to keep the ratio of fake users to total users at 35% (up 15% from scenario 3). The final dataset has a total of 938 users and 18533 ratings.

The results figures in the Table 4.7 again show the improvement in the performances of the shilling attack detection models. The two supervised algorithms CoDetector and DegreeSad gave the best results, with about 50% of fake users detected.

Table 4.7 Results of SDLib algorithms using YelpZip dataset (Scenario 4)

YelpZip Dataset: Scenario 4					
Method	Class	Precision	Recall	F1-Score	Support
CoDetector	0	0.7243	0.6872	0.7053	195
	1	0.4455	0.4900	0.4667	100
DegreeSAD	0	0.7481	0.7778	0.7626	126
	1	0.5000	0.4590	0.4786	61
FAP	0	0.5842	0.5256	0.5534	508
	1	0.2397	0.2857	0.2607	266
PCASelectUsers	0	0.6445	0.8933	0.7488	609
	1	0.3085	0.0881	0.1371	329
SemiSAD	0	0.6667	0.9120	0.7703	125
	1	0.4211	0.1231	0.1905	65

Table 4.8 summarizes the statistics of datasets in 4 scenarios, as well as the best results in each test. The figures again show the effect of dataset on the performance of the shilling attack detection algorithms. In particular, the dataset should not be too large, so that the algorithm can be run in a testing environment. In addition, the algorithms give more accurate results when the percentage of fake profiles and the average ratings per profile are larger.

Table 4.8 Dataset statistics and prediction results in 4 testing scenarios using YelpZip

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Number of profiles	260,277	8,453	1,629	938
Number of genuine profiles	198,049	8,124	1,300	609
Number of attack profiles	62,228	329	329	329
Proportion of attack profiles	24%	4%	20%	35%
Number of ratings	608,598	173,448	33,682	18,533
Number of ratings by genuine profiles	520,702	167,407	27,641	12,498

Number of ratings by attack profiles	87,896	6,041	6,041	6,041
Average number of ratings by a genuine profile	2.6	20.6	21.3	20.5
Average number of ratings by an attack profile	1.4	18.4	18.4	18.4
Highest number of ratings by one profile	197	197	136	136
Highest number of profiles with the same number of ratings	170,098 (1 rating)	1,120 (10 ratings)	223 (10 ratings)	136 (10 ratings)
Best algorithm in SDLib (based on Recall of class 1)	DegreeSAD	FAP	CoDetector	CoDetector
Precision (Class 1)	0.2211	0.0406	0.2640	0.4455
Recall (Class 1)	0.0472	0.1944	0.3267	0.4900

5. Analysis and comparison

In chapters 2 and 3, we have seen that it is difficult to find pre-built fake review detection models for research. However, it is possible to build a new machine learning model based on the latest models and studies. My model using 6 behavioral features and 4 linguistic features has given good results. In particular, the Sensitivity obtained from the model was up to 91% in the experiment with the YelpZip dataset. This shows the effectiveness of my model in detecting fake text reviews. Considering that the reviews were labeled by a complex system built by Yelp, it can also be interpreted that my fake review detection model somewhat simulates how Yelp classifies reviews of its users.

In chapter 4, instead of building a completely new shilling attack detection model, I focused on studying the performances of the algorithms in the SDLib library. The algorithms were tested with different datasets, first with 2 default datasets Amazon and AverageAttack, then a few versions of YelpZip. Although the algorithms have been proven effective in separate studies, my experiment shows that they only work well if the dataset is suitable. Specifically, 2/5 algorithms did not work with Amazon dataset. The SemiSAD algorithm also gave a relatively low Recall result (only 51%) for the group of fake profiles for this dataset. For the

Commented [EP6]: Check the numbers against your numbers in Chapter 3. Do you have 7 linguistic features? You only explained 4 in Chapter 3.

AverageAttack dataset, the algorithms perform much better, except SemiSAD completely fails to detect attack profiles.

Comparing the two datasets Amazon and AverageAttack led to an assumption that the dataset used in shilling attacks models should have a relatively large percentage of attack profiles (9% - 38%), and average rating number of users must be high. This assumption was proven correct after the experiment between SDLib and YelpZip were performed. Specifically, through each adjustment of the YelpZip dataset in the above direction, the results obtained are getting higher and higher.

The result of the above experiment, with about 50% of fake reviews owners identified by shilling attack models, suggesting a connection does exist between these two types of attacks. Better experimental results could be possible, if there is a good standard dataset built for this research direction.

The ability to combine the two types of attack detection approaches is therefore, in my opinion, feasible. A system that uses both types of attack detection models can be as follows: when a user is new and has not posted many ratings and reviews, the shilling attacks detection model would be less effective, so a fake reviews detection model should be preferred. When the numbers of reviews and ratings from a user increase to a certain number, determining of suspicious users is likely to be accomplished by shilling attacks detection models and the two types of models can be used together.

6. Conclusion

In this thesis, the way fake review detection models and shilling attack detection models work has been studied. A fake review detection model has been built with Accuracy up to 86% and Recall up to 91% in the experiments with YelpZip dataset. The effectiveness of shilling attack detection algorithms in SDLib library has been demonstrated through experiments with two given datasets, thereby finding out the importance of dataset for the performances of these models. Finally, these algorithms are used to find the creators of fake reviews in several different versions of the YelpZip dataset. The obtained results confirm once again the importance of the dataset in the evaluation of shilling attack detection algorithms. In additions, several ways to build a good dataset have been found. The result of 50% fake reviews creators detected by SDLib also shows a connection between the two types of attacks as well as two directions of attack detection, opening an opportunity to combine these two types of models.

In the future, more modern fake reviews and shilling attack detection models can be investigated. These models can be built on deep learning instead of machine learning, or big data instead of traditional datasets with limited capacity. New datasets with a sufficiently large sample size and reliably classified can also be developed. Next, fraud detection models that synthesize two research directions can be built.

References

- [1] Tourism Teacher, "How does the hotel star rating system work?," Tourismteacher.com, 28 05 2022. [Online]. Available: <https://tourismteacher.com/hotel-star-rating-system/>. [Accessed 02 06 2022].
- [2] R. Feloni, "How the Michelin Guide made a tire company the world's fine dining authority," BusinessInsider, 24 10 2014. [Online]. Available: <https://www.businessinsider.com/history-of-the-michelin-guide-2014-10>. [Accessed 02 06 2022].
- [3] D. Kaemingk, "Online reviews statistics to know in 2022," Qualtrics, 30 10 2020. [Online]. Available: <https://www.qualtrics.com/blog/online-review-stats/>. [Accessed 02 06 2022].
- [4] "How COVID-19 Changed Online Reviews," PowerReviews, [Online]. Available: <https://www.powerreviews.com/blog/how-covid-19-changed-online-reviews/>. [Accessed 02 06 2022].
- [5] E. McAfee, "Why Amazon's Crackdown On Fraudulent Sellers Is A Win For Everyone," Forbes, 7 7 2021. [Online]. Available: <https://www.forbes.com/sites/forbesbusinesscouncil/2021/07/07/why-amazons-crackdown-on-fraudulent-sellers-is-a-win-for-everyone/?sh=5ca1200e9697>. [Accessed 02 06 2022].
- [6] Amazon News, "Twitter," 22 02 2022. [Online]. Available: <https://twitter.com/amazonnews/status/1495854588964098053>. [Accessed 02 06 2022].
- [7] S. Hollister, "Amazon says it's permanently banned 600 Chinese brands for review fraud," theverge.com, 17 09 2021. [Online]. Available: <https://www.theverge.com/2021/9/17/22680269/amazon-ban-chinese-brands-review-abuse-fraud-policy>. [Accessed 02 06 2022].
- [8] D. Berthiaume, "chainstorage.com," 05 09 2022. [Online]. Available: <https://chainstoreage.com/amazon-shuts-down-fake-review-brokers>. [Accessed 02 06 2022].
- [9] Yelp Inc., "Survey finds what makes reviews trustworthy — consumers want more than ratings," Yelp, 07 12 2021. [Online]. Available: <https://blog.yelp.com/news/survey-finds-what-makes-reviews-trustworthy-consumers-want-more-than-ratings/>. [Accessed 02 06 2022].

- [10] Which, "How to spot a fake review," which.co.uk, 13 01 2022. [Online]. Available: <https://www.which.co.uk/reviews/online-shopping/article/online-shopping/how-to-spot-a-fake-review-aiDaS3e1ivfr>. [Accessed 02 06 2022].
- [11] N. Jindal and B. Liu, "Analyzing and detecting review spam," in *International Conference on Data Mining (ICDM 2007)*, 2007.
- [12] A. Heydari, M. Tavakoli and N. Salim, "Detection of fake opinions using time series," *Expert Systems with Applications*, vol. 58, pp. 83-92, 2016.
- [13] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter and H. A. Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, vol. 2, no. 1, pp. 1 - 24, 2015.
- [14] R. Mohawesh., S. Xu, S. N. Tran, R. Ollington, M. Springer, Y. Jararweh and S. Maqsood, "Fake Reviews Detection: A Survey," *IEEE Access*, vol. 9, pp. 65771-65802, 2021.
- [15] S. Noekhah, N. H. Zakaria and N. Salim, "A Novel Model for Opinion Spam Detection Based on Multi-Iteration Network Structure," *Advanced Science Letters*, vol. 24, no. 2, pp. 1437-1442, 2018.
- [16] F. Abri, L. F. Gutierrez, A. S. Namin, K. S. Jones and D. R. W. Sears, "Fake Reviews Detection through Analysis of Linguistic Features," in *IEEE International Conference on Machine Learning Applications (ICMLA 2020)*, Miami, Florida, 2020.
- [17] N. Husain, H. T. Mirza, I. Hussain, F. Iqbal and I. Memon, "Spam Review Detection Using the Linguistic and Spammer Behavioral Methods," *IEEE Access*, vol. 8, pp. 53801-53816, 2020.
- [18] A. Rastogi, M. Mehrotra and S. S. Ali, "Effective Opinion Spam Detection: A Study on Review Metadata Versus Content," *Journal of Data and Information Science*, vol. 5, no. 2, p. 76–110, 2020.
- [19] A. Gupta, A. Gawad, A. Hule, G. Shanbhag, V. Kadam and P. Gupta, "Yelp-dataset-Fake-Reviews," 22 04 2019. [Online]. Available: <https://github.com/guptaa3/Yelp-dataset-Fake-Reviews>. [Accessed 03 04 2022].
- [20] M. G. Campana and F. Delmastro, "Recommender Systems for Online and Mobile Social Networks: A survey," *Online Social Networks and Media*, vol. 3, pp. 75-97, 2017.
- [21] S. K. Lam and J. Riedl, "Shilling Recommender Systems for Fun and Profit," in *The 13th International World Wide Web Conference (WWW2004)*, New York, USA, 2004.

- [22] F. Rezaimehr and C. Dadkhah, "A survey of attack detection approaches in collaborative filtering recommender systems," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2011-2066, 2021.
- [23] J. Wang and Q. Tang, "Recommender Systems and their Security Concerns," University of Luxembourg, Luxembourg, 2015.
- [24] I. Gunes, C. Kaleli, A. Bilge and H. Polat, "Shilling attacks against recommender systems: a comprehensive survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 767-799, 2014.
- [25] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: a review," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 291-319, 2020.
- [26] F. He, X. Wang and B. Liu, "Attack detection by rough set theory in recommendation system," in *IEEE international conference on granular computing (GrC)*, 2010.
- [27] Z. Yang, Z. Cai and X. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems," *Knowledge-Based Systems*, vol. 111, pp. 144-158, 2016.
- [28] J. Yu, "SDLib," [Online]. Available: <https://github.com/Coder-Yu/SDLib>. [Accessed 02 06 2022].
- [29] T. Dou, J. Yu, Q. Xiong, M. Gao, Y. Fang and Q. Song, "Collaborative Shilling Detection: Bridging Factorization and User Embedding," in *Collaborative Computing: Networking, Applications and Worksharing*, Edinburgh, UK, 2017.
- [30] J. Cao, Z. Wu, B. Mao and Y. Zhang, "Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system," *World Wide Web*, vol. 16, no. 05, pp. 729-748, 2013.
- [31] B. Mehta, "Unsupervised Shilling Detection for Collaborative Filtering," *AAAI*, pp. 1402-1407, 2007.
- [32] I. T. Jolliffe, "Principal Component Analysis," *Technometrics*, vol. 45, no. 3, p. 276, 2003.
- [33] Y. Zhang, Y. Tan, M. Zhang, Y. Liu, C. Tat-Seng and S. Ma, "Catch the Black Sheep: Unified Framework for Shilling Attack Detection Based on Fraudulent Action Propagation," in *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, 2015.

- [34] W. Li, M. Gao, H. Li, J. Zeng, Q. Xiong and S. Hirokawa, "Shilling attack detection in recommender systems via selecting patterns analysis," *IECE Transactions on Information and Systems*, vol. 99, no. 10, pp. 2600-2611, 2016.
- [35] J. Ni, "Amazon Review Data (2018)," [Online]. Available: <https://nijianmo.github.io/amazon/index.html>. [Accessed 03 02 2022].
- [36] N. Hussain, "Amazon Product Review (Spam and Non Spam)," 2022. [Online]. Available: <https://www.kaggle.com/datasets/naveedhn/amazon-product-review-spam-and-non-spam>. [Accessed 2022 02 02].
- [37] J. K. Rout, A. Dalmia, S. K. Rath, B. K. Mohanta and A. H. Gandomi, "Detecting Product Review Spammers Using Principles of Big Data," *IEEE Transactions on Engineering Management*, pp. 1 - 12, 2021.
- [38] L. A. Shebuti Rayana, "YelpZip dataset," [Online]. Available: <http://odds.cs.stonybrook.edu/yelpzip-dataset/>. [Accessed 02 02 2022].
- [39] M. (Username), "Yelp Dataset," [Online]. Available: <https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/discussion/156218>. [Accessed 02 02 2022].
- [40] S. Rayana and L. Akoglu, "Collective Opinion Spam Detection: Bridging Review Networks and Metadata," in *The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney NSW Australia, 2015.
- [41] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *The Eighth International AAAI Conference on Weblogs and Social Media*, Michigan, USA, 2014.
- [42] S. Abirami and P. Chitra, "The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases," *Advances in Computers*, vol. 117, no. 1, pp. 339-368, 2020.
- [43] "Decision Tree Classifier," Science Direct, [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/decision-tree-classifier>. [Accessed 07 06 2022].
- [44] N. Donges, "Random Forest Algorithm: A Complete Guide," Built In, 14 04 2022. [Online]. Available: <https://builtin.com/data-science/random-forest-algorithm>. [Accessed 07 06 2022].
- [45] A. V. Ratz, "Multinomial Naïve Bayes' For Documents Classification and Natural Language Processing (NLP)," 17 05 2021. [Online]. Available:

- <https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6>. [Accessed 07 06 2022].
- [46] T. W. Edgar and D. O. Manz, Research Methods for Cyber Security, 2017.
- [47] "Classification: ROC Curve and AUC," Google Developers, 10 02 2020. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>. [Accessed 08 06 2022].
- [48] N. Hussain, H. T. Mirza, A. Ali 1, F. Iqbal 2, I. Hussain and M. Kaleem, "Spammer group detection and diversification of customers' reviews," *PeerJ Computer Science*, pp. 7:e472 DOI 10.7717/peerj-cs.472, 2021.
- [49] F. Singh, "Sentiment Analysis Made Easy Using VADER," Analytics India Magazine Pvt Ltd, 8 12 2020. [Online]. Available: <https://analyticsindiamag.com/sentiment-analysis-made-easy-using-vader/>. [Accessed 2 5 2022].
- [50] "sklearn.tree.DecisionTreeClassifier," [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>. [Accessed 02 05 2022].
- [51] K. v. Abrams, "Global Ecommerce Forecast 2021," Insider Intelligence; eMarketer, 2021.
- [52] T. Kumari and P. Bedi, "A comprehensive study of shilling attacks in recommender systems," *International Journal of Computer Science Issues (IJCSI)*, vol. 14, no. 4, pp. 44 - 50, 2017.