UNIVERSITY OF NAIROBI

Faculty of Science and Technology

Department of Computing and Informatics

CONTEXT AWARE COMPUTATIONAL TRUST MODEL FOR ROBUST AND ACCURATE RECOMMENDER SYSTEMS ALGORITHMS FOR ECOMMERCE PLATFORMS

By

EDWIN OUMA NGWAWE

REG NO: P80/54974/2019

SUPERVISORS:

DR. ELISHA O. ABADE

DR. STEPHEN MBURU

Thesis submitted in partial fulfillment of the requirements for the award of the Doctorate Degree in Computer Science.

Definition of Terms

Online service – a service available on or performed using the Internet or other computer network

Trust – perceived belief that a party will deliver on his obligation.

Scale development – Creation of a set of questionnaire items which can be used to measure a latent variable.

Recommender systems – software tools used to help a user choose a suitable item online

DECLARATION

Student Declaration

This thesis is my original work and has not been presented for award of any degree in any University

23/05/2022

Signature
Edwin Ouma Ngwawe

Date

Supervisors

This thesis has been submitted for examination with my approval as the University Supervisor

04/06/2022

Signature
Dr. Stephen Mburu

Date

03/06/2022

Signature
Dr. Elisha Abade.

Date

Abstract

Online shopping has become part and parcel of our lives and more so as aggravated by the emergence of COVID-19 pandemic which necessitated need for social distancing and also work from home. This has led to unprecedented rise in online shops and consequently a myriad of alternatives for shoppers to consider before committing to a purchase. The myriad of alternatives has put a tall order on users in terms of information overload during decision making and made some shoppers to just rely on guesswork, putting them at a danger of losing income or lives to unscrupulous vendors. It is prudent to have a way of evaluating the how trustworthy an online shop is beforehand in order to assist the buyers to make meaningful decisions in time. In this study, we create a scale to estimate how trustworthy an online service provider is. We carry out a survey and then use factor analysis to come up with a model for estimating trustworthiness of an ecommerce platform from the consumer perspective. 2104 valid responses were attained from a total of 3,244 responses received from Google form whose link was shared directly to participant by reaching to them physically. The trust scale was then taken through reliability and validity tests. Confirmatory factor analysis yielded four components, which are security, privacy, deception and reliability. Cronbach's alpha is found to be 0.956. The model is tested empirically and is found to improve the robustness and prediction accuracy of collaborative recommendation algorithms significantly.

Contents

CHAPTER ONE: INTRODUCTION

1.0 Background

With the improvement of computing technologies, online shopping has become a new normal to the way of life. Aggravated by the global pandemic, COVID 19, the need for social distancing has further increased the demand for online shopping as people try to stay or work at home but still run their day-to-day errands, online. This has indeed come with its own challenges. The massively online phenomenon has led vendors to try to maximize their profits by following the people and also place their products on the online market place, and they have placed their products online massively. This has led to a myriad of alternatives and thereby bringing along information overload to the online users. Indeed it is not practical for an average user to asses all possible options unaided and end up purchasing optimally online. Unscrupulous vendors have realized this sudden shift to the online market and are trying to take advantage of shoppers by luring them using various digital marketing techniques and finally ending up defrauding them. There exist mechanisms to help shoppers choose the right product or services, such as the use of recommender systems, which help users to choose suitable items. The loophole again is that the recommender systems depend on historic data to predict the choices. This data can still be manipulated by unscrupulous practitioners such as by use of profile injection. This loophole have brought about trust issues and the statistical mechanisms behind them are too efficient to be detected by the conventional anomaly detection mechanisms, neither do they involve breaking into an IT system but are so effective that they influence the mathematical properties of unaided recommender system in such a way that misleads unsuspecting online shoppers in to making unsuitable decisions and commitments to purchase. This brings about existential trust issues or elevates perceived risk in e-commerce which needs to be taken care of. (Zait and Bartea, 2011), (Lake et al., 2021), (Shani and Gunawardana, 2011), (Burke, O'Mahony and J., 2011)

1.1 Problem statement

With the growing speeds and computing power of Information technology (IEEE, 2011),(White, 2012), the connectivity and integration of various computing services have emerged, including automation of shopping experiences. The amount of data being processed and presented to the user at any given time is huge in volume, leading to a problem of information overload. This is aggravated by the global pandemic, COVID 19, which necessitated the need for physical distancing and working from home.

Recommender systems can help to alleviate this burden of decision making from the users(Ricci, Rokach and Shapira, 2011). Much as collaborative filtering is seen as a successful technique for recommender systems (Koren and R, 2011), it still emerges that malicious vendors manipulate the outcome of a recommender system unfairly to their advantage hence leaving a room for distrust in recommender systems(Victor, Cock and Cornelis, 2011), (O'Donovan and Smyth, 2005). Depending on the interest, an adversary can perform product push, which involves unfairly promoting a product which otherwise would not be the most suitable for the shopper or product nuke which involves demoting the product which is indeed fairly the most suitable for the shopper. These two situations can easily be achieved by an attack to the mathematical properties of the recommendation algorithm such as profile attack(Burke, O'Mahony and J., 2011). Left unabated this vulnerability in the collaborative recommendation engines can cause inconveniences, economic loss or even loss of life to the online shopper and therefore has brought about trust issues or elevated the perceived risk in e-commerce (Zait and Bartea, 2011).

1.2 Purpose

The purpose of this research work is to develop a scale for estimating trustworthiness of an online shop in a context aware fashion. This will be used to improve the robustness of collaborative recommendation engines. It is anticipated that the scale will help filter out untrustworthy online shops and protect the online shoppers from potential fraud, and thereby not only making online shopping a safe exercise but also, improve the online shopping experience since shoppers will get suitable items recommended to them and will trust that it is indeed the most suitable item and therefore will shop with a peace of mind.

1.3 Hypothesis

We hypothesize that the developed trust scale will improve the robustness and the prediction accuracy of a collaborative recommendation algorithm.

1.3.1. Null Hypothesis for Robustness

H_o: Autonomous trust model has no significant effect on the robustness of a collaborative recommendation algorithm.

H_1: The autonomous trust model has positive significant effect on the robustness of a collaborative recommendation algorithm.

1.3.2 Null Hypothesis for Prediction Accuracy

H_o: Autonomous trust model has no significant positive effect on prediction accuracy of a collaborative recommendation algorithm.

H_1: The autonomous trust model has a significant positive effect on the prediction accuracy of a collaborative recommendation algorithm.

1.3.3 Testing the hypothesis

We used the one tailed paired two sample for means t-test from the Data analysis Tool pack plug-in in Microsoft Excel to compute the p-values, σ.

1.4 Research Objectives

In this context, objectives are a set of (S - Specific, M - Measurable, A - Achievable, R-Realistic/Relevant, T- Time bound/boxed) activities, whose deliverables will be put together to amount to the achievement of the purpose of this research project.

1.4.1 General Objective

The general objective of a project is the goal or aim of the project. The goal of this work is to improve the robustness and accuracy of collaborative recommender algorithm in the wake of increased online presence.

1.4.2 Specific Objectives

1) To determine the indicators of trust in online services
2) To develop a model for estimation of trustworthiness of an online shop.
3) To embed the new trust model as a new parameter, called trust adjustment factor, into the classical collaborative recommendation algorithm so as to create a new trust enhanced algorithm.
4) To deploy the new algorithm into an empirical setup for proof of concept.
5) To assess the impact of the new trust parameter on collaborative recommendation engine as far as the robustness and prediction accuracy are concerned.

1.5 Research Questions

The research questions are:

1) What are the indicators of trustworthiness in online shop from the perspective of a Kenyan online shopper?
2) How can we estimate the trustworthiness of an online shop beforehand?
3) How can incorporate the estimated trust parameter into existing collaborative recommendation algorithm to make it more robust and accurate?

4) How can we deploy the new trust enhanced algorithm into an empirical set up for production purposes?
5) What is the impact of the new trust parameter on the robustness and prediction accuracy properties of collaborative recommendation algorithm?

1.6 Scope

The scope of this work is about how to improve the robustness and accuracy of collaborative recommendation algorithm. It involves finding the indicators of trust, creating a model to aggregate the trust indicators into one value to be used as an additional parameter in the recommendation algorithm, and then designing a setup for deploying the new algorithms into an empirical production set up and finally analyzing the impact of the new parameter on robustness and prediction properties of the collaborative recommendation algorithms.

1.7 Significance

The results of our research will be helpful in:

1) Contribute to the scientific body of knowledge a scientific model to estimate trustworthiness automatically for computational use.
2) Improve online experience for users because trust enhanced recommendation algorithm is more accurate in providing recommendations thus aiding users in making decisions aptly.
3) Mitigate the possible economic loss when a successful attack is made against the mathematical properties of the naïve collaborative recommendation algorithm.
4) Abate the danger of losing lives when a successful attack is made against the mathematical properties of the naïve collaborative recommendation algorithm.
5) Discuss the impact of trust parameter to properties of recommendation algorithm.

1.8 Limitations

The key limitation this study is that the output is a data driven solution which is a context-aware model and for it to be deployed in a different context, then another study will be done to bring in data for the specific context. Therefore replicating the model to another context requires more efforts than in the case of a non-context-aware model.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction to Recommender Systems

According to (Ricci, Rokach and Shapira, 2011), recommender systems are tools used to assist a user to choose a suitable item amidst a myriad of alternatives. They are used to improve online experience by helping user discover new items of interest and also solve problems of information overload by quickly helping the user to aptly choose one item amongst the millions of the options available online. Example empirical applications of recommender system include Facebook friend suggestion, Youtube video suggestion, Google and yahoo search advertising. They all choose a suitable product depending on the user profile as per the figure one below.

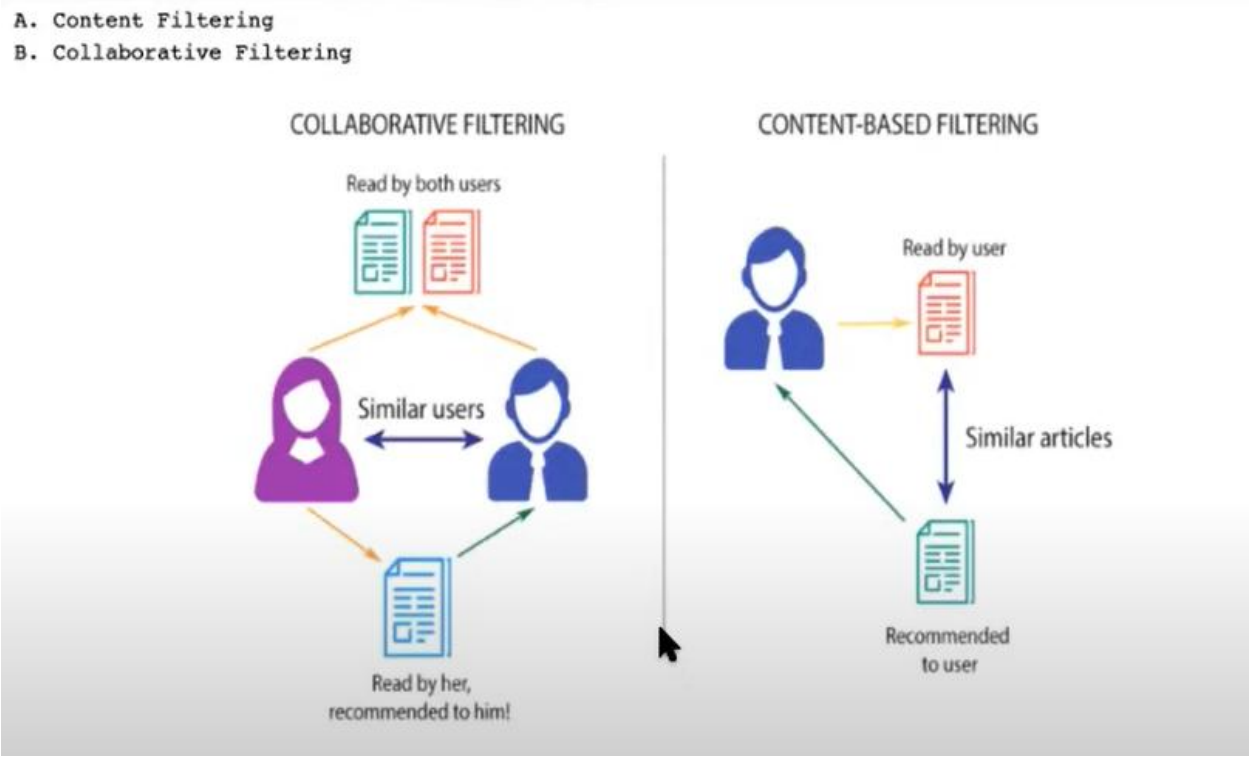


Figure 1 Recommender system working principles

Table 1 Example Utility Matrix

	Avator	LoTR	Matrix	Pirate
Alice	1		2	
Bob		5		3
Carol	2		1	
David				4

In the utility matrix shown in table 1, we can see that the users have rated some movies and not others. It could be that the users have not watched these movies which they have not rated or they may have watched the movies but just did not bother to rate the movies or any other reason.

In general, most utility matrices are going to be as sparse as this one since many users may actually not have watched many movies, or many users may just have chosen not to provide the ratings.

The key objective of a recommender system is to figure out these unknown values.

According to (Leskovec, Rajaraman, & Ullman, 2014), the challenges with recommender systems are:

- ✓ Gathering the known ratings
  - How do we collect the data in the utility matrix
- ✓ Estimating unknown ratings from the known ratings that have been gathered
- ✓ Evaluating the ratings estimation methods

About gathering ratings, users can be asked to explicitly ask the user to rate items on the site. Even though this has the advantage of simplicity, it does not scale up well as only a few users will leave ratings on the items they have purchased. Most users will buy an item or service but will not come back to leave a rating. This form of data acquisition is excellent since it is more accurate but often insufficient so in many cases we need to complement it with implicit rating. Implicit rating involves learning ratings from user actions, for example, in an online shopping

website; a purchase implies a high rating. Even though implicit ratings can be scaled up, one can never learn negative ratings with this approach. In practice, most recommender systems use both implicit and explicit ratings.

To solve the second problem of extrapolating utilities, the following approaches are used:

- ✓ Collaborative filtering
- ✓ Content based filtering
- ✓ Latent factor filtering

The third problem is solved mostly by checking the accuracy of the created recommendation engine.

2.1.2 Types of Recommender Systems

According to (Lu et al., 2014), (Ricci, Rokach and Shapira, 2011), (Hwang and Chen, 2006) recommender systems can be classified according to the technique used to implement the recommendation and also according to the application domain.

The most widely used recommender systems are content-based collaborative filtering recommender systems, below we discuss the principles of operation.

(i) Content-based recommendation

Using this approach, the system recommends items to the active user, the items which are similar to the items which have been rated highly by the active user in the previous interactions.

The plan of action is:

- ✓ Find the items the user likes
- ✓ Build the item profiles
- ✓ Build user profile, inferred from items profile
- ✓ Match user's profile to the catalog
- ✓ Find similar items in the catalog
- ✓ Recommend these items to the user.

To build item profile, for example in a text document, one may use term frequency – inverse document frequency (TD-IDF) (Leskovec, Rajaraman, & Ullman, 2014).

To build user profile, we consider weighted averages because a user may like some items more than others.

To match the profiles, we use similarity estimates such as the cosine similarity.

The advantages of content based approach include:

- No need of data from other users
- Able to recommend new and unpopular items
- Able to give explanation for their recommendation which boosts user confidence, this will be discussed in the next section on properties of recommender systems.

Content based approach has some disadvantages such as:

- Finding appropriate features of some items is hard, for example, finding features of images, movies and music so as to box them into, say, genres.
- Overspecialization, this means that it never recommends items outside the user's profile. This is also about serendipity which is discussed in the next section, about properties of a good recommender system.
- Cold Start problem for new users. Recommender system require some historic data on order to learn patterns so that they can predicct with acceptable accuracy. A good recommender system shoud not require too much data in order to attain its optimal levels of prediction. This property of recommender system is also known as coverage. See properties of recommender systems section.

(ii) Collaborative filtering approach

In this approach, we have two forms of filtering, namely user-user filtering and item-item filtering

In user – user filtering, the main idea is that: considering a user x, we to find a set N of users whose ratings are similar to user x ratings. We then estimate user x's ratings based on the ratings of users in set N. This set of users in N is called the neighborhood of user x.

To get similar users we can use cosine similarity (Lahitani, Permanasari and Setiawan, 2016) , Jaccard Similarity Index (Niwattanakul et al., 2013) or Pearson correlation (also known as centered-cosine)(Sheugh and Alizadeh, 2015).

These similarity calculation techniques have been described in section 2.2.

Again in item-item collaborative filtering, we can use same similarity metrics and prediction functions as for the user-user model.

It is always confusing for beginners to get the difference between the three approaches mentioned above but the core discriminating factors is that for content-based filtering, there is an some form of predicting items based on individual user's preference based on his own individual previous preferences, whereas for user-user collaborative filtering, the prediction is based on other user's preferences who are deemed similar to the current user while for item-item collaborative filtering, there is no involvement of the current user and items are rated absolutely on the ratings of items that are deemed similar to the item in question.

Item-item vs. user-user collaborative filtering.

In theory, user-user model and item-item collaborative filtering models looks similar and alternative approaches, however, in practice, item-item approach outperforms user-user approach in many cases. This is because items are 'simpler' than users. This is to say items have smaller sets of features while users have a variety of tastes so items similarities become more meaningful than user similarities. For example, a user may listen to music in a given genre when in a certain mood and again listen to another genre when in another different mood which makes profiling him/her a bit complex if not less meaningful.

Other than the techniques used to offer recommendation, recommender systems can also be classified according to their domains of applications. (Lu et al., 2014) discusses how to classify recommender systems based on domains of application.

2.1.3 Evaluating Recommender Systems

To test the performance of a recommender system, like any other prediction and simulation models that uses numerical methods and as is also as applicable to other areas of artificial intelligence, we divide the utility matrix of historic data into two with some percentage so that we have training data and test data (known but withheld data). The test data is normally hidden

from the prediction algorithm. We then use the algorithm to predict these known but withheld values and compare the output of the recommender system with the known but withheld values so as to evaluate whether a recommender system is doing a good job or not.

Evaluating the predictions

The most common way of evaluating whether a recommender system is doing a good job or not involves comparing the predictions against withheld ratings to determine accuracy using root mean square error (RMSE). However in most use cases, accuracy is not usually the only desirable property of a recommender system and in many cases other properties as discussed in the next section also comes into play.

2.1.4 Properties of recommender systems

Properties of recommender systems are the parameters that can be measured in order to evaluate effectiveness of a recommender system. (Shani and Gunawardana, 2011) has discussed the properties of recommender system at length. We will mention a few for the purposes of this discussion.

Prediction Accuracy

Depending on user's circumstances or situation or needs, he/she may want a recommender system to behave in different ways. For example, an online store keeper may want a recommender system which predicts with high accuracy that a certain online shopper will purchase a given item, based on that shopper's profile so as to maximize on the little time that the shopper visits his online store as predicting a wrong item means a wasted opportunity to sell since the web page space is limited and if the user doesn't see what he may want to purchase displayed to him (as an output computed by a recommender system) within a few minutes, he just moves on without purchasing anything. For example, when a book called *Touching the Void* by Chris Anderson was first published, it didn't attract attention of many readers. Years later, the same author published another book in the same topic, mountain climbing, called *Into Thin Air*, which was bought by many through Amazon book store. Amazon's recommender system noticed a few users who bought both book and started recommending *Touching the Void* to any other user who had bought or was considering buying *Into thin Air*. This accuracy of prediction for this recommender system made *Touching the Void* far much more popular than *Into Thin Air* even though it had been there for years. The accuracy of the recommender system made the sales soar very high.

Robustness

A good recommender system should be resilient against attacks on its mathematical properties such as profile injection. The work of (Burke, O'Mahony and J., 2011) describes a robust collaborative recommendation. They categorize the attacks as a promotion attack, where an attacker will want to manipulate the recommender system into outputting an item that is not fairly suitable to the user, and nuke attack, where an attacker will want to manipulate the recommender system in such a way that he prevents it from recommending to the user, an item which if fairly suitable to the user. This may be due to competition or other interests.

Serendipity

At the same time, the seller will want the recommender system to be serendipitous, i.e. be able to output items which the user did not know about, for example, a recommender system may accurately predict that a movie customer who has rated a movie by the name Harry Potter I highly will obvious like or rate Harry Potter II, and Harry Potter III highly, however the recommender system may notice some features in another movie called Star Wars, which the user might not even know exists, that are similar to some features in the Harry Potter I. The recommender system may then try out the user with the other movie called Star Wars instead. If the user ends up liking this Star Wars, then the recommender system is more useful to them as it helps him find new items than predicting obvious ones and boasting of accuracy.

Serendipity will means that the system tries to predict something outside user's norms as inferred from his profile and this means that the there is a high chance of being totally off locus and compromise the accuracy of the recommender system hence a balance has to be achieved and appropriate tradeoff between these two properties be taken into account. This means that the designer needs to be aware beforehand which factor leads to which property and which ones does not during the design process.

User preference

Other factors such as user preference may also come into play as some recommender system designers might be out for research and others out for profit and these will influence their design decisions. Depending on the use case, a designer may want a scalable system or just a small recommender system. This will also be one of the properties that will always inform recommender system design process decisions and it is proper for the designer to know how this

is affected by incorporating more attributes to the a recommender system such as trust and thus the effect of such new additions must be tested and reported beforehand.

Coverage

Another property of recommender system is coverage. Coverage refers to amount of data required by a recommender system on order to start outputting reliable prediction. Since a recommender system relies on historic data to make their prediction, a good recommender system should not require too much data for it to start making accurate predictions. A recommender system that requires too much historic data is likely to run into a cold boot problem, a scenario where it is difficult to start making accurate recommendations or predictions as the recommender system is still learning patterns in the little data it has. This is mostly common in collaborative filtering when a new user or customer who still has no profile tries to purchase something. In many practical use cases new user recommendations are started from some system wide averages and then the user's profile evolve and become more and more individualized to the user (Leskovec, Rajaraman, & Ullman, 2014). This sometimes necessitates a survey on target market.

Confidence

Another property of a good recommender system is that it should build user confidence. It would be advisable to let the user know why a certain recommendation was arrived at by explaining why it arrived at a certain decision. For example a good recommender system should be able to explain to the user that, "I have recommended this article to you since it has content from Syria and in the last three visits, you have read articles that have content about Syria and not for any other ulterior reason". To achieve this, it is necessary to get data about the user and sometimes passive data about the user are actively gathered through means such as a survey in the target market just to get a bank of possible explanations.

2.2 Measuring of Similarity

In collaborative recommendation, items are recommended based on similarity of profiles of users in the space. The following are the techniques of measuring the similarities.

i. Cosine similarity (Lahitani, Permanasari and Setiawan, 2016)
ii. Jaccard Similarity Index (Niwattanakul et al., 2013).
iii. Pearson correlation (also known as centered-cosine)(Sheugh and Alizadeh, 2015).

2.3 Providing the recommendations

We use weighted average to provide recommendation by giving more weight to ratings from users who are more similar to the active user than those who are as similar.

$$P_{i,c} = \bar{S}_i + \frac{\sum_{n=1}^{n} sim(i,j).(S_{j,c} - \bar{S}_j)}{\sum_{n=1}^{n} sim(i,j)} \qquad (4)$$

Where n represents that there are n nearest neighbors

2.4 Prediction Accuracy

To check prediction accuracy, we use Mean Absolute Error (MAE) or Root Mean Square Error (RMSE)(Chai and Draxler, 2014).

2.5 Attacks against the mathematical properties of recommendation engines

(Burke, O'Mahony and J., 2011) Discusses several attacks that can be used to invade a naïve recommender system and also mention how to check the effectiveness of an attack both in terms prediction shifts and hit ratio.

2.6 Trust and recommender systems

Recommender systems are meant to suggest suitable items to users as they shop online. Trust on the other hand is the belief that the other party will carry out his obligation as expected. It is therefore incumbent to protect the recommender systems against invasion by malicious vendors who work for other interests than the interests of active user, and who may want to manipulate the working principles and trick the tool into doing something unexpected.

2.6.1 Augmenting trust into Recommender systems

In 2017, (Yin, Wang and Park, 2017) tried with success to incorporate trust based on sociology into collaborative recommender systems. He discovered that trust improves the prediction accuracy of the recommender system, measured by both Mean Absolute Error and Root Mean Square Error. The key limitations with this work were that the researchers used a dataset which is now not only not being supported but the method of collecting that data required large human efforts as users were expected to make deliberate efforts to explicitly provide their trust opinions against each other in the system and this approach is not only expensive effort wise, but is also subject human bias. The study has shown that if trust is incorporated into a recommender system

then it improves the most desirable property of recommender system – the accuracy as measured by mean absolute error (MAE) and Root Means Square Error (RMSE). This is as demonstrated by the graphs below which depict their results. Therefore the need to figure out how to measure trustworthiness in an autonomous fashion is currently incumbent. The graphs indicate the relationship between prediction accuracy of a recommender algorithm, as measured by Mean Absolute Error (MAE) (shown in Figure 2) and also as measured by Root Mean Square Error (RMSE) (as shown in for Figure 3). Here the study looks at the Collaborative Filtering Recommendation Algorithm (CFRA), discussed in section 2.1.2(ii), Common Filtering Recommendation Algorithm with Trust incorporated, (CFRAT) which is a new concept of incorporating trust to a recommender system and also a Hybrid Recommendation Algorithm, with Trust (HRAT). The HRAT is a recommendation algorithm which uses both Content-based approach, discussed in section 2.1.2 (i) and Collaborative Filtering Recommendation algorithm discussed in section 2.1.2 (ii) combined.

As can be seen from the graphs, it is evident that:

- ✓ When trust is incorporated, the error goes down, measured by both mean absolute error (MAE) and root means Square Error (RMSE) which is an indication of improved recommendation accuracy.
- ✓ The more the neighbors, the more the error goes down, in other words the prediction accuracy improves.
- ✓ Hybrid Recommendation Algorithm incorporated with trust (Trust adjustment factor) or trust enhanced Hybrid Recommender Algorithm (HRAT) outperforms trust enhanced Common Filtering Recommendation Algorithm (CFRAT) which in turn outperforms the classical Collaborative Filtering Recommendation Algorithm (CFRA), in terms of prediction accuracy.

It is important to note that the more the neighbors, other factors remaining the same, the better the prediction. This can be interpreted that more neighbors imply more data to learn a prediction pattern from, and therefore a more accurate prediction. As a matter of fact, choosing the 'proper' number of neighbors is also a topic of scientific research on its own. (Ricci, Rokach and Shapira, 2011).

This study confirmed to us that trust is a necessary parameter, if we were to be keen on prediction accuracy of a collaborative recommendation algorithm. However, as discussed earlier, this study was carried out from a dataset where users explicitly indicated their trust levels against other users in the system in a publicly available opinions website. This does not only require extra efforts from the users as discussed in section 2.1.1 but also is susceptible to cold start problems as discussed in section 2.12 (i).

It is also worth noting that the data set that was used in this study (Yin, Wang and Park, 2017) is no longer available to researchers as the website is no longer there and the data which was downloaded earlier from the website can now be considered as old.

The discussion in this section necessitates the need to come up with new mechanisms of acquiring the trust constructs for the purpose of computation for recommender system algorithms.

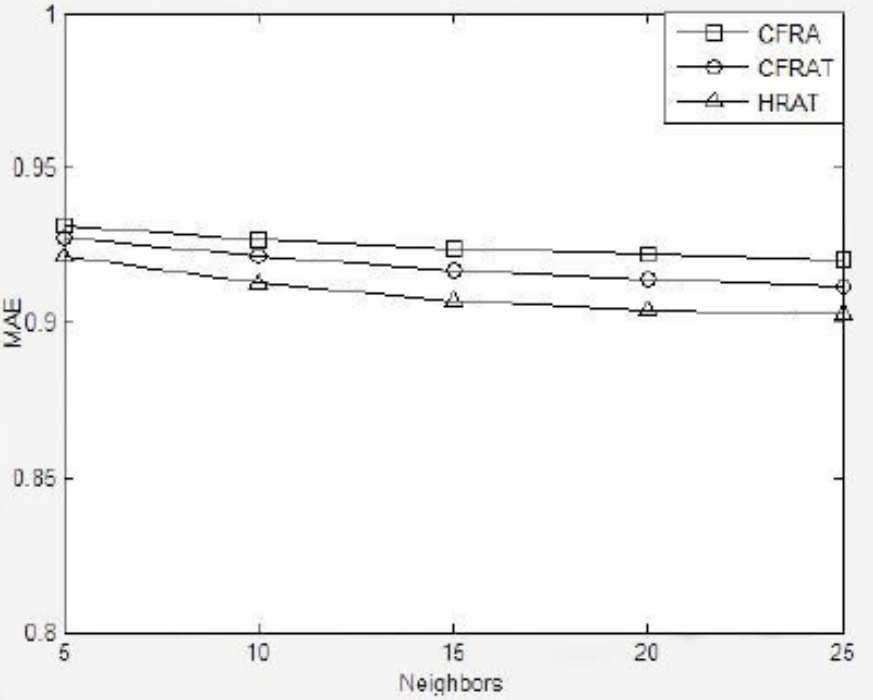


Figure 2 Accuracy results comparison of variants of recommender systems using MAE (source: Yin, Wang and Park, 2017)

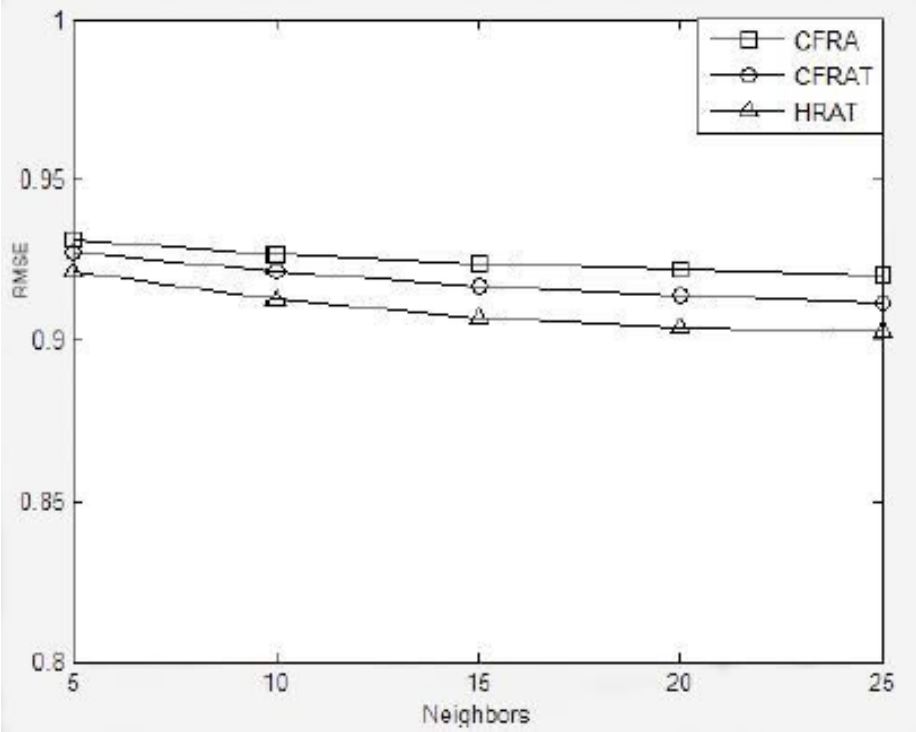


Figure 3 Accuracy results comparison of variants of recommender systems using RMSE (source: Yin, Wang and Park, 2017)

2.7 Trust

2.7.1 Overview of Trust

Trust is the substantiated belief that an agent will comply with the expected standards in a given context and deliver desired results. It is a latent construct that cannot be measured directly but through some indicators which are based on the said context. Many sources in literature associate trust closely with ethics. In online shopping experience, it is the shopper's desire to get the item he is buying for in the promised form, at the right place and at the promised price and promised time. However, some unscrupulous online vendors exploit the obscurity of online systems, in the sense that the buyer does not have the full view of the promised item at the time of purchase and therefore dupe the buyer. Traditionally, there are indicators that buyers use to discern fraud and make informed decision in the normal brick and mortar shop, but at an online shop, the buyer is limited only to the information the seller has provided to him about the item or service, and therefore the buyer is at the mercy of the seller, to some extent.

2.8 Trust Measurement Methods

2.8.1 Scale Development using Factor Analysis

Factor analysis is a regression method that is applied to discover root causes that explain hidden factors that are present in data. It is a method used to explore datasets to find out why data is acting in a certain way or to describe the data. Factors are also known as latent variables or constructs, that is, variables that are quite meaningful but are inferred and not directly observable. For example, imagine you are a marketing data scientist and that you must add to a file actionable customer segments for use in strategic marketing planning. You have got a response from customer survey. You can apply factor analysis to group respondents into meaningful customer segments based on similarities on how responses tend to answer a specific subset of survey questions. So factor analysis is a method that you can use to regress on features in order to discover factors that you can use as variables to represent the original dataset. It is important to note that factor analysis is a two step process which involves:

- ✓ Exploratory Factor Analysis (EFA)
- ✓ Confirmatory Factor Analysis (CFA)

The main difference is that for the Exploratory Factor Analysis, we are keen on reducing the measured variables into few meaningful factors by identifying some structural relationships between the measured variables. Here the factor loadings are calculated, which determine what amount of relationship exists between a variable and a certain factor. We rotate the factors to get a nice distribution so that the variables are not loaded into one factor but they are distributed across several factors. We can have an example of orthogonal rotation where we assume that variables are uncorrelated; we can also have oblique type of rotation where we assume that the variables have some degree of correlation among them. Here, the main task is to explore relationships between exogenous and endogenous variables (indicators vs. factors or latent variables or constructs). Once we have these factors, say F1, F2, F3, F4, we can then use the values derived from each of them for some kind of predictive modeling like in a multiple regression(Suhr, 2009).

For the case of Confirmatory Factor analysis, as the name suggests, the researcher already has some prior knowledge about constructs and the variables in terms of which variable will make up to which factor or a construct and has some theoretical foundation about that so here, the task is

not to explore any more but to confirm that whatever was thought of is actually true just as quality measure.

The two steps above (EFA and CFA) are also referred to as measurement model in Structural Equation Modeling, discussed later in this section. Research work such as (Sergio, 2007) has studied the ethics on online retail in European context using the measurement models described above (CFA and EFA) and has provided some factors that shoppers are worried about which implies trust, alongside the variables or indicators that the factors are inferred from. The factors are:

- ✓ Security
- ✓ Reliability
- ✓ Fear of deception
- ✓ Privacy

These factors can be used for scientific prediction of trustworthiness of an online shop. Since the community norms are different from context to context and from continent to continent, it is very likely to realize that the indicators will be very different in another context or continent, since many of the clues that are used to detect fraudulent activities are usually based on previous experiences and insights that have been passed from one generation to the other over hundreds of years and these vary greatly from community to community. Other contributing factors that vary with context are previous personal experiences, societal norms, laws and regulations, economic factors (McLeod, 2018), exposure to information/general awareness.

The significance of this will be to say appeal to all communities by instilling confidence in them, from their perspectives, by taking care of how they evaluate trust in the online shops or any other context.

2.8.2 Structural Equation Modeling (SEM)

Structural Equation Modeling involves creating equations which depict relationships among constructs involved in some analysis. Structural Equation Modeling comprises of three important parts, namely:

- ✓ Factor analysis
- ✓ Regression Analysis

- ✓ The Chi Square value, which is largely used to test the goodness of fit.

Again, the constructs are unobservable and can only be measured through some items or variables in the questionnaire. A construct can be measured by any number of items in the questionnaire, but a researcher need to be careful not to take very few variables as this will lead into a situation where a model cannot properly explain itself, a shortcoming known as under fitting or low bias, in other words, if to be applied in artificial intelligence, then a machine learning model cannot fit the training data or generalize to new data. Again using so many variables can lead to over fitting or high variance of the model, where a model can produce almost 'accurate' performance in the training data, but does not generalize, such that it cannot produce accurate results in unforeseen data and this is against our objective (International Business Machines Corporation (IBM), 2019).

So in structural equation modeling, we create two models, namely:

- ✓ Measurement Model
- ✓ Structural model

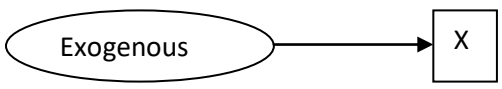
In the measurement model, one measures whether the variables are actually measuring the constructs, or not, using the EFA and CFA discussed above in this section.

In Structural model, we seek to see the structure of the relationships.

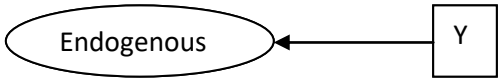
Here we have several types of relationships that we need to asses so as to form an equation that can be used for mathematical prediction of latent variables or constructs, something in the form of $y = mx + c$, where y is the desired construct to be predicted, m is the weight and coefficient of the relationship and c is some constant such as error constant.

The types of the relationships are:

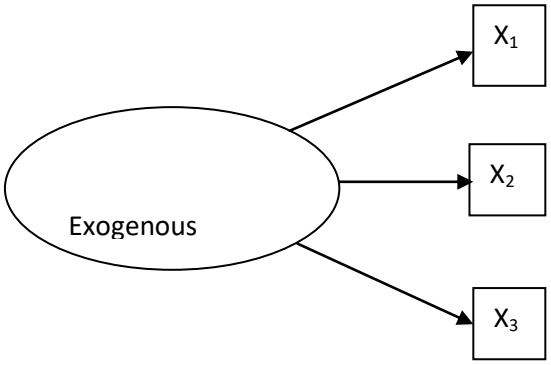
(i) A relationship between a construct and a measured variable, which can be exogenous or endogenous



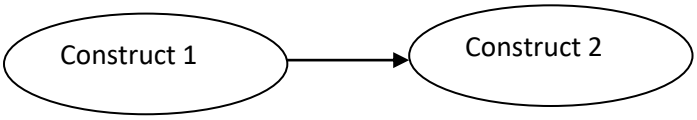
Or



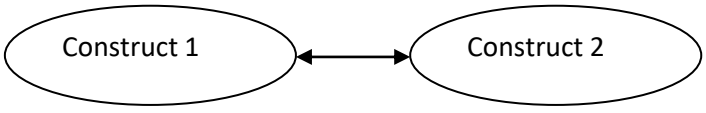
(ii) Relationship between a construct and multiple measured variables.



(iii) Dependence Relationship between Two Constructs (Structural Relationship)



(iv) Correlation Relationship between Constructs



A measurement model can be represented by types i, ii, iv, while a structural model includes all the types of relationships.

A structural model is a one where a researcher is basically testing a set of constructs with an intention to measure the relationships and how it affects and tries to determine the path estimates and from there he can come to some kind of a conclusion or an inference. With this type of modeling, trustworthiness of a service can be inferred or predicted computationally from the indicators. On the other hand, measurement model we only check if the variables are actually measuring the construct. So in measurement model, we say that the variables are actually measuring the construct and in structural model, we say that there is a relationship and we are trying to find that relationship.

Since SEM explains the observed covariance among a set of measured variables, by estimating the observed covariance matrix with estimated covariance matrix. (The estimated covariance matrix is constructed based on the estimated relationships among the variables), it is desirable that the difference is as small as possible to ensure that what is observed vs. what is expected are more or less the same.

We use chi square to test this.

$$\chi^2 = \frac{(\text{Observed Matrix (O)} - \text{Expcted Matrix (E)})^2}{\text{Expected Matrix (E)}} \qquad (9)$$

With the above equation, we can see that chi square becomes large in two cases:

- ✓ When the difference between observed value (O) and expected value (E) is so large.
- ✓ When the expected value E is just so small

In these two cases, we will understand that there is a significant difference between observed and the estimated model, and this is against our wish, since we do not want a very high difference

between the estimated and the observed value so in SEM, a low chi square value is more desired as it means a better fitting model while a high chi square value implies a poorly fit model.

It is important to note that no SEM model should be developed without any underlying theory, since the SEM software, Mostly AMOS graphic, will still give you some results with whichever data but if the results cannot be founded on any theory, then it is not important. The basis of SEM is always some theory unless there is a scientific reason that the theory still needs to be developed, in which case is also good thing but must be done with care. This second scenario is important as it can contribute to the body of knowledge, otherwise any researcher should be discouraged from using SEM without a proper thought process.

To define individual constructs, the following steps need to be taken:

i. Operationalization of the construct
ii. Use of scale from prior research
iii. New scale development
iv. Pre-testing of the construct.

To define the individual constructs, as said earlier, one needs to go with some theory. That is to say, they need to understand what the construct is and why it is required and what variables are or would affect the construct. Only when these questions are answered can one then develop a construct. Then one needs to support it with sufficient literature and research.

Once one has defined the construct, then one can use a new scale, but many at times, even in the cases of confirmatory factors, scales from previous researches are used since such a scale usually has been tested and confirmed or validated somewhere else. So if you are a researcher who is trying to use this construct for some other study (but it has to be theoretically sound) then scales from prior research can be taken and again checked for validation.

As said earlier, a new scale can be developed and validated. This method is always appreciated because it results into adding more to the body of knowledge or is like the researcher is contributing to the knowledgebase.

Once the construct has been defined and a scale is developed, then the construct needs to be pre-tested. We need to test whether the variables are loading to the constructs properly or not. For

example if you had taken, say, five variables then you test and realize that either one or just some, or even all of them are not loading properly onto the construct, then there is no point of going ahead with that scale. Also the variables could be cross-loading and this is equally dangerous so again in this case you will need to drop the scale.

So under SEM, the researcher works in two stages. First he checks the measurement model – if the variables measure a construct as checked through factor loadings and the extent of measurement errors and then he checks the structural model where he focuses on relationships of the construct as given by the structural model.

2.8.3 Other Trust Measurement models

Other sources show that trust measurement can also be modeled using:

- ✓ Using open network environment (Beth, Borcherding and Klein, 1994)
- ✓ Trust Measurement in Health (Jones and Barry, 2011)
- ✓ Using Techniques of Social Trust (Welch, Hinnant and Moon, 2005)
- ✓ Using Techniques of Game theory (Wang et al., 2016)
- ✓ Using Techniques of Psychology (Jiang et al., 2016)

2.9 Distrust

Trust and distrust have been looked at differently by several researchers. Some researchers consider trust and distrust as one term which exists on the opposite sides of a measure while other studies consider trust and distrust as two different measures which can exist independently and simultaneously in the same context. The study (Victor et al., 2009) has discussed the effects of distrust on recommender systems.

2.10 Theoretical Framework

According to (Keith, 1960), the consumer is in the middle, and not the company and so the company revolves around them and not the other way round so it is of uttermost concern that we get to consider how the consumers perceive the presentation of services we offer them including how they assess trustworthiness of the presented services. The study exemplifies the case of

earth and the heavenly bodies in the universe where it was believed that the heavenly bodies revolve around the earth until the Galileo Galilei's discovery that indeed the earth rotates daily on its axis and revolves around the sun, so does the company revolve around the consumers and it is therefore paramount to focus more on the problems of marketing rather than the problems of production, that is to shift focus from the product that we can make and present to the product that the consumer wants us to make, and therefore to focus more on the market place than the company and therefore attaining a marketing oriented economy or marketing companies, from the word go. The study also states that marketing department should develop criteria to determine which products to market and that "these criteria were, and are, neither nothing more nor less than those of the consumer herself". While developing these criteria, then what do we measure to ascertain that the criteria are indeed the desired one? Thomassen (2003, p. 69) defines customer satisfaction as follows: "the perception of the customer as a result of consciously or unconsciously comparing his experiences with his expectations". Indeed the criteria should be such that it maps the customer's experience with a product to his expectations in order to attain satisfaction which improves loyalty and this has a positive impact to the sales volume and consequently the profits, which are at the core of the businesses' existential in the first place (Fraering and S. Minor, 2013). (Rust and Oliver, 1994) defines customer satisfaction as an extent to which a person believes that an experience creates positive feelings. Another study, (Kotler and Keller, 2006) defines satisfaction as a judgment between performance and expectation of a product and (Zeithaml and Bitner, 2003) defines satisfaction as "Satisfaction is the consumer fulfillment response. It is a judgment that a product or service feature, or the product of service itself, provides a pleasurable level of consumption-related fulfillment." Indeed for the said satisfaction to be achieved by deliberately matching the expectation to the final customer's experience, then it is desirable that the consumer's expectations are known beforehand by the business.

In consideration to measure of satisfaction, several studies have looked at contributing factors. In the personal selling arena, (Román and Munuera-Alemán, 2005) define ethical sales behavior as ''fair and honest actions that enable the salesperson to foster long-term relationships with customers based on customer satisfaction and trust''. (Sergio, 2003) reveals that in financial services, a salesperson's ethical behavior leads to higher customer satisfaction, trust and loyalty to the bank that the salesperson represents. Consumers' impressions of a firm's ethical conduct

can be influenced by employees' actions during service delivery (McIntyre, L. and Gilbert, 1999), and so can it be influenced by the presentation of the ecommerce platform. Bricks and mortar stores may be able to signal longevity, and ethical behavior, by factors such as their location and their employees, whereas Internet retailing is ''inherently limited in its ability to offer high-trust persuasive communication '' (Grewala, Iyer and Levya, 2004).

There have also been studies based on conceptual contributions on internet ethics such as (Stead and Gilbert, 2001) and (Tavani, 2000) as well as research on online trust (Bart et al., 2005), (Belanger, Hiller and John, 2002), (Miyazaki and Fernandez, 2001). In 2007, (Roman, 2007) used structural equation modeling to construct and validate a scale for measuring ethics of an online retailer. Ethics imply reliability and therefore closely related to trustworthiness. He ended up with four constructs; however, there is still need to check the results against a different target context. (Zait and Bertea, 2012) presents methods of developing a scale to measure perceived risk in e commerce and specifically presents a method for testing the discriminant validity using data from two surveys.

In 2011, Burke, Michael and Neil (Burke, O'Mahony and Hurley, 2011) carried out a research on Robust Collaborative Recommendation algorithms. They outlined clearly the weakness of unaided collaborative filtering recommendation algorithm. Recommender systems are tools which are meant to alleviate information overload by suggesting to consumers a suitable item to purchase amidst a myriad of alternatives and to a large extent this implies marketing the specific items that are being considered for recommendation. This study highlighted how exposed collaborative filtering recommendation algorithm is to manipulation such as by product nuke or product push which involve inserting fake profiles into the database, an attack known as profile attack.

In 2015, (Yasmin, Tasneem and Fatema, 2015) carried out an empirical study on the effectiveness of digital marketing techniques. They collected empirical data on digital marketing and analyzed using various statistical tools and techniques. The study demonstrated the importance of digital marketing for both marketers and consumers so it is a worthwhile idea save for potential abuse by possible manipulations, for example in the case of Pay per click advertising, which is a way of using search engine advertising to generate clicks to a website rather than earning those clicks organically. This generation of clicks involves automatic biding

for display space in the web page that the web visitor is reading, or ranking in the search engine results page with the background idea that the more conspicuous the advertisement space is on the web page or the higher the ranking in the search engine results page, the more likelihood that the web page visitor will click on a link which directs them to the target online shop and the higher likelihood to make a sale. If left for pricing only as the factor determining the score, then a malicious vendor can outbid benign vendors, but with a malicious intention.

Sponsored search or search advertising enable advertisers to target consumers based on the query they have entered. To some extent, these sponsored searches qualify as a means of recommendation or marketing because once output is returned, the shopper considers the ranking and positioning of items on the webpage as one of the indicators of the superiority of a product The following studies on sponsored searches have focused on maximizing the advertisers profit but with not much regards to the ethics or trustworthiness of the service to be offered(Cornière, 2016), (Athey and Nekipelov, 2010), (Narayanan and Kalyanam, 2015), (Aggarwal et al., 2009), (Ghose and Yang, 2009). This approach therefore still causes the need to look for how to incorporate a trust parameter to remain existential.

It is also possible to estimate trustworthiness of a vendor by providing shoppers with a feedback form to report their experience and assess the satisfaction by comparing the expectation against the actual experience with the vendor such as the method taken by (Jumia KE, 2021). This however is reactionary rather than deterrent measure and will allow for zero day attacks to go through and becomes effective only after adjustments made based on a few reported successful penetrations.

Study in (Zait and Bartea, 2011) discusses how to estimate perceived risk in ecommerce. The paper presents three methods which can be used to assess discriminant validity for multi-item scales. Q-sorting is presented as a method that can be used in early stages of research, being more exploratory, while the chi-square difference test and the average variance extracted analysis are recommended for the confirmatory stages of research. The paper describes briefly the three methods and presents evidence from two surveys that aimed to develop a scale for measuring perceived risk in e-commerce.

From the ongoing discussion, it is safe to say that ethical standards are indeed an estimator consumer satisfaction and a natural estimator of consumer trust. We therefore choose to go with the approach proposed by (Roman, 2007) in this study because this approach touches on ethical behavior of online retailer and this is a natural predictor of trust.

CHAPTER 3: METHODOLOGY

3.1 Introduction

Methodology describes the series of steps that we followed in the research process. It describes the procedures carried out in data collection, the scientific tests that were carried out during the data analysis, as well as the procedures and tools used.

3.2. Area of Study

Our research was about incorporating trust parameter into the classical collaborative recommendation algorithm with the aim of improving its robustness in order to make it more resilient to profile injection attacks by filtering out suspected fraudulent online shops from the recommender system output.

This, in turn, also improves the prediction accuracy of the recommender system and thereby also improves the online experience for the online shoppers since the online users get the items they want more aptly and therefore saving them time and also saves them the burden of choice amidst the myriad of items available in the online space, which run into millions.

3.3 Work Breakdown Structure

In order to achieve the purpose of the research project, we broke down the problem into a work breakdown structure which consisted of work packages. The work packages were in line with the specific objectives of the project, in such a way that the deliverable of each work package corresponded to an achievement of one specific objective in the research project. Having gotten the desired work packages, we then identified the set of activities that when executed within each work package will amount to a deliverable of the work package. We describe the activities that were carried out in the research project, categorized under work packages that correspond to the project objectives below.

In general, some procedures in this methodology sections were extracted and used to create the standard operations procedures (SOPs) tools for the research assistants to rely on after training, and these were augmented with the constant availability of the principal investigator (PI) for consultations, guidance and general supervision.

3.3.1 Key Research Work Packages

The entire research project research involved the following work packages:

i. Item generation
ii. Exploratory Factor Analysis
iii. Confirmatory Factor Analysis (Model Development).
iv. Prototype development and deployment
v. Experimental tests

The rest of this chapter is organized as per the work packages or specific objectives of the research project.

3.4 To determine the indicators of trust in online services.

We needed the trust as a quantity which can be constructed and used as a new parameter in the collaborative recommendation algorithm. Naturally, trust is a latent variable in such a way that it cannot be measured directly but can only be inferred from its indicators which we seek to determine with this objective.

3.4.1 Item generation

To get the questionnaire items for the study, we first adopted some the items from previous studies (Sergio, 2007).

We then reviewed the items in in-depth interviews and focused group discussions in order to:

i. Define the dimensions of the trust construct.
ii. Generate new questionnaire items
iii. Perform a thorough evaluation of the questionnaire item wording
iv. Eliminate any redundant, ambiguous, or poorly worded items.

3.4.2 Focus Group Discussion Questions

The following were the focus group discussion questions

1. Have you purchased an item online at any time in your life?
2. If you have purchased an item online, what was the motivation? If not yet then please explain if you can one day purchase an item online or possibly what is the hindrance?
3. When did you purchase your first and last items online?
4. How was the online shopping experience?
5. Were there any noteworthy concerns?
6. Is there any kind of products (goods or services) that you cannot consider buying online?

7. Would you please identify the exact website where you purchased your last item from and also, if possible, let us know why you chose that website?
   Trust was then defined to the participants as the belief that the online retailer will fulfill his obligation and then a List of dimensions of trust from literature were shown to the participants
8. From the list of dimensions of online ethics, which ones did you find on the website where you last made your online purchase?
9. In the list of dimensions of trust construct shown, in your opinion, what do you think should be added?
10. From the list of dimensions of trust construct, in your opinion, what do you think is not representative of trust construct and should be removed?

3.4.3 Focus Group Composition

Each focus group session consisted of 6-12 members. The members were conveniently sampled from members of a leading university in Kenya's community. These members comprised of faculty members, non-faculty members and students. Some of the members had purchased items online and therefore familiar with the online experience while some of the members had never purchased any item online. It was important to understand the online shopping experience for those who had purchased items online and also understand the reasons as to why others had not. There was both a facilitator as well as a note taker in each focus group discussions. The facilitator also played the role of the moderator.

The composition of each focused group discussion was:

- ✓ Handpicked students only
- ✓ Randomly selected students
- ✓ Selected faculty members only
- ✓ Selected non-faculty members only
- ✓ A mixture of selected students, selected faculty and selected non-faculty members.

The key contributing factor to the selection criteria was willingness to participate and a balance between those who had purchased items online and those who had not.

Other contributing factors were demographic factors such as:

- ✓ Gender
- ✓ Age
- ✓ Income
- ✓ Specialty.

The focus group sessions which involved students were deliberately made to have bigger number for students because it was presumed that when students outnumber faculty members, then they naturally consider the discussion a student level affair than in the opposite case.

3.4.4 Focus Group Process

Each of the focused group discussion was held within the university premises where all members were familiar with and comfortable with. Each Session was planned for two hours and even though we tried to get responses for each of the question from each of the participant, there were deliberate efforts to make the process as informal as possible in order to allow members to participate as freely as possible and contribute towards the discussion maximally.

3.4.5 Focus Groups Termination

The focused group discussions were iterated until the theory saturation was attained, this is when no more new knowledge was being generated after a certain number of the focused group discussions had been conducted.

3.4.6 In-depth Interviews Composition

The in-depth interview participants constituted of members of faculty in leading universities in Kenya. The interviews were brought to an end after attainment of theory saturation, which is when no more new knowledge was coming in after the number of participants had been interviewed.

3.4.7 In-depth Interview Questions

The in-depth interview questions were similar to the focus group discussion questions listed above, only that we sought more expounded answers.

3.4.7 In-depth Interview Process

In the interviews, the process involved first defining trust as the belief that the online retailer will fulfill his obligation.

Then a list of dimensions of trust from literature was shown to the participants.

The participant was then invited to provide his input.

The interviewer ensured that at the end of the in-depth interview the interviewee had responded to all the questions in the focus group questions.

This approach was taken because it was necessary to have the interview process flow as naturally as possible in order to capture maximally from the interviewee.

3.4.8 In-depth Interviews Termination

The in-depth interviews were iterated until the theory saturation was attained, this is when no more new knowledge was being generated after a certain number of in-depth had been conducted.

3.4.9 Item generation outcome

In total, we performed 6 focus group discussions and 6 in-depth interviews.

At the end of focus group discussions and in-depth interviews, 61 items were finally generated from the literature interviews and the focus group discussions and in-depth interviews.

3.4.10 Items thematic review

These items were then submitted to a panel of expert judges (members of faculty from the school of business) in order to assess its content validity.

The panel of experts checked the items for ambiguity, clarity, triviality, sensible construction and redundancy, as well as to making sure that the items reflected the definition of trust.

After the elimination of redundant items or ‘‘not representative’’ items, the experts agreed that the items adequately represented the trust construct.

The remaining items are reported in section 4.2, table 2.

These remaining items were then used to prepare a questionnaire for the Exploratory Factor Analysis, here in called the first study.

3.4.11 The first study (Exploratory Factor Analysis)

Sample and data collection

The unit of analysis in this study was the individual consumers.

These were people who had either ever purchased an item online or have never purchased an item online.

Data collection for item refinement was undertaken with the larger community of a university in Kenya. The survey was conducted by Google forms.

In the Google form, there was an introduction section, where the purpose of the survey was described, and there after inviting the participant to fill in the e-questionnaire.

Online data collection possesses numerous advantages over conventional interviewing methods. The decision to perform an online survey was considered because according to (Best and Krueger, 2002), online surveys offer a more efficient and convenient form of data collection. In addition, an online approach can be more effective for identifying and reaching online users and cut out the respondents to only participants who naturally have a potential to purchase online and not being inhibited by factors beyond our concern such as lack of internet connectivity.

3.4.12 Data Analysis

We used Exploratory Factor Analysis (EFA) and Principal Component Analysis (PCA) tests as the key statistical tests.

Exploratory factor analysis traditionally has been used to explore the possible underlying factor structure of a set of measured variables without imposing any preconceived structure on the outcome while the principal component analysis reduces the number of observed variables to a smaller number of principal components which account for most of the variance of the observed variables,(Suhr, 2009).

The Exploratory Factor Analysis produces maximum likelihood factor analysis while the Principal Component Analysis produces an unrotated principal component analysis.

For Exploratory Factor Analysis, we obtain uniqueness, factor loadings, scree plot, eigen values, parallel analysis, optimal coordinates, acceleration factor.

For Principal Component Analysis, we obtain PCA Importance of Components, Loadings, Scree Plot and biplot.

3.4.13 Statistical Tools

We used R Studio Statistical Program, (The R Foundation, 2021) as our data analysis program. Within this program, the following functions were used:

The **princomp( )** function which produces an unrotated principal component analysis.

The **factanal( )** function which produces maximum likelihood factor analysis.

We report the results of these exercises in section 4.2

3.5 To construct model for measurement and estimation of trust (Scale development) using Factor Analysis.

3.5.1 Introduction

As described in section 2.6.1, factor analysis is a regression method that is applied to discover root causes that explain hidden factors that are present in data. It is a method used to explore datasets to find out why data is acting in a certain way. Factors are also known as latent variables or constructs, that is, variables that are quite meaningful but are inferred and not directly observable.

3.5.2 Sampling Technique

Our target population was adults (people who have attained the age of eighteen years) and are currently living in Kenya.

We used purposive sampling and carried out a sampled nation-wide survey to confirm the findings of the exploratory factor analysis stage results.

We sampled the counties according to old administrative provinces.

We then took into consideration the counties with high income (the metropolitan counties), the counties associated with middle level income as well as the counties that are associated with low income.

This consideration was founded on the fact that trusts in online services, which is a subset of ecommerce, and ecommerce is an economic affair will largely be affected by economic situation of the respondents.

With this understanding, we sampled the counties in such a way that we ended up with counties associated with high income (the metropolitan counties), the counties associated with middle level income as well as the counties that are associated with low income.

The categorization of the counties using economic situation was informed by the report of the Kenya National Bureau of Statistics on Counties (Kenya National Bureau of Statistics (KNBS), 2019). This is a body which was established by the act of parliament in 2006. The body is mandated with collecting, analyzing and disseminating statistics in Kenya, and is also the custodian of the Kenyan statistics.

It has offices both at the headquarters in Nairobi and in all the 47 counties.

Margin of error/Significance level (σ): 0.05 (5%)

Confidence level: 95%

Response distribution: 50%

Suggested sample size for each county: 377 (A target population greater than 20,000)

Table 2 Purposive sampling by county

County	Type	Former Province	Population	Population capped at	Scientifically Acceptable Sample size	Responses
Nairobi	High Income	Nairobi	4,397,073	20,000	377	411
Nyeri	Middle level Income	Central	759,164	20,000	377	410
Homabay	Middle level Income	Nyanza	1,131,950	20,000	377	412
Turkana	Low Income	Rift Valley	926,976	20,000	377	400
Kwale	Low Income	Coast	866,820	20,000	377	390
Kakamega	Middle Income	Western	1,867,579	20,000	377	404

Kitui	Middle income	Eastern	1,136,187	20,000	377	390
Mandera	Middle income	North Eastern	867,457	20,000	377	427
Total						**3,244**

3.5.3 Data Collection Procedure

We created a questionnaire using Google Forms data collection tool.

We then had research assistants physically on the ground reaching out to respondents, introducing themselves and the agenda of the study and then requesting the respondent to either accept the Google form link shared on Whatsap® so that the respondent could fill in the questionnaire on his own electronic device or just to provide questionnaire answers to the research assistant so that the research assistant could fill in the questionnaire using the research assistant's had held electronic device such as a smart phone or a tablet.

3.5.4 Responses

In response, we got a total of 3,244 successful responses. After data cleaning which involved careful removal of incomplete records or records that clearly were not representative, we remained with a total of 2104 valid records. According to (Parasuraman, Zeithaml and Malhotra, 2005), this number of responses is satisfactory for SEM analysis of this nature as the work which suggests that 2000 responses are sufficient. The respondents were adults (people aged 18 years and above) which cut across all demographics.

3.5.5 Data Analysis

In this stage, we used to use Structural Equation Modeling (SEM)(Hox and Bechger, 2014), (Stein, Morris and Nock, 2012).

We used Confirmatory Factor Analysis test as the key statistical test.

Here we obtain several trust models, namely:

i. One factor trust model
ii. Two factor trust model

iii. Three factor trust model

iv. Four Factor trust model

v. Four factor with a second order factor trust model

vi. Factor loadings, fit statistics, and data reliability.

3.5.6 Statistical tools used in data analysis

We used R Studio Statistical Program, (The R Foundation, 2021) as our data analysis program. Within this program, the following utility was of a great help in getting the fit statistics: FitMeasures function.

This function is available in *lavaan* package (Rosseel, 2012).

Other important R functions used are:

SemPaths from the *semPlot* package to get the path diagrams for our models.

Inspect function from *lavaan* package to get the factor loadings.

cronbach.alpha function from *ltm* package to get the cronbach's alpha for measuring data reliability.

We report the results of this exercise in section 4.3

3.6. Augmenting the trust model as a new parameter, called trust adjustment factor, into the classical collaborative recommendation algorithm to create a new trust enhanced algorithm.

For this exercise, we adopt the known steps of the Common Filtering Recommendation Algorithm, which have been tried and tested. This has been described in the work of Yin, Wang and Park. (Yin, Wang and Park, 2017).

To mention briefly, these steps are:

3.6.1 Algorithm Steps for the classical collaborative recommendation

Step 1: Input user i and user-item matrix S[m,n].

Step 2: Calculate the similarities between the target user i and other users in user data according to the user item matrix.

Step 3: Choose the first n similar users of the target user i as the NN(i) according to the similarities.

Step 4: Calculate the predicted scores of target user i to every item according to the formula (4).

Step 5: Arrange the items from big to small according to the value of predicted scores.

Step 6: Choose the former N items as recommendations to the target user i.

Step 7: Output the recommendations.

3.6.2 The algorithm steps to derive the trust parameter

We rely on the trust measurement models constructed in section 3.5 above.

The idea is based on the fact that the trust (perceived risk) construct is not a binary value which can exist simply as true or false but is a complex construct whose value can exist to some extent up to a given degree and therefore must be measured with a scale and then a scientific threshold be introduced, which can now be reduced to a binary value of true or false.

We also recognize that factor loadings are the correlation between observed variables and factors, are standardized regression weights if variables are standardized (weights used to predict variables from factor), and are path coefficients in path analysis,(Suhr, 2009), or put in other words, factor loading is the correlation coefficient between the observed variable and the contributing factor(Statistics Solutions, 2020), and therefore a measure of association and a test of significance or variable coefficients (Bartlett, 1950), (Bartlett, 1951), (Burt, 1952), (Rao, 1955), (Akaike, 1987).

We therefore we consider the factor loadings as a weight of the corresponding indicators since the factor loadings corresponds to the predictive power of the indicator.

We then proceed as follows.

Step 1: Get the loadings between the observed variables and the first-order constructs that the observed variables measure from the trust model.

Step 2: For each-observed variable in the model, get the weight (w_{of}) with which it contributes to the first order construct that it measures by taking quotient from its loading divided by the sum of all loadings of all the observed variables that measure that first order construct.

Step 3: Compute the degree with which each first-order construct is present (p_f) in a site by checking the presence of indicators on site and summing the weights of all present indicators that have been computed in step 2. For indicators which are not present on the site, we consider their value as zero.

Step 4: For each first-order construct on the model, get the weight (w_{fs}) with which it contributes to the second-order construct by taking quotient of its loading onto the second construct divided by the sum of all loadings of the first-order constructs onto the second order construct (trust).

Step 5: Compute the degree with which second order construct (trust) exists in the site by getting the sum of the products of first-order construct's degree of presence (p_f) and their corresponding weights (w_p) with which they contributes to the second-order construct (trust), for all the first-order constructs.

Note: In this step 5, we consider the signage such that if the constructs in the first order is manifested by observable variables which are negatively correlated, as shown by the biplot obtained in section 3.4.3, then we subtract the contribution of that construct from the trust score, instead of adding.

3.6.3 The mathematical expression for the deriving the trust parameter

We start by assessing the degree with which first order indicators are present and thereby construe the extent with which first level constructs of trust are present in a given ecommerce platform.

Each construct of our model is considered a term in the equation of computing the final trust value.

$$Construct_i = \frac{\sum_{i=1}^{n} C_i Y_i}{\sum_{i=1}^{n} C_i} \qquad 10$$

Where:

C = the variable coefficient for the indicators of the constructs. The coefficients are obtained from the model.

Y = the boolean (0 or 1) value which indicates as to whether the indicator is sufficiently present in the e-commerce platform or not

We get the terms recursively for all constructs. The direction with which construct or term for this case affects trust is gotten from the biplot. It can be positive or negative.

After computing the value for each term, in the first order construct, we aggregate the values of all construct into one trust parameter using the equation below.

$$\text{Trust} = \frac{\sum_{i=1}^{n} C_i \mathrm{T}_i}{\sum_{i=1}^{n} C_i} \qquad 11$$

Where:

C = the variable coefficient for the first order constructs. These coefficients are obtained from the model. It indicates the weight or the relative importance of this particular first order construct in arriving at the second order construct, which is the trust value which we are looking for.

T = the value of the first order construct as obtained in equation 10.

3.6.4 Arriving at the trust threshold

In order to arrive at the threshold value, below which we consider a vendor not sufficiently trustworthy to be considered in the recommendation system, we use Cochran (Israel, 1992) sampling formula. This is because we expect a very large population of vendors in the long run.

Again we have a background idea that we need a number of vendors which fairly represents the total number vendors in our system. So once we get the sample number, n, that represents our vendor population, then we rank the vendors according to the computed trust scores in equation 11 and take the first n.

$$\eta_0 = \frac{Z^2 pq}{e^2}$$

Where:

η_o = sample size

Z_2 = the abscissa of the normal curve that cuts off an area α at the tails (1 - α equals the desired confidence level, e.g., 95%).

e = the desired level of precision

p = the estimated proportion of an attribute that is present in the population

q = 1-p.

The value for Z is found in statistical tables which contain the area under the normal curve.

3.6.5 The Algorithm steps of Trust Enhanced Collaborative Filtering Recommendation Algorithm (CFRAT)

Step 1: Compute the trust score of each vendor and hence item using the trust model obtained in objective 2 – as described in section 3.62.

Step 2: Compute the minimum trust score accepted for all items (take the least of top n trusted vendors) as described in section 3.63

Step 3: Input user i and user-item-trust-minimum_trust_score matrix S[m,n,t,mt].

Step 4: Filter out items not meeting threshold for trust

Step 5: Input user i and user-item matrix S[m,n].

Step 6: Calculate the similarities between the target user i and other users in user data according to the user item matrix.

Step 7: Choose the first k similar users of the target user i as the NN(i) according to the configurations, default k = 40 in the library.

Step 8: Calculate the predicted scores of target user i to every item according to the formula (4).

Step 9: Arrange the items from big to small according to the value of predicted scores.

Step 10: Choose the former N items as recommendations to the target user i.

Step 11: Output the recommendations.

3.7 To deploy the new trust enhanced algorithm into an empirical setup for production purposes.

3.7.1 Introduction

With the new algorithm in place, as described in section 3.6, it was time to assess its effectiveness in an empirical set up. We created an online shop at www.filteredkenya.co.ke as an online aggregation shop where many affiliates are selling through. The affiliates represent the represent natural online the vendors. The store is created using WordPress™ CMS (WordPress.com, 2016) and is driven by Electro template (Themeforest, 2015) which is anchored on Woocommerce (Woocomerce, 2016). Woocommerce is a customizable, open-source e-commerce platform built on WordPress. We then used web scrapping technique and augment with manually going through the affiliate affiliate's website in order to determine the presence of indicators of trust obtained in section 3.4 and reported in section 4.2, table 5.

3.7.2 Infrastructural set up

We use a shared cloud hosting server (Cpanel, LLC, 2020), and also a cloud server of 32GB RAM, 8vCPU, 580GB Disk running Centos Operating system of version 8 with PHP 7 and MySQL Version 8 database server installed.

3.7.3 Database Design

We create four key tables namely:

- ✓ Users table
- ✓ Orders table
- ✓ Items table
- ✓ Recommendations table
- ✓ Affiliates table
- ✓ Trust_indicators table

The key fields on the tables are:

Customers table

- ✓ User_id
- ✓ First_name

- ✓ Last_name
- ✓ Date_of_birth
- ✓ Group_id
- ✓ Gender
- ✓ Residence
- ✓ Address
- ✓ Mobile_number
- ✓ E-mail_address

Orders table

- ✓ Order_id
- ✓ Item_id
- ✓ Customer_id
- ✓ Order_status
- ✓ Created_at
- ✓ Updated_at
- ✓ Competed_at

Items table

- ✓ Item_id
- ✓ Name
- ✓ SKU
- ✓ Description
- ✓ On_offer
- ✓ Price
- ✓ Image URI
- ✓ Thumbnails URI
- ✓ Quantity

Recommendations table

- ✓ Recommendation id
- ✓ Customer_id
- ✓ Item_id
- ✓ Is_active
- ✓ Created_at
- ✓ Updated_at

Affiliates table

- ✓ Affiliate_id
- ✓ Affiliate_name
- ✓ Affiliate_cartegory
- ✓ Affiliate_address

Trust Indicators table

The fields for this table corresponds to the indicators described in tables I to table 3.

3.7.4 Coming up with sales items and reaching out to the online buyers

To get the items for the prototype shop, we benchmarked with the offerings of the existing online shops, and then with time studied the buying trends and also considered the shoppers feedbacks. We then marketed this ecommerce shop through posters at prime places and also digital marketing techniques as described in (Yasmin, Tasneem and Fatema, 2015).

3.7.5 Gathering Ratings

We prompt users to rate items through the dashboard notifications, SMS and e-mail.

We also accept ratings from unverified purchases and consider qualitative reviews provided along the ratings in order to capture the sentiments of the users who are not buying particular items.

At this stage, the user is encouraged to rate the end to end experience with the process, from the time they discovers the item up to the actual experience with the product, in a case where a user buys the item – in some cases the user doesn't, and assess satisfaction level in a rating scale of 1-5 as a measure to whether he is satisfied, that is whether the end to end process has met his expectation, has exceeded the expectation or has not met the expectation.

3.7.6 Generating recommendations

We run the recommendation engine every midnight using Linux cron job.

The recommendation engine computes the recommendations using *python's surprise* library and stores the output in form of user and the corresponding recommended items on a database table.

These recommendations are then communicated to shop customers if they are equal or greater than the threshold (set at 4.0) through one of the following channels:

- ✓ SMS marketing message (considering the Communication Authority of Kenya's non disturb hours)
- ✓ Email marketing message to the customer's email address.
- ✓ A recommendation on the shop's landing page dashboard when a customer logs into the e-commerce shop.

3.7.7 Online Evaluation of Success of a recommendation and Evaluation of the Prediction Accuracy

We use both explicit and implicit way of gathering user ratings on our products. For explicit rating, the users are encouraged to fill an item review form and provide their 1 to 5 rating.

For implicit rating, the user had to perform one or more of the following actions upon the reception of a recommendation through any of the channels above.

- ✓ Purchase the product
- ✓ Add the product to wish list
- ✓ Adds the product to products comparisons list

For these actions, we also consider a rating of 5.

3.7.8 Offline Evaluation

We record all the data for the purposes of further offline evaluations described in the next section (section 3.8).

3.8 Testing the Impact of the Quantified Trust as a Trust Adjustment Factor on the Performance of Recommendation Algorithms for prediction accuracy and robustness.

For this objective, we performed comparative analysis tests between our new trust enhanced collaborative recommendation algorithms vs. the classical collaborative recommendation algorithm.

We focused on robustness of the recommender system against a profile injection attack and the prediction accuracy of the system.

3.8.1 Measuring the prediction accuracy

To test prediction accuracy, we use MAE and RMSE as by (Shani and Gunawardana, 2011) and reported our results in section 4.5.

To measure prediction accuracy, we used the following tools to automate the process.

i. *train_test_split()* function from *model_selection* module in the python *surprise* library
ii. *accuracy* module from the python *surprise* library

We use these tools for different number of neighbors.

The number of neighbors is passed as a parameter during algorithm object instantiation.

3.8.2 Measuring Robustness

To test the robustness of our new trust enhanced algorithm, which is the key for our problem statement, we adopt the procedures and metrics for evaluation of robustness recommender algorithm. These procedures and metrics have been described in (Burke, O'Mahony and Hurley, 2011).

We attack every item individually by inserting fake user profiles which nuke or promote the item as suitable for the stage of the research.

We use the following steps to mount the different attack models

3.8.3 Random Attack (*Basic Attack)*

i. Assign random ratings distributed around the overall mean assigned to the filler items
ii. Assign a pre-specified rating assigned to the target item, r_{max} for push, r_{min} for nuke

3.8.4 Average Attack (*Basic Attack)*

i. For each filler item, assign a rating that corresponds to (either exactly or approximately) to the mean rating for that item, across the users in the database who have rated it.
ii. Assign a pre-specified rating assigned to the target item, r_{max} for push, r_{min} for nuke

3.8.5 Bandwagon Attack (*Low-knowledge attacks)*

i. Associate the attacked item with a small number of frequently rated items
ii. Assign a pre-specified rating assigned to the target item, r_{max} for push , effective for user-based, not item-based algorithm

3.8.6 Segment Attack (*Low-knowledge attacks)*

i. Find a targeted group of users with known or easily predicted preferences
ii. Assign a pre-specified rating assigned to the target item, r_{max} for push, r_{min} for nuke

3.8.7 Love/Hate Attack - Nuke Attack

i. Assign r_{min} to the target item.
ii. Assign r_{max} to all other filler items.

3.8.8 Reverse Bandwagon Attack - Nuke Attack

i. Identify items that tend to be rated poorly by many users
ii. Assign these items low ratings together with the target item

3.8.9 Popular Attack (Informed)

i. Get the average rating for the target item

ii. Rates the filler items either r_{min} +1 and r_{min}, according to whether the average rating for the item is higher or lower

iii. For negative prediction shifts, assign the target item a rating of r_{min}, and ratings of r_{max} and r_{max-1} to the filler items.

3.8.10 Probe Attack Strategy

i. Create a seed profile

ii. Use the seed profile to generate recommendations from the system (will be well-correlated with real users' opinions)

iii. Learn the system with these recommendations

iv. Use the knowledge to perform an attack – To mount a segment attack, probe narrowly and to mount an average, prove widely

3.9 Hypothesis Testing

3.9.1 Introduction

In this work, we have two properties of the collaborative recommendation system algorithm to test the effect of trust against.

These are the algorithm robustness and the prediction accuracy of the trust enhanced algorithm.

3.9.2 Hypothesis Testing Step

According to (Shafer and Zhang, 2012), the following are the steps to hypothesis testing

1. Identify the null and alternative hypotheses.

2. Identify the relevant test statistic and its distribution.

3. Compute from the data the value of the test statistic.

4. Construct the rejection region.

5. Compare the value computed in Step 3 to the rejection region constructed in Step 4 and make a decision. Formulate the decision in the context of the problem, if applicable.

Since the two properties which we are interested in are measured with different metrics, we had to formulate two hypotheses, one for each of the properties.

Again within each of the properties, we have more than one metric which indicates the property.

As a result we also formulate sub hypotheses which measure different metrics and further on base our judgments on the outcome of all of them to make a conclusion.

3.9.3 Robustness

H_o: Autonomous trust model has no significant effect on the robustness of a collaborative recommendation algorithm.

H_1: The autonomous trust model has positive significant effect on the robustness of a collaborative recommendation algorithm.

The robustness of a collaborative recommender system is measured using two metrics, namely:

i. Prediction shift
ii. Hit Ratio

The prediction shift is the difference between the rating before and after attack. The lower the prediction shift after an attack, the more robust the algorithm is against that attack. See section

So the hypothesis can be expressed mathematically as:

$$H_o: \mu_{predo} = \mu_{pred} \text{ or } H_o: \mu_{predo} - \mu_{pred} = 0$$

Where:

μ_{predo} is the prediction shift before embedding trust

μ_{pred} is the prediction shift after embedding trust

The Hit Ratio is the average likelihood that a top N recommender will recommend the pushed item. The lower the hit ratio after an attack, the more robust the algorithm is against that attack.

So the hypothesis for the hit ratio can be mathematically expressed as:

$$H_o: \mu_{hit\_ro} = \mu_{hit\_r} \text{ or } H_o: \mu_{hit\_ro} - \mu_{hit\_r} = 0$$

Where:

μ_{hit_ro} is the hit ratio before embedding trust

μ_{hit_r} is the hit ratio after embedding trust.

Since there are several forms of attacks that can be carried out on the algorithm, it is prudent to break down the hypotheses into sub hypotheses are used to make a conclusion.

Table 3 Sub hypotheses for measuring the robustness of the algorithm after embedding trust

S/N	Description	Sub Hypothesis	P - value (σ)	T-stat	t-critical one tail	Number of observations (n)	Remark	Reject Null?
1	Prediction Shift for product push attack on user based collaborative filtering algorithm	Average						
		Bandwagon						
		Random						
2	Hit Ratio for product push attack on user-based collaborative filtering algorithm	Average						
		Bandwagon						
		Random						
		Baseline						
3	Prediction Shift for product push attack on item-based collaborative filtering algorithm.	All Users						
		In Segment						
4	Hit Ratio for product push attack on item-based collaborative filtering algorithm	All User						
		In Segment						
		Baseline						
5	Prediction shifts achieved by nuke attacks against the user-based algorithm	Average						
		Bandwagon						
		Random						

		Love/Hate						
		Reverse Band Wagon						
6	Prediction shifts achieved by nuke attacks against the item-based algorithm	Average						
		Bandwagon						
		Random						
		Love/Hate						
		Reverse Band Wagon						
7	Hit ratios achieved by the popular, probe and average push attacks against the user-based algorithm.	Popular						
		Probe						
		Average						

3.9.4 Prediction Accuracy

H_o: Autonomous trust model has no significant effect on the prediction accuracy of a collaborative recommendation algorithm.

H_1: The autonomous trust model has positive significant effect on prediction accuracy of a collaborative recommendation algorithm.

Since we use the mean absolute error and root mean square error as a measure of accuracy, it comes out naturally that the lower the error, then the higher the prediction accuracy.

These hypotheses can be mathematically expressed as:

$$H_o: \ \mu_o = \mu \text{ or } H_o: \ \mu_o - \mu = 0$$

So the sub hypotheses for this case become

For measuring prediction accuracy through mean absolute error:

$$H_{MAEo}: \ \mu_{MAEo} = \mu_{MAE} \text{ or } \mu_{MAEo} - \mu_{MAE} = 0$$

Where:

μ_{MAEo} is the Mean Absolute Error before embedding trust

μ_{MAE} is the Mean Absolute Error after embedding trust

For measuring prediction Accuracy through Root Mean Square Error:

$$H_{RMSEo}: \ \mu_{RMSEo} = \mu_{RMSE} \text{ or } \mu_{RMSEo} - \mu_{RMSE} = 0$$

Where:

μ_{RMSEo} is the Root Mean Square Error before embedding trust

μ_{RMSE} is the Root Mean Square Error after embedding trust

Table 4 Sub hypotheses for measuring the prediction accuracy of the algorithm after embedding trust

S/N	Description	Sub hypothesis	P - value (σ)	T - stat	t-critical one tail	Number of observations (n)	Remark	Reject Null?
1	Measuring prediction accuracy through Mean Absolute Error	Mean Absolute Error						
2	Measuring prediction accuracy through Root Mean Square Error	Root Mean Square Error						

3.9.5 Hypothesis Statistic

In order to identify the relevant test, we consider the concept of central limit theorem (Kwak and Kim, 2017). According to the central limit theorem, the means of a random sample of size, *n*, from a population with mean, μ, and variance, σ^2, distribute normally with mean, μ, and variance, σ2n. Using the central limit theorem, a variety of parametric tests has been developed under assumptions about the parameters that determine the population probability distribution. For this, we choose t-test. The t-test is the small sample analog of the z test which is suitable for large samples. A small sample is generally regarded as one of size n<30. A t-test is necessary for small samples because their distributions are not normal.

Since all of our observations are less than 30, we use t-statistic.

3.9.5 Confidence Level

We choose a confidence level of 95% since this is the most widely used and should be changed only when there is a strong reason to, which we did not have, so, σ or significant level is 0.05.

3.9.6 Hypothesis Testing Tool

We used the one tailed paired two sample for means t-test from the Data Analysis Tool pack plug-in in Microsoft Excel to compute the p-values, σ.

The results of this procedure can be interpreted by looking at (Dorfman, 2019).

3.9.7 Decision Making

In order to make a decision from the p-value, we compare it with the significance level α.

The general rule is that when the p value is less than the significance level, we reject the null hypothesis.

CHAPTER 4: RESULTS

4.1 Introduction

Having satisfactory carried out the research project with the steps described in chapter 3, we here report our research project results.

The results have been categorized according to each of the project objectives as below.

4.2 Determining the indicators of trust in online services

4.2.1 Accepted questionnaire items

After a thematic review of items gotten from the output of the focus group discussions, in-depth interviews and literature, the following questionnaire items were accepted.

Table 5 Accepted Questionnaire Items

Constructs of Trust	Items/Indicators of the Constructs			
	Items to measure		How to measure	When to measure
	Item's Variable name S/N	Item Description		
Security (L1)				
	S1	The security policy is easy to understand	True of false	Before purchase
	S2	The site displays the terms and conditions of the online transaction before the purchase has taken place	True of false	Before purchase
	S3	The site provides information about the company behind the site	True of false	Before purchase
	S4	The site appears	True of false	Before

		to offer secure payment methods		purchase
	S5	You can confirm the details of the transaction before paying	True of false	Before purchase
	S6	This site has adequate security features	True of false	Before and after purchase
Privacy (L2)				
	P1	The site clearly explains how user information is used	True of false	Before purchase
	P2	Only the personal information necessary for the transaction to be completed needs to be provided	True of false	Before and after purchase
	P3	Information regarding the privacy policy is clearly presented	True of false	Before purchase
Deception (L3)				
	D1	The site exaggerates the benefits and characteristics of its offerings	True of false	Before and after purchase
	D2	It is not entirely truthful about its offerings	True of false	Before and after purchase
	D3	The site uses misleading tactics to convince consumers to	True of false	Before and after purchase

		buy its products		
	D4	This site takes advantage of less experienced consumers to make them purchase	True of false	Before and after purchase
	D5	This site attempts to persuade you to buy things that you do not need	True of false	Before and after purchase
	D6	The site items are abnormally priced, as compared to other sites	True of false	Before and after purchase
Reliability/Fulfillment (L4)				
	R1	The price shown on the site is the actual amount billed	True of false	Before and after purchase
	R2	You get what you ordered from this site	True of false	Before and after purchase
	R3	The products I looked at were available	True of false	Before purchase
	R4	Promises to do something by a certain time, they do it	True of false	Before and after purchase
Demographic Information (L5)				
	DE1	Age	Years between birth and the last birthday	At the point of filling the questionnaire
	DE2	Gender	Male or female	At the point of filling the questionnaire

	DE3	County of Residence	Current county of residence	At the point of filling the questionnaire
	DE3	Marital Status	Options: ✓ Single (never married) ✓ Married or domestic partnership ✓ Widowed ✓ Divorced ✓ Separated	During the time of shopping
	DE4	Level of education	Highest schooling level completed: ✓ No schooling completed ✓ Class 8 (KCPE) ✓ Form 4 (KCSE) ✓ Trade/technical/vocational training (Certificate) ✓ Diploma ✓ Bachelor' Degree ✓ Master's Degree ✓ Doctorate Degree	During the time of shopping
	DE5	Employment Status	Options: ✓ Employed for wages ✓ Self-employed ✓ Out of work and looking for work ✓ Out of work but not currently looking for work ✓ A student ✓ Retired	During the time of shopping
	DE6	Employment Type	Classes: ✓ Information Technology related ✓ None Information Technology related	During the time of shopping
	DE7	Income	Monthly income in KES	During the time of shopping

Shopping Characteristics (L6)				
	SC1	Frequency	Average number of items purchased per month in the last three months	A period of the past three months as of the time of data collection
	SC2	Time spent online	Average number of hours spent online per week in the last three months	A period of the past three months as of the time of data collection
	SC3	Type of services purchased online	Classes to select multiple from: ✓ Digital content e.g. software and music or videos ✓ Electronic items such as phones and laptops ✓ Fashion items such as clothes, shoes and watches ✓ Home items such as toiletry and utensils ✓ Health and beauty products such as medicine and beauty creams ✓ Food Items such as snacks ✓ Baby Products such as diapers, wipes, feeding bottles ✓ Office Products such as pens and other stationery or office furniture	A period of the past three months as of the time of data collection
	SC4X	Reasons to buy online	Reasons to select multiple from: ✓ Reliable because I can track my purchase ✓ Saves time	A period of the past three months as of the time of data collection

			✓ Is cheaper compared to brick and mortar shop ✓ I purchase because of disruptive online advertisements such as when doing my work online and an advert for online service pops up	
	SC5	Money spent per purchase	Average amount in Kenya shillings that is spent at every checkout (might contain several items at ago) Classes: ✓ Less than KES 500 ✓ KES 500 to KES 5000 ✓ KES 5001 to KES 10000 ✓ Over KES 10000	A period of the past three months as of the time of data collection
	SC6	Money spent per item	Average amount in Kenya shillings that is spent on every item purchased online. Classes: ✓ Less than KES 500 ✓ KES 500 to KES 5000 ✓ KES 5001 to KES 10000 ✓ Over KES 10000	A period of the past three months as of the time of data collection
	SC7	Money spent per month	Average total amount spent every month on online purchases. Classes: ✓ Less than KES 500 ✓ KES 500 to KES 5000 ✓ KES 5001 to KES 10000 ✓ Over KES 10000	A period of the past three months as of the time of data collection
	SC8X	Most preferred online shop	Options to select one: ✓ Jumia	A period of the past

			✓ Alibaba ✓ Amazon ✓ Kilimal ✓ Shopit ✓ Other	three months as of the time of data collection
	SC9	Most preferred payment method	Options to select one: ✓ Mobile money such as MPESA, Airtel Money, Telkom T-Kash ✓ Credit card such as VISA and Mastercard ✓ Cash on delivery so you pay cash when goods are delivered to your door step or at the pickup station. ✓ Other	A period of the past three months as of the time of data collection
	SC10X	Most liked feature in the most preferred online shop	Options to select one: ✓ Security ✓ Privacy ✓ Reliability ✓ Non Deception ✓ Other	A period of the past three months as of the time of data collection
	SC11X	Most disliked factor that can deter online shopping	Options to select one: ✓ Online insecurity ✓ Lack of privacy ✓ Unreliability of online services ✓ Fear of deception ✓ Other	A period of the past three months as of the time of data collection

Note: D6 cross loaded with D3 so dropped after factor analysis

4.2.2 First Study Questionnaire Responses

The questionnaire was then created and sent out to responses and below is a screenshot of responses as shown in the Google forms responses interface

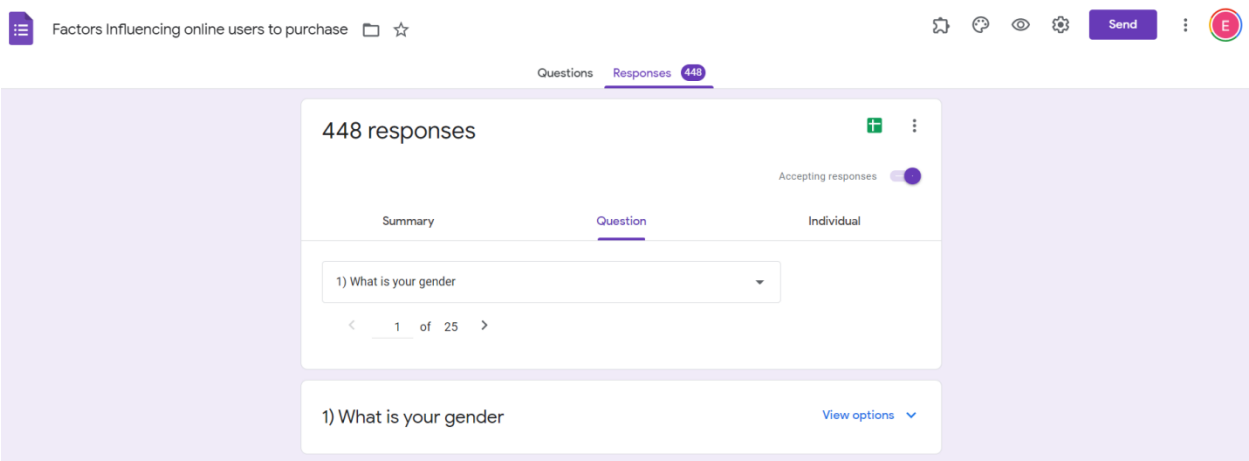


Figure 4 First Study Google For Questionnaire Responses

4.2.3 First Study Demography

Results at this stage show that of the respondents so far, they are aged between 18 -55 years, 56% male and 44 percent female. The level of education of the respondents ranges between diploma students to PhD holders.

4.2.4 First Study mount spent per month in online purchases

We realize that 50% of them buy at least 2 items online per month, 25% buy 3-5 items online every month, 6% buy more than 10 items online while 19% do not shop online.

We also realize that majority (38.6%) spend only between KES 1 and KES 10000.

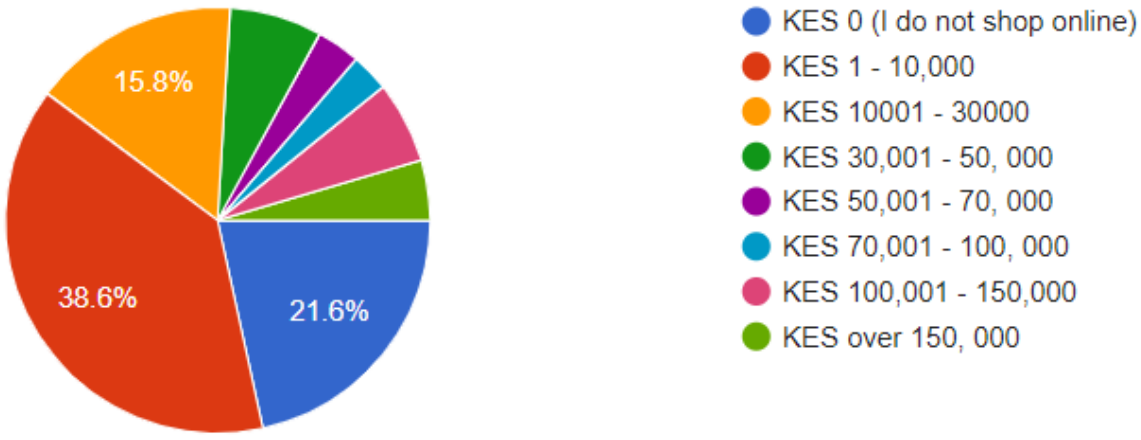


Figure 5 Average amount spent in online purchases in Kenyan Shillings per month

4.2.5 Awareness of benefits of shopping online

We also realized that the participants are cognizant and appreciate the benefits of online shopping.

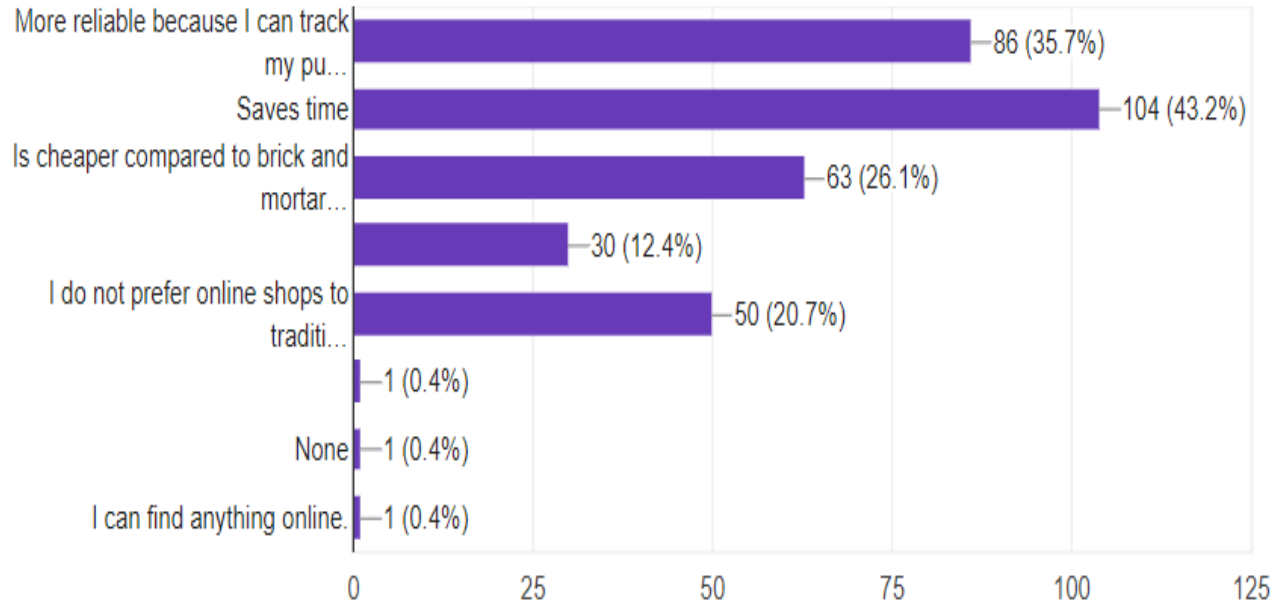


Figure 6 Factors that encourage online shopping

From the results, 43.2% agree that online shopping saves time, 35.7% feel that if done right can be more reliable since they can track their purchases as it gets delivered while 26.1% agree that it is cheaper than brick and mortar shops, 12.4% say they buy impulsively because of disruptive online marketing which they would not have done be it not for online marketing and again 20% say they do not shop online at all.

4.2.6 Factors that hinder online shopping

Then we realize that of the key factors that prevent them from shopping online, the leading one is fear of deception (40.7%) followed by unreliability of online services (22.8 %) then online security site (19.5%) and privacy (12.9%)

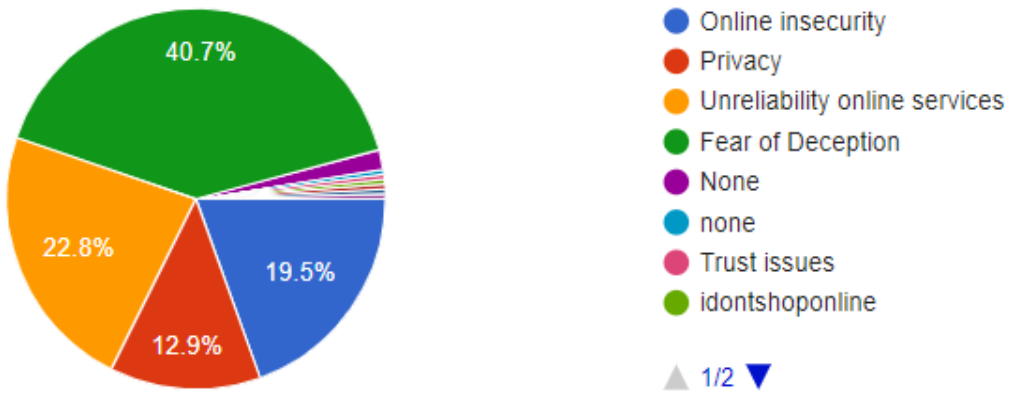


Figure 7 Factors that hinder online shopping

4.2.7 Indicators of Trust

(Sergio, 2007) Describes privacy, reliability, security and deception as latent variables or constructs. This is to say they are variables which cannot be observed directly but can only be estimated through other observable variables. These observable variables can be used as the indicators to estimate the latent variables.

We here present the responses to the indicators of privacy, reliability, security and deception.

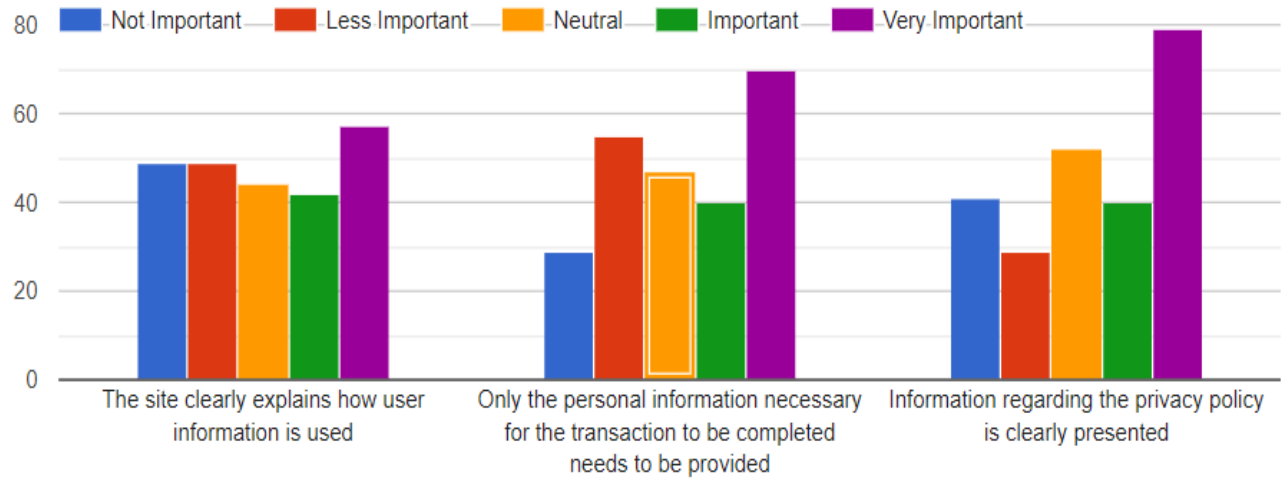


Figure 8 Responses to indicators of privacy

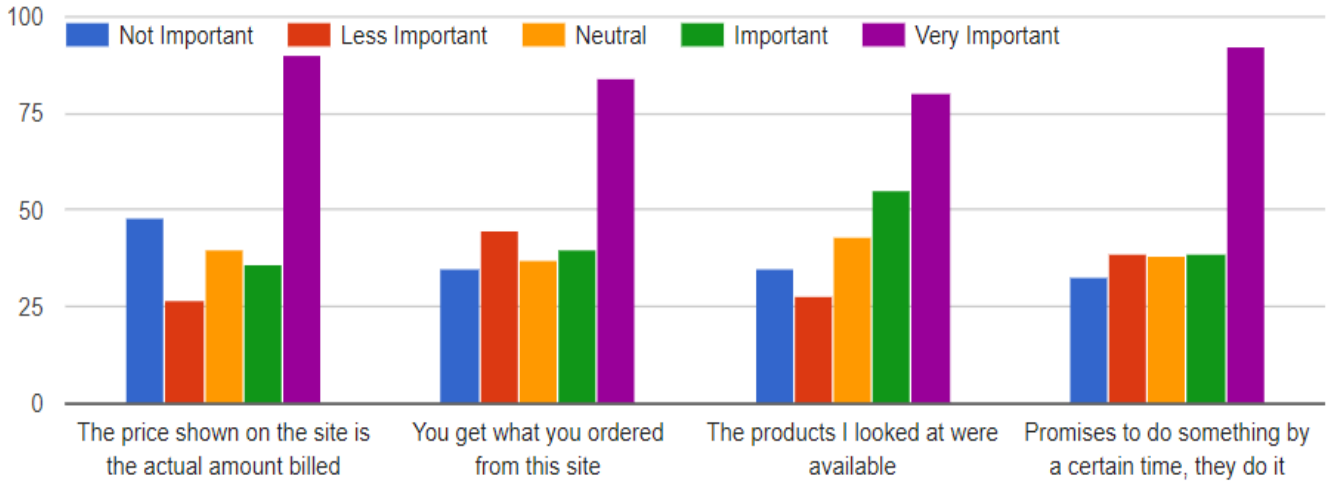


Figure 9 Responses to indicators of reliability.

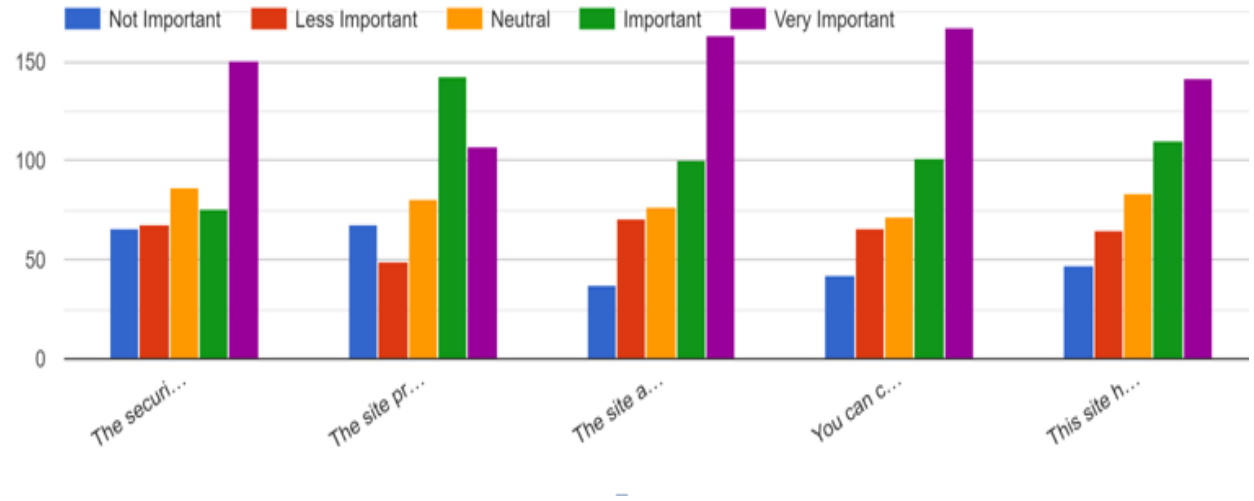


Figure 10 Responses to indicators of online security

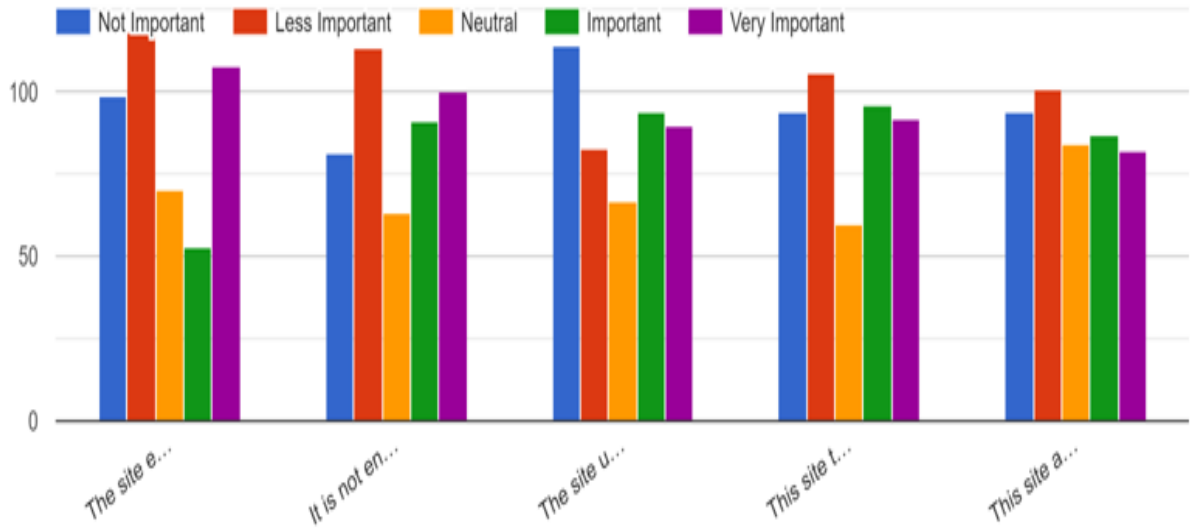


Figure 11 Responses to the indicators of non-deception

4.2.8 Exploratory Factor Analysis – Eigenvalues (Scree Test), with Abnormal Pricing

```
# Determine Number of Factors to Extract
library(nFactors)
ev <- eigen(cor(mydata)) # get eigenvalues
ap <- parallel(subject=nrow(mydata),var=ncol(mydata),
  rep=100,cent=.05)
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)
plotnScree(nS)
```

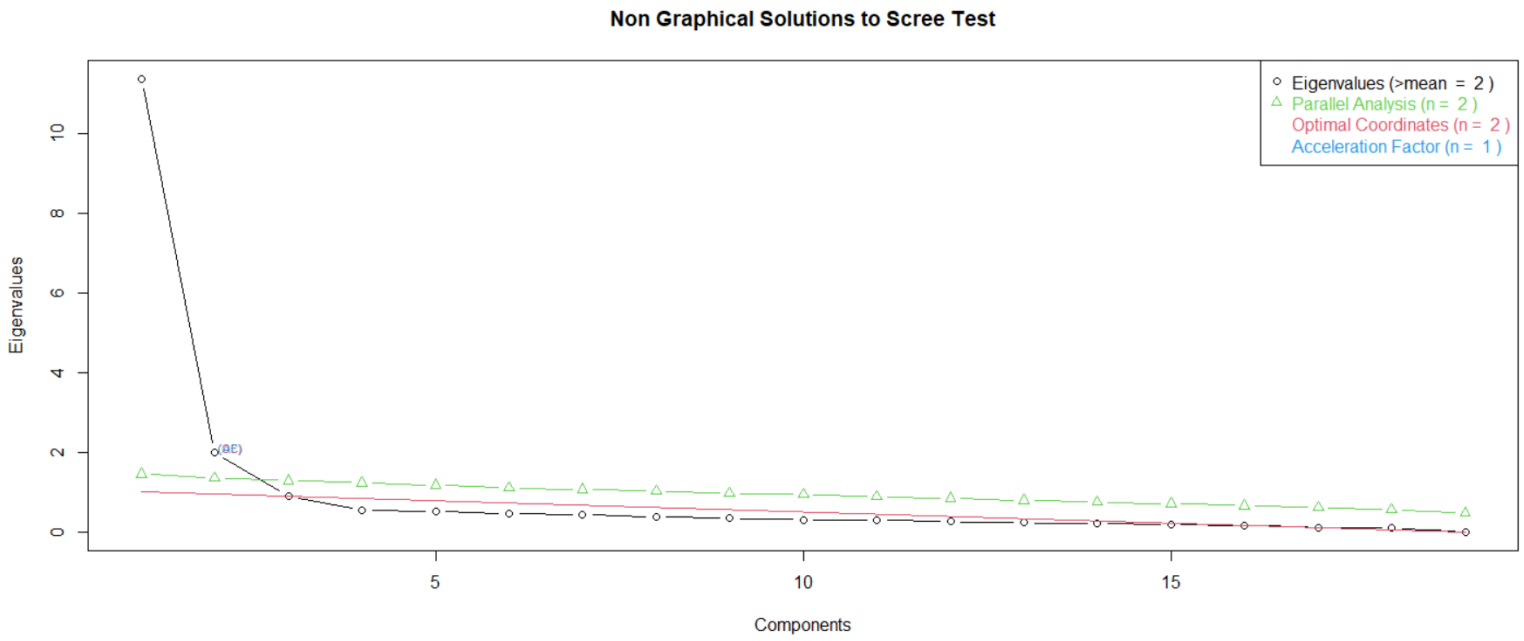


Figure 12 Exploratory factor analysis - eigenvalues (scree test), with abnormal pricing

4.2.9 Exploratory Factor Analysis – Uniqueness, with Abnormal Pricing

What is not measured (i.e., the uniqueness or error term variance), (Gorsuch, 1983).

```
> # Maximum Likelihood Factor Analysis
> # entering raw data and extracting 4 factors
> # with varimax rotation
> fit <- factanal(mydata, 4, rotation="varimax")
> print(fit, digits=2, cutoff=.5, sort=TRUE)
```

```
> print(fit, digits=2, cutoff=.5, sort=TRUE)

Call:
factanal(x = mydata, factors = 4, rotation = "varimax")

Uniquenesses:
security_policy_understandable terms_and_conditions_displayed      company_owner_information         secure_payment_methods            transaction_details
                          0.25                           0.23                           0.33                           0.22                           0.24
             security_features          user_information_used     necessary_information_only      privacy_policy_presented          exeggerates_benefits
                          0.23                           0.31                           0.36                           0.31                           0.42
       truthful_about_offering       uses_misleading_tactics               takes_advantage            things_not_needed               abnormal_pricing
                          0.36                           0.01                           0.21                           0.40                           0.00
         actual_amount_billed                get_what_ordered      products_looked_available                 time_keeping
                          0.45                           0.22                           0.33                           0.21
```

Figure 13 Exploratory factor analysis - uniqueness, with abnormal pricing

4.2.10 Exploratory Factor Analysis – Factor Loadings, with Abnormal Pricing

```
Loadings:
                                Factor1 Factor2 Factor3 Factor4
security_policy_understandable  0.80
terms_and_conditions_displayed  0.83
company_owner_information       0.71
secure_payment_methods          0.75
transaction_details             0.72
security_features               0.77
user_information_used           0.61
necessary_information_only      0.62
privacy_policy_presented        0.57
exeggerates_benefits                    0.63
truthful_about_offering                 0.71
uses_misleading_tactics                 0.73            0.59
takes_advantage                         0.83
things_not_needed                       0.72
abnormal_pricing                        0.71            0.62
actual_amount_billed                            0.50
get_what_ordered                                0.69
products_looked_available                       0.60
time_keeping                                    0.75

                Factor1 Factor2 Factor3 Factor4
SS loadings       5.66    4.29    3.04    0.90
Proportion Var    0.30    0.23    0.16    0.05
Cumulative Var    0.30    0.52    0.68    0.73
```

Figure 13 Exploratory factor analysis – factor loadings table, with abnormal pricing

4.2.11 Principal Component Analysis - Importance of Components or Communality (Variance Accounted For), with Abnormal Pricing

```
> mydata <- read.csv("preprocessed_data_efa_abnormal_pricing.csv")
> fit <- princomp(mydata, cor=TRUE)
> summary(fit) # print variance accounted for
```

```
> summary(fit) # print variance accounted for
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14   Comp.15
Standard deviation     3.3701654 1.4158104 0.94856576 0.74994360 0.72577871 0.68223344 0.66725380 0.61912826 0.59495324 0.56761305 0.55251477 0.51880122 0.49107121 0.47663569 0.44559574
Proportion of Variance 0.5977903 0.1055010 0.04735668 0.02960081 0.02772393 0.02449697 0.02343303 0.02017473 0.01862997 0.01695708 0.01606698 0.01416604 0.01269215 0.01195693 0.01045029
Cumulative Proportion  0.5977903 0.7032913 0.75064796 0.78024877 0.80797270 0.83246967 0.85590270 0.87607743 0.89470740 0.91166448 0.92773146 0.94189749 0.95458965 0.96654657 0.97699687
                          Comp.16     Comp.17    Comp.18      Comp.19
Standard deviation     0.424430181 0.359375518 0.34478395 0.0942964213
Proportion of Variance 0.009481104 0.006797409 0.00625663 0.0004679903
Cumulative Proportion  0.986477971 0.993275380 0.99953201 1.0000000000
```

Figure 15 Principal components analysis – Importance of components or Communality (variance accounted for), with abnormal pricing.

4.2.12 Principal Components Analysis – Loadings, with Abnormal Pricing

> loadings(fit) # pc loadings

```
> loadings(fit) # pc loadings

Loadings:
                               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17 Comp.18 Comp.19
security_policy_understandable  0.232  0.232  0.272  0.162                0.151         0.571   0.184          0.344   0.213   0.149   0.149   0.343   0.267
terms_and_conditions_displayed  0.228  0.250  0.310  0.178                       -0.148  0.277  0.130         -0.291   0.352  -0.273   0.115          -0.343  -0.460
company_owner_information       0.230  0.126  0.334 -0.218         -0.146  0.325  0.180  0.201 -0.413   0.221  -0.390  -0.220   0.328           0.181           0.101
secure_payment_methods          0.245  0.240                       -0.229 -0.194 -0.293         -0.176  -0.182          -0.160  -0.343  -0.346           0.343   0.492
transaction_details             0.247  0.202         0.141  0.108 -0.107  0.119         0.253 -0.248  -0.245  -0.221   0.472  -0.259   0.305  -0.292   0.103  -0.348
security_features               0.242  0.230  0.139         0.165 -0.227 -0.237 -0.239 -0.256          0.172   0.116  -0.203  -0.103   0.101   0.436          -0.550
user_information_used           0.250                -0.105 -0.413  0.327  0.116 -0.100                0.319   0.132  -0.196          -0.249  -0.412   0.336  -0.317
necessary_information_only      0.238  0.138         -0.162 -0.477                -0.403 -0.409         -0.118          0.261   0.202   0.177          -0.358   0.210
privacy_policy_presented        0.248                       -0.171  0.501         0.178  0.266  0.160  -0.136  -0.262          -0.447           0.399  -0.231
exeggerates_benefits            0.212 -0.243  0.247  0.149  0.227  0.368  0.289  0.240 -0.523 -0.305          0.257          -0.179                   0.106
truthful_about_offering         0.207 -0.281  0.147 -0.364  0.240        -0.580                               0.307  -0.199   0.228                  -0.308  -0.140
uses_misleading_tactics         0.225 -0.366         0.366 -0.253 -0.203                0.121                 0.147                                  0.106                   0.711
takes_advantage                 0.208 -0.368         -0.298               -0.160         0.160         -0.564          -0.170   0.307   0.238   0.115   0.373
things_not_needed               0.192 -0.358         -0.130  0.347         0.477 -0.397  0.152  0.381                          -0.132  -0.169  -0.105  -0.253
abnormal_pricing                0.225 -0.358         0.396 -0.252 -0.231                                  0.153                                                  -0.700
actual_amount_billed            0.217        -0.345 -0.376 -0.159 -0.461  0.186  0.531 -0.211  0.170                  -0.207
get_what_ordered                0.243        -0.353         0.216  0.132                0.162 -0.183          0.261   0.369   0.362  -0.529   0.215          -0.109
products_looked_available       0.231        -0.371  0.365  0.198                      -0.315  0.169  -0.115  -0.533  -0.288   0.278          -0.172
time_keeping                    0.230        -0.464         0.214                       0.202 -0.140   0.308   0.315  -0.220           0.533                  0.262

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17 Comp.18 Comp.19
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053  0.053
Cumulative Var  0.053  0.105  0.158  0.211  0.263  0.316  0.368  0.421  0.474  0.526  0.579  0.632  0.684  0.737  0.789  0.842  0.895  0.947  1.000
```

Figure 16 Principal Components Analysis – loadings, with abnormal pricing

4.2.13 Principal Component Analysis – Scree Plot, with Abnormal Pricing

> plot(fit,type="lines") # scree plot

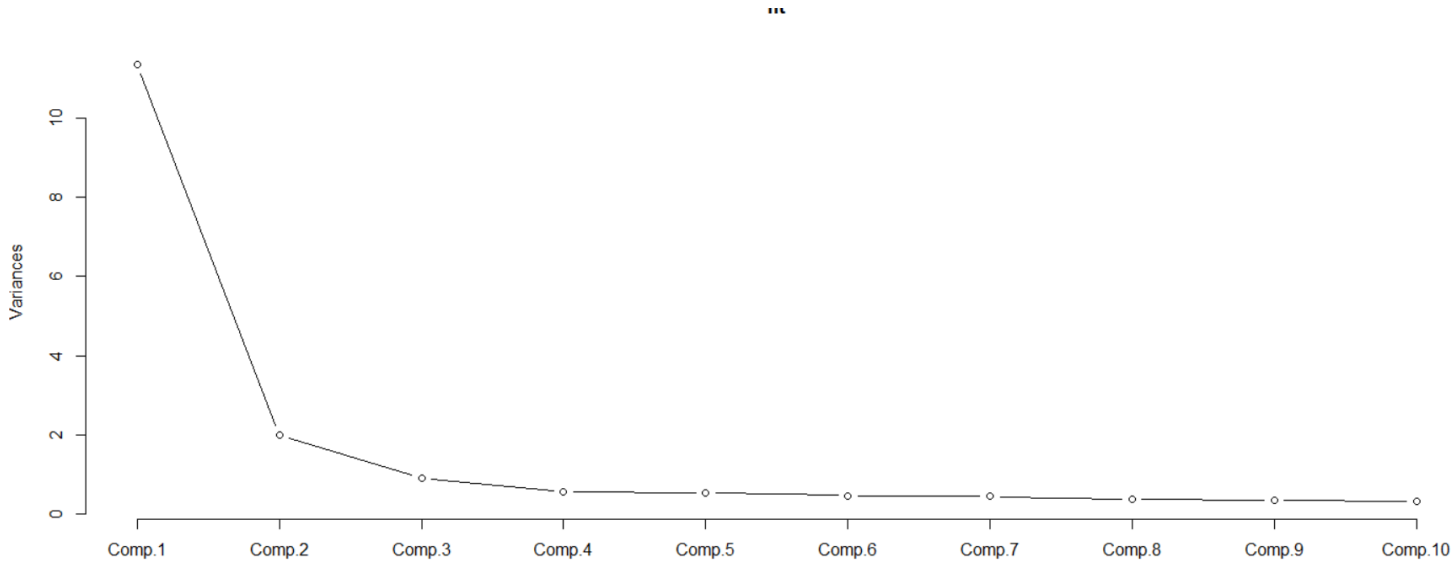


Figure 17 Principal components analysis - scree plot, with abnormal pricing

4.2.14 Principal Components Analysis – Distance Biplot, with Abnormal Pricing

```
> biplot(fit)
```

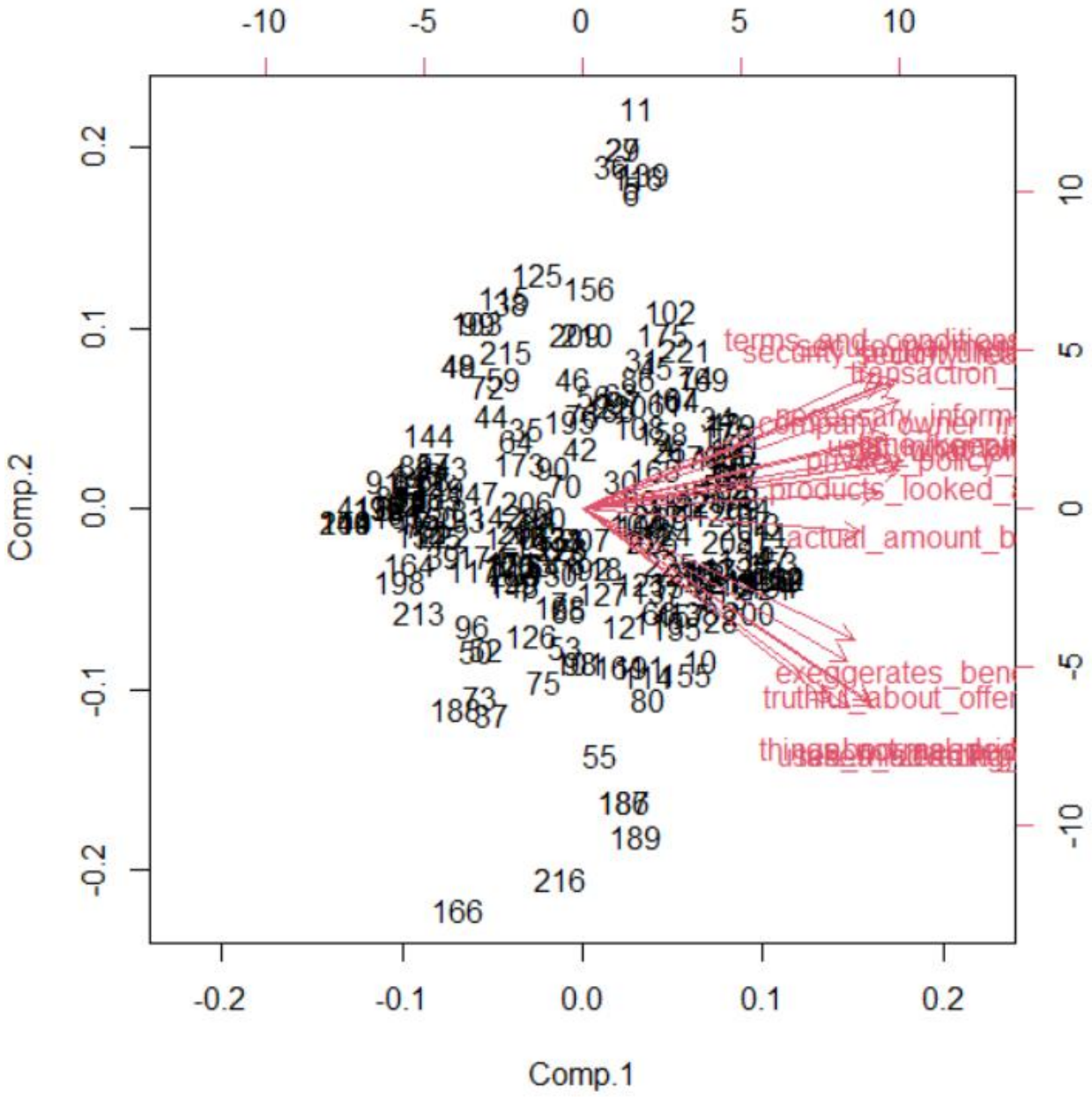


Figure 14 Principal Components Analysis - distance biplot, with abnormal pricing

4.2.15 Exploratory Factor Analysis – Eigenvalues (Scree Test), without Abnormal Pricing

```
# Determine Number of Factors to Extract
library(nFactors)
ev <- eigen(cor(mydata)) # get eigenvalues
ap <- parallel(subject=nrow(mydata),var=ncol(mydata),
  rep=100,cent=.05)
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)
plotnScree(nS)
```

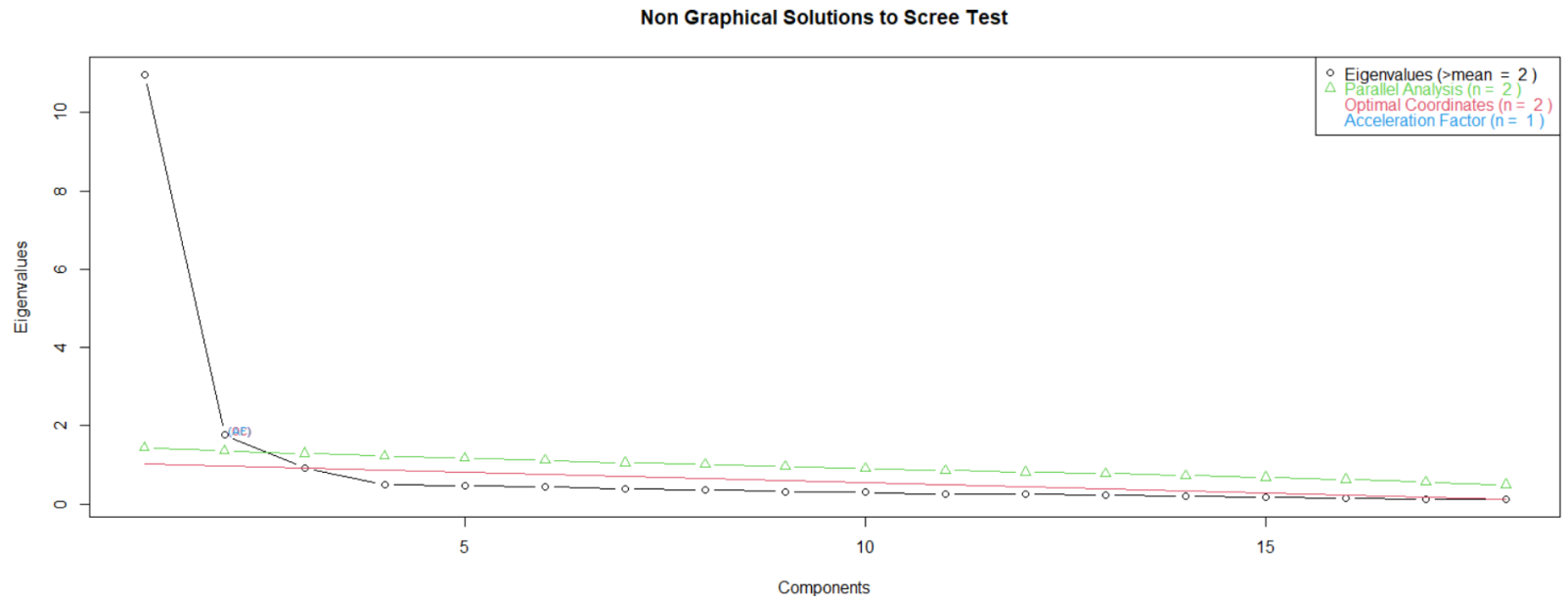


Figure 19 Exploratory factor analysis - eigenvalues (scree test), without abnormal pricing

4.2.16 Exploratory Factor Analysis – Uniqueness, without Abnormal Pricing

What is not measured (i.e., the uniqueness or error term variance) (Gorsuch, 1983).

```
> # Maximum Likelihood Factor Analysis
> # entering raw data and extracting 3 factors,
> # with varimax rotation
> fit <- factanal(mydata, 4, rotation="varimax")
> print(fit, digits=2, cutoff=.5, sort=TRUE)
```

```
factanal(x = mydata, factors = 4, rotation = "varimax")

Uniquenesses:
security_policy_understandable terms_and_conditions_displayed     company_owner_information          secure_payment_methods 
                          0.27                           0.23                           0.34                           0.21 
             transaction_details                security_features           user_information_used 
                          0.24                           0.22                           0.00 
       necessary_information_only          privacy_policy_presented          exeggerates_benefits 
                          0.32                           0.29                           0.40 
       truthful_about_offering          uses_misleading_tactics                 takes_advantage 
                          0.37                           0.28                           0.24 
                things_not_needed              actual_amount_billed                get_what_ordered 
                          0.37                           0.23                           0.22 
        products_looked_available                  time_keeping 
                          0.29                           0.26
```

Figure 15 Exploratory factor analysis - uniqueness, without abnormal pricing

4.2.17 Exploratory Factor Analysis – Factor Loadings, without Abnormal Pricing

```
Loadings:
                              Factor1 Factor2 Factor3 Factor4
security_policy_understandable 0.76
terms_and_conditions_displayed 0.81
company_owner_information      0.68
secure_payment_methods         0.73
transaction_details            0.70
security_features              0.76
necessary_information_only     0.56
privacy_policy_presented       0.51
exeggerates_benefits                   0.67
truthful_about_offering                0.71
uses_misleading_tactics                0.75
takes_advantage                        0.82
things_not_needed                      0.75
actual_amount_billed                           0.69
get_what_ordered                               0.70
products_looked_available                      0.68
time_keeping                                   0.72
user_information_used                                  0.68

                Factor1 Factor2 Factor3 Factor4
SS loadings        5.01    3.92    3.42    0.87
Proportion Var     0.28    0.22    0.19    0.05
Cumulative Var     0.28    0.50    0.69    0.73

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 195.06 on 87 degrees of freedom.
The p-value is 2.89e-10
```

Figure 16 Exploratory factor analysis – factor loadings table, without abnormal pricing

4.2.18 Principal Component Analysis - Importance of Components or Communality (Variance Accounted For), without Abnormal Pricing

```
> mydata <- read.csv("preprocessed_data_efa.csv")
> fit <- princomp(mydata, cor=TRUE)
> summary(fit) # print variance accounted for
```

```
> summary(fit) # print variance accounted for
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5     Comp.6     Comp.7     Comp.8     Comp.9    Comp.10    Comp.11    Comp.12    Comp.13    Comp.14    Comp.15
Standard deviation     3.2924354 1.3895884 1.00079794 0.79803248 0.66944856 0.60255945 0.57692310 0.56629626 0.54957153 0.52578763 0.48891728 0.48455265 0.46025397 0.45601899 0.44190416
Proportion of Variance 0.6022295 0.1072753 0.05564425 0.03538088 0.02489785 0.02017099 0.01849113 0.01781619 0.01677938 0.01535848 0.01328001 0.01304396 0.01176854 0.01155296 0.01084885
Cumulative Proportion  0.6022295 0.7095048 0.76514908 0.80052996 0.82542781 0.84559880 0.86408993 0.88190612 0.89868550 0.91404398 0.92732399 0.94036795 0.95213649 0.96368945 0.97453830
                          Comp.16     Comp.17     Comp.18
Standard deviation     0.421346888 0.379322412 0.369989081
Proportion of Variance 0.009862956 0.007993638 0.007605107
Cumulative Proportion  0.984401255 0.992394893 1.000000000
```

Figure 17 Principal components analysis – Importance of components or Communality (variance accounted for), without abnormal pricing.

4.2.19 Principal Components Analysis – Loadings, without Abnormal Pricing

> loadings(fit) # pc loadings

```
> loadings(fit) # pc loadings

Loadings:
                           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17 Comp.18
security_policy_understandable  0.225  0.180  0.367         0.584         0.295         0.142  0.130           0.381                                        0.207   0.330
terms_and_conditions_displayed  0.224  0.249  0.398         0.302 -0.137        -0.273                                -0.542                 -0.111          -0.307  -0.367
company_owner_information       0.235  0.176  0.284        -0.157  0.177 -0.599  0.356  0.425 -0.243                                -0.128  0.101                   0.126
secure_payment_methods          0.250  0.226  0.117  0.122 -0.399         0.185 -0.122 -0.173                 -0.166          -0.415 -0.128          -0.171   0.592
transaction_details             0.251  0.213         0.150 -0.317               -0.215         0.103   0.450  0.273                  0.407   0.435  -0.133  -0.191
security_features               0.251  0.198  0.165  0.152 -0.347         0.210  0.134 -0.198         -0.345  0.216   0.107  0.166  -0.216  -0.162   0.423  -0.410
user_information_used           0.247        -0.119 -0.532         0.137         0.149         0.191  -0.323 -0.125   0.321 -0.421   0.253   0.237   0.115  -0.133
necessary_information_only      0.241        -0.112 -0.536 -0.135  0.195  0.384  0.176 -0.101          0.205 -0.112  -0.433  0.327          -0.156
privacy_policy_presented        0.247        -0.165 -0.410        -0.224 -0.207 -0.497         -0.236          0.398   0.141          -0.350  -0.168
exeggerates_benefits            0.223 -0.315  0.125         0.169  0.125 -0.415        -0.717  0.136                  -0.141          -0.144           0.113   0.131
truthful_about_offering         0.218 -0.347  0.128                -0.641         0.184                 -0.239  0.199  -0.317 -0.113   0.310          -0.227
uses_misleading_tactics         0.224 -0.376                                                      -0.301          -0.159  0.557   0.243   0.391  -0.340           0.116
takes_advantage                 0.212 -0.423                -0.139 -0.107  0.176 -0.154  0.311                  -0.271          0.116  -0.277   0.530   0.359   0.105
things_not_needed               0.207 -0.410         0.132         0.450  0.114         0.234  0.291           0.185          -0.297 -0.247  -0.224  -0.319  -0.248
actual_amount_billed            0.246        -0.317  0.147  0.176 -0.179  0.126  0.554         -0.159  0.255           0.313          -0.336   0.200  -0.274
get_what_ordered                0.253        -0.307  0.164        -0.244 -0.115         0.126  0.338  0.383  -0.223 -0.113  -0.188   0.103  -0.393   0.426
products_looked_available       0.238        -0.348  0.294  0.221  0.292        -0.178         -0.558 -0.201          -0.332 -0.227   0.145                   -0.112
time_keeping                    0.242        -0.400  0.198                -0.119 -0.125  0.128  0.414  -0.435                  0.481                  -0.233   0.192

                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17 Comp.18
SS loadings      1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
Proportion Var   0.056  0.056  0.056  0.056  0.056  0.056  0.056  0.056  0.056   0.056   0.056   0.056   0.056   0.056   0.056   0.056   0.056   0.056
Cumulative Var   0.056  0.111  0.167  0.222  0.278  0.333  0.389  0.444  0.500   0.556   0.611   0.667   0.722   0.778   0.833   0.889   0.944   1.000
```

Figure 23 Principal Components Analysis – loadings, without abnormal pricing

4.2.20 Principal Component Analysis – Scree Plot, without Abnormal Pricing

> plot(fit,type="lines") # scree plot

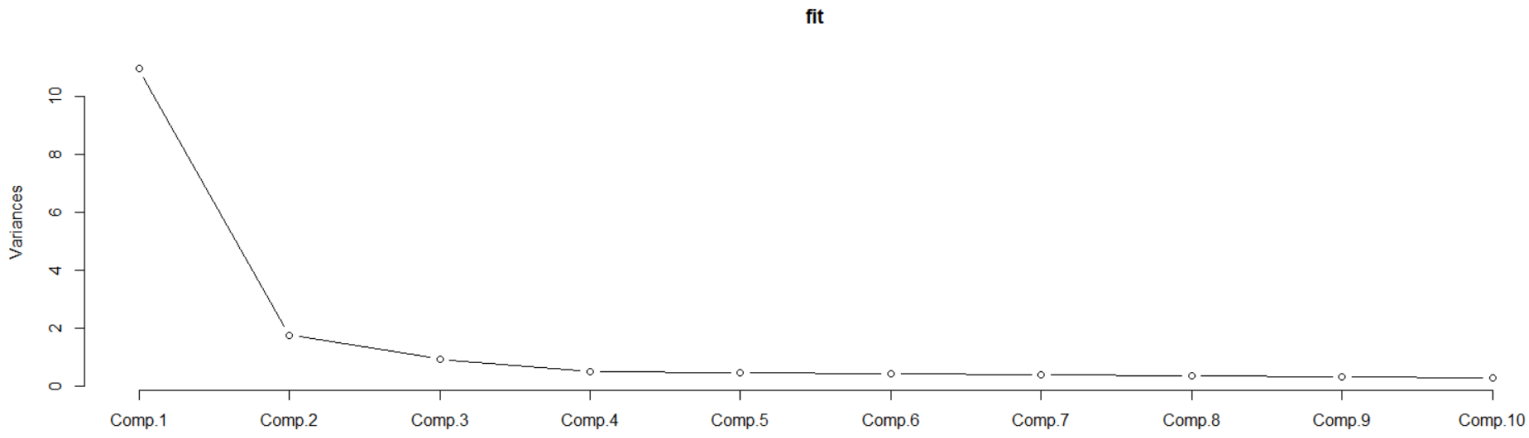


Figure 18 Principal components analysis - scree plot, without abnormal pricing

4.2.21 Principal Components Analysis – Distance Biplot, without Abnormal Pricing

```
> biplot(fit)
```

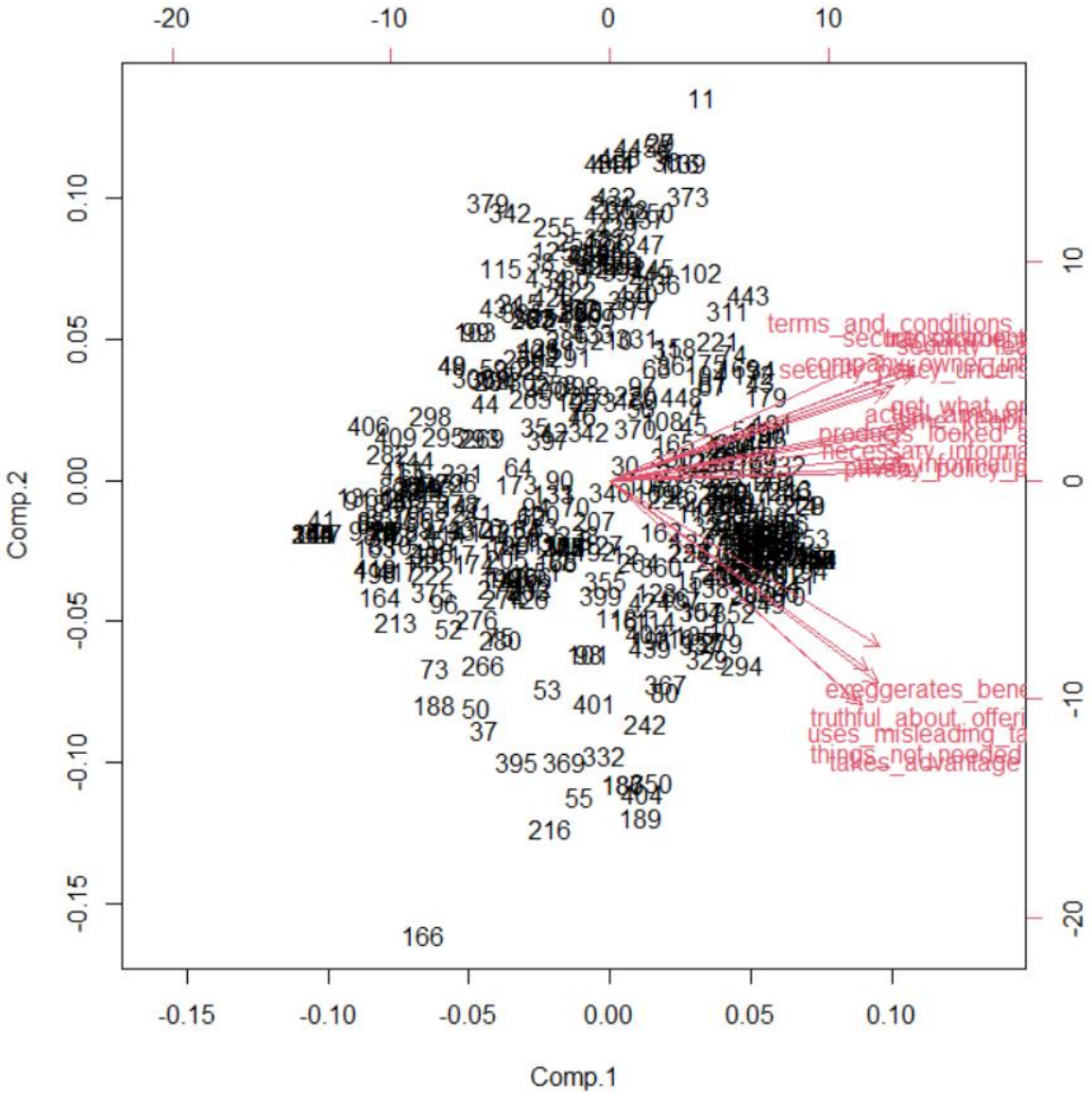


Figure 19 Principal Components Analysis - distance biplot, without abnormal pricing

4.2.22 First Study Data Reliability

```
> library(ltm)
Loading required package: MASS
Loading required package: msm
Loading required package: polycor
> cronbach.alpha(mydata)

Cronbach's alpha for the 'mydata' data-set

Items: 18
Sample units: 448
alpha: 0.959
```

Figure 2620 Data Reliability

4.3 Development of a Model for Estimation of Trustworthiness of an Online Shop

4.3.1 Introduction

After the exploratory factor analysis stage carried out as described in section 3.4.2 and arriving at a rough idea that the items we got at the items generation stage, described in section 3.4.1, were actually true representatives of the constructs of trust, we carried out a second study for confirmatory purposes.

4.3.2 Responses of the second study

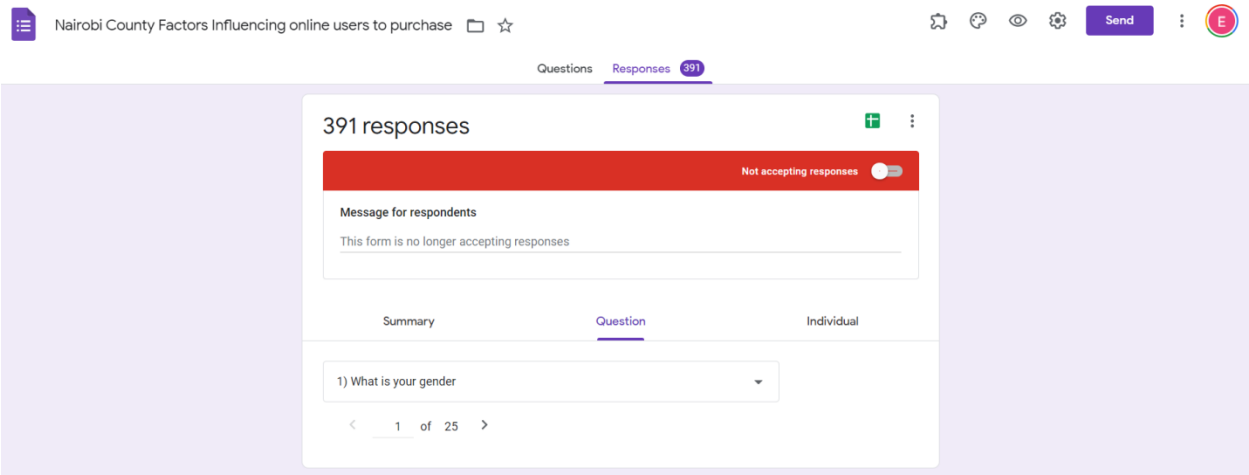


Figure 27 Nairobi county responses

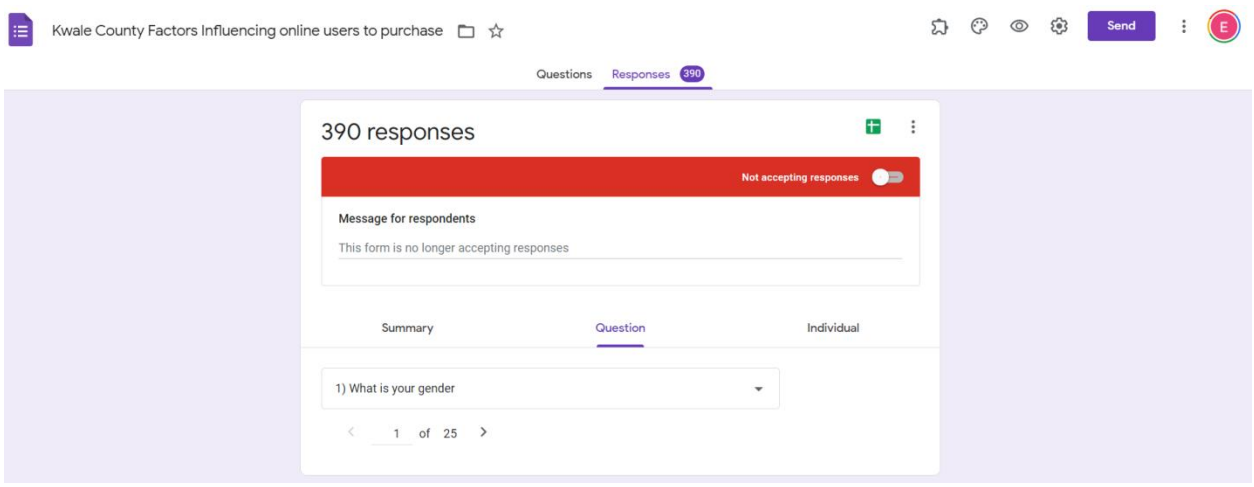


Figure 28 Kwale county responses

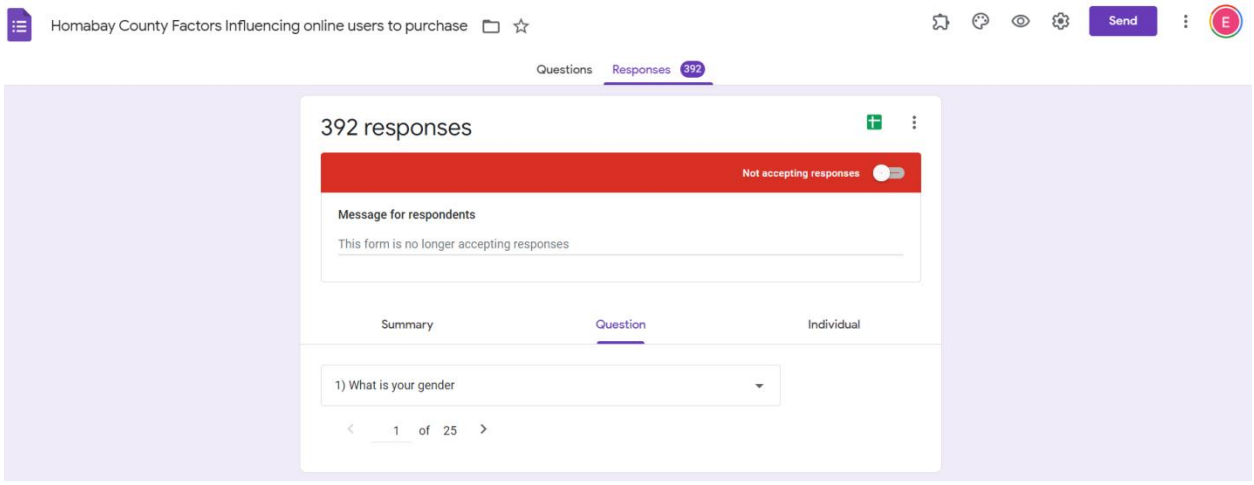


Figure 29 Homabay County Responses

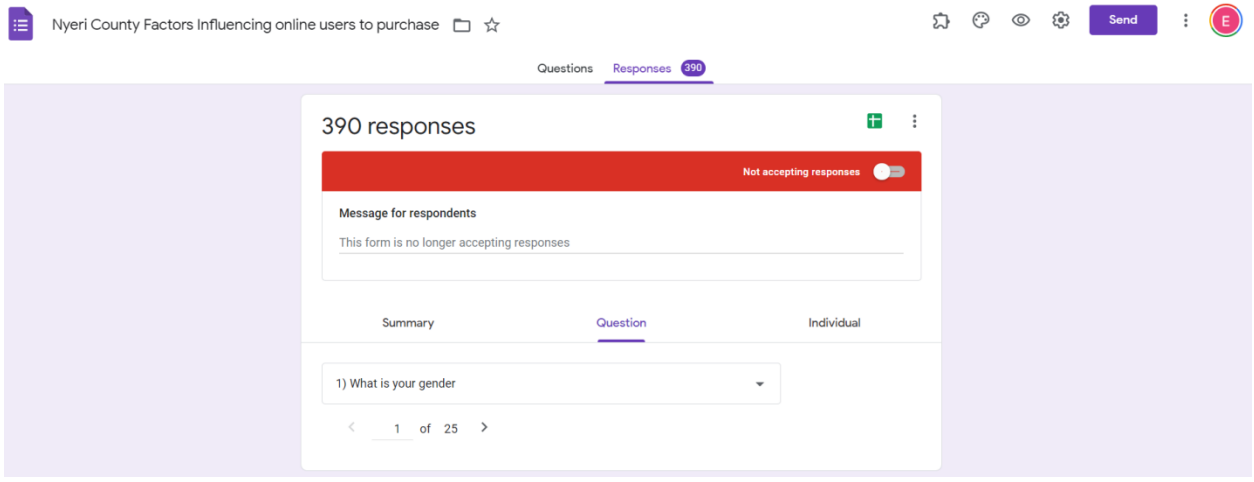


Figure 30 Nyeri County Responses

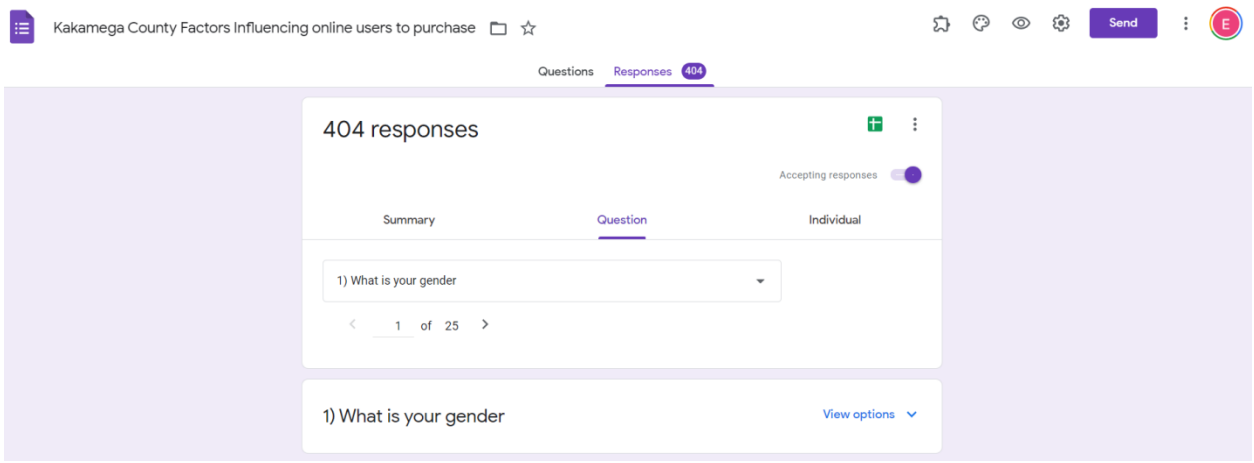


Figure 31 Kakamega County Responses

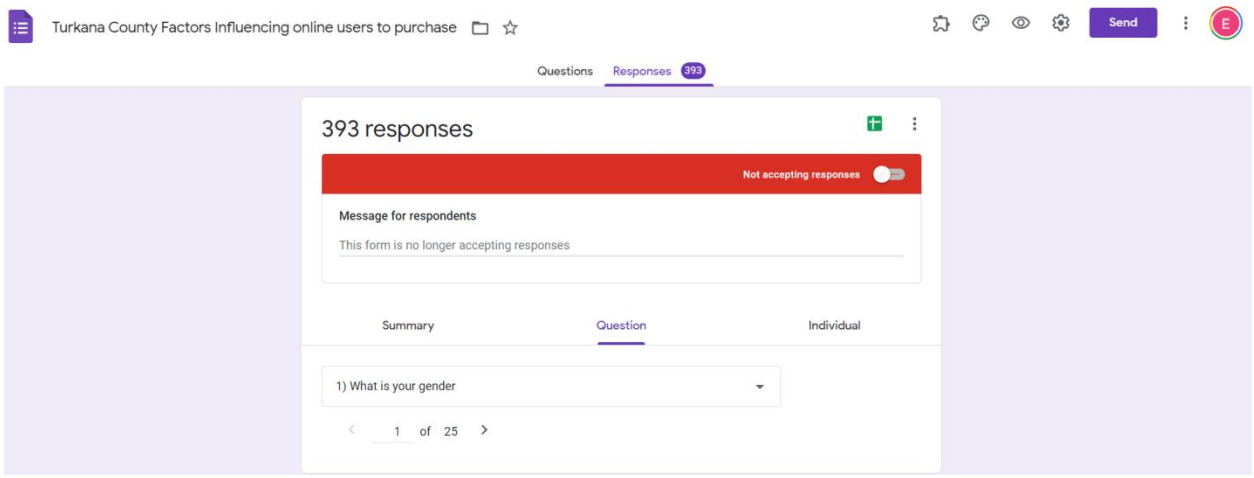


Figure 32 Turkana County Responses

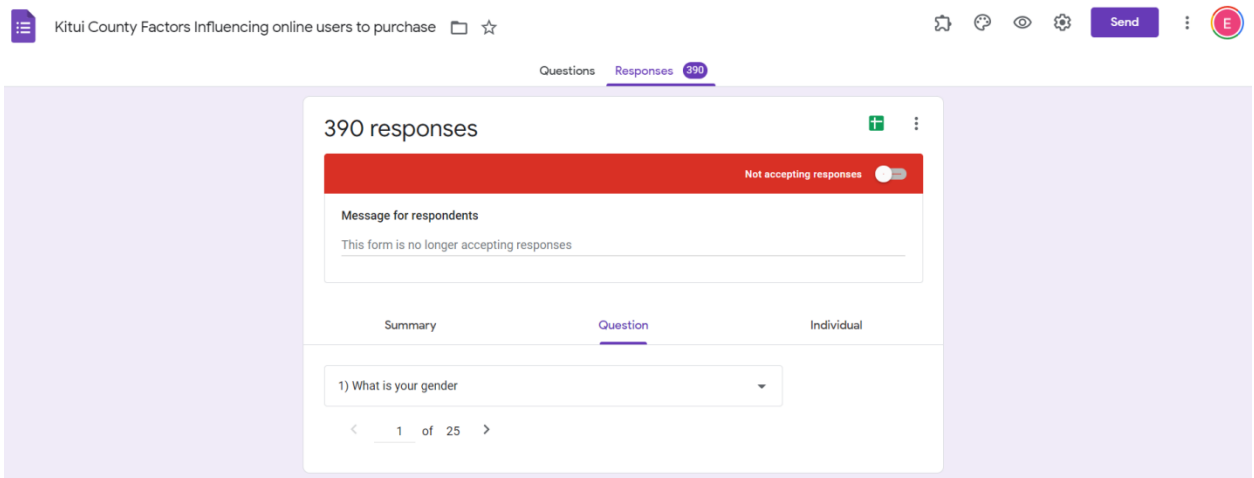


Figure 33 Kitui County Responses

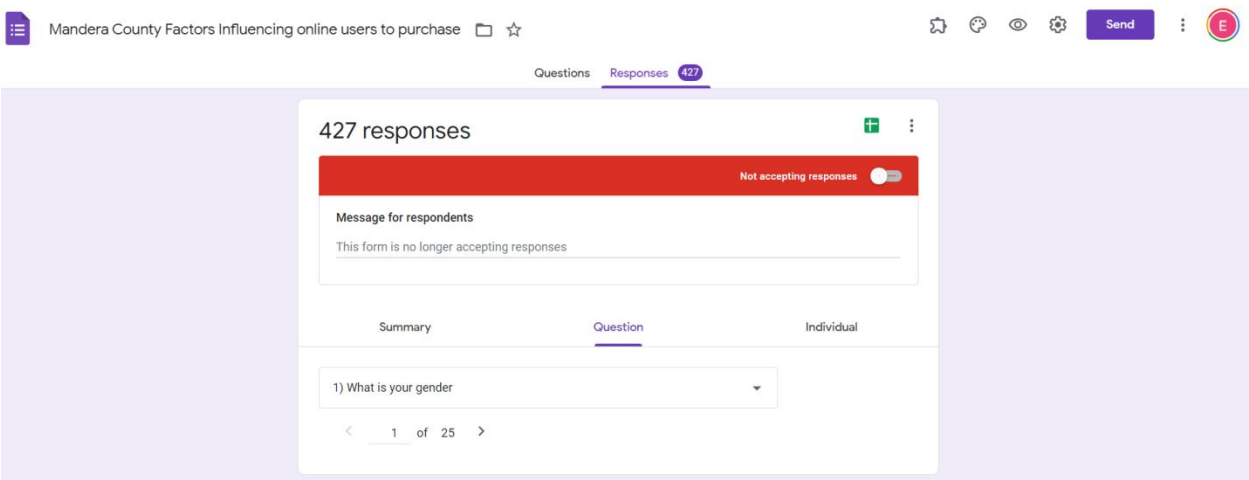


Figure 34 Mandera COunty Responses

4.3.3 Four factor Trust Model

Security, non-deception, reliability, privacy

```
> model <- '

+     security  =~ security_policy_understandable + terms_and_conditions_displayed +
company_owner_information + secure_payment_methods + transaction_details  + security_features

+     non_deception  =~ exeggerates_benefits + truthful_about_offering + uses_misleading_tactics +
takes_advantage + things_not_needed

+     reliability  =~ actual_amount_billed + get_what_ordered + products_looked_available +
time_keeping

+     privacy  =~ user_information_used + necessary_information_only + privacy_policy_presented

+ '

> mydata <- read.csv("preprocessed_data_cfa.csv")

> fit <- cfa(model, data = mydata)

> summary(fit, fit.measures = TRUE, standardized = TRUE)
```

```
> summary(fit, fit.measures = TRUE, standardized = TRUE)
lavaan 0.6-9 ended normally after 42 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        42

  Number of observations                          2104

Model Test User Model:

  Test statistic                              1453.112
  Degrees of freedom                               129
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             33686.452
  Degrees of freedom                               153
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.961
  Tucker-Lewis Index (TLI)                       0.953
```

Figure 35 Four Factor Model p-value, TLI, CLI

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -49471.462
  Loglikelihood unrestricted model (H1)     -48744.906

  Akaike (AIC)                               99026.923
  Bayesian (BIC)                             99264.290
  Sample-size adjusted Bayesian (BIC)        99130.852

Root Mean Square Error of Approximation:

  RMSEA                                          0.070
  90 Percent confidence interval - lower         0.067
  90 Percent confidence interval - upper         0.073
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.033

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

Figure 36 Four Factor Model RMSEA, SRMR

```
> model_loadings <- inspect(fit, what = "std")[["lambda"]]
> model_loadings
```

```
> model_loadings
                                secrty nn_dcp rlblty privcy
security_policy_understandable  0.778  0.000  0.000  0.000
terms_and_conditions_displayed  0.812  0.000  0.000  0.000
company_owner_information       0.806  0.000  0.000  0.000
secure_payment_methods          0.866  0.000  0.000  0.000
transaction_details             0.872  0.000  0.000  0.000
security_features               0.877  0.000  0.000  0.000
exeggerates_benefits            0.000  0.807  0.000  0.000
truthful_about_offering         0.000  0.828  0.000  0.000
uses_misleading_tactics         0.000  0.891  0.000  0.000
takes_advantage                 0.000  0.884  0.000  0.000
things_not_needed               0.000  0.853  0.000  0.000
actual_amount_billed            0.000  0.000  0.852  0.000
get_what_ordered                0.000  0.000  0.881  0.000
products_looked_available       0.000  0.000  0.822  0.000
time_keeping                    0.000  0.000  0.863  0.000
user_information_used           0.000  0.000  0.000  0.865
necessary_information_only      0.000  0.000  0.000  0.839
privacy_policy_presented        0.000  0.000  0.000  0.853
>
```

Figure 37 Four Factor Model Loadings

> semPaths(fit, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

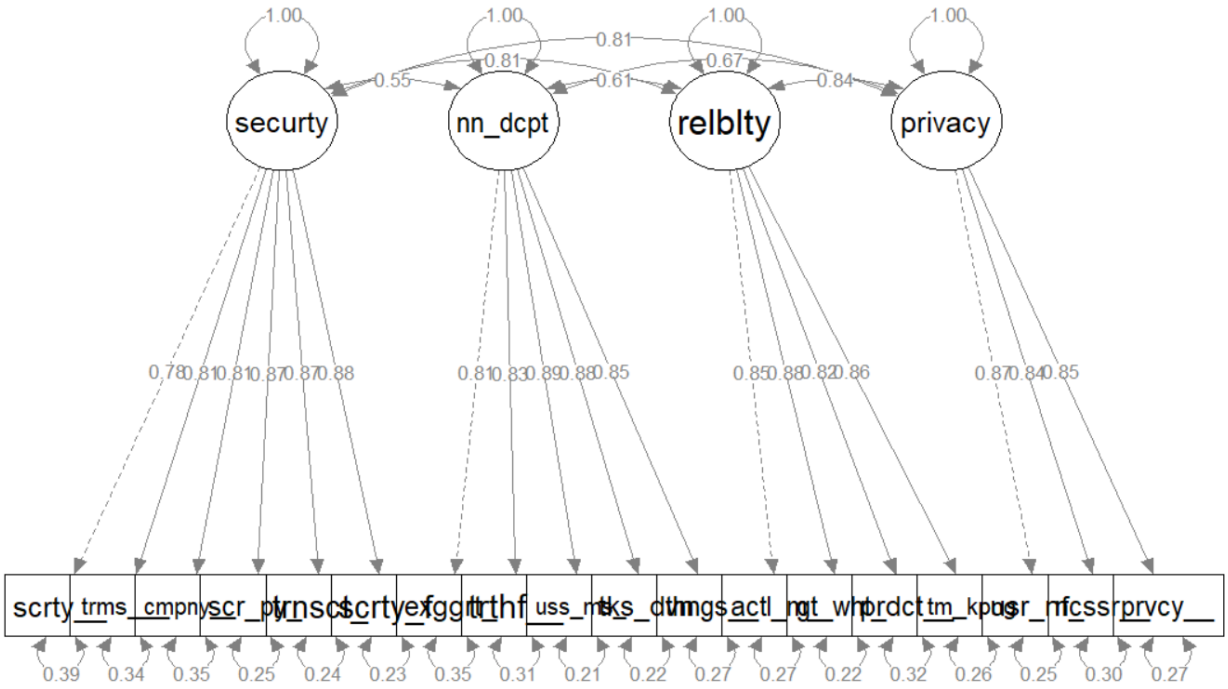


Figure 38 Four Factor Model path diagram

4.3.4 Three factor Trust Model

Three factors (privacy+security,fulfillment, non-deception)

> model <- '

+ security_and_privacy =~ security_policy_understandable + terms_and_conditions_displayed + company_owner_information + secure_payment_methods + transaction_details + security_features + user_information_used + necessary_information_only + privacy_policy_presented

+ non_deception =~ exeggerates_benefits + truthful_about_offering + uses_misleading_tactics + takes_advantage + things_not_needed

+ reliability =~ actual_amount_billed + get_what_ordered + products_looked_available + time_keeping

```
+ '

> fit <- cfa(model, data = mydata)

> summary(fit, fit.measures = TRUE, standardized = TRUE)
```

```
> summary(fit, fit.measures = TRUE, standardized = TRUE)
lavaan 0.6-9 ended normally after 42 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        39

  Number of observations                          2104

Model Test User Model:

  Test statistic                              2841.106
  Degrees of freedom                               132
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             33686.452
  Degrees of freedom                               153
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.919
  Tucker-Lewis Index (TLI)                       0.906
```

Figure 39 Three Factor Model p-value, CFI, TLI

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -50165.459
  Loglikelihood unrestricted model (H1)      -48744.906

  Akaike (AIC)                               100408.918
  Bayesian (BIC)                             100629.330
  Sample-size adjusted Bayesian (BIC)        100505.423

Root Mean Square Error of Approximation:

  RMSEA                                          0.099
  90 Percent confidence interval - lower         0.096
  90 Percent confidence interval - upper         0.102
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.054

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

Figure 40 Three Factor Model RMSEA, SRMR

```
> model_loadings <- inspect(fit, what = "std")[["lambda"]]
> model_loadings
```

```
> model_loadings
                                  scrt__ nn_dcp rlblty
security_policy_understandable     0.770  0.000  0.000
terms_and_conditions_displayed     0.791  0.000  0.000
company_owner_information          0.798  0.000  0.000
secure_payment_methods             0.848  0.000  0.000
transaction_details                0.860  0.000  0.000
security_features                  0.857  0.000  0.000
user_information_used              0.763  0.000  0.000
necessary_information_only         0.757  0.000  0.000
privacy_policy_presented           0.765  0.000  0.000
exeggerates_benefits               0.000  0.807  0.000
truthful_about_offering            0.000  0.827  0.000
uses_misleading_tactics            0.000  0.892  0.000
takes_advantage                    0.000  0.883  0.000
things_not_needed                  0.000  0.854  0.000
actual_amount_billed               0.000  0.000  0.852
get_what_ordered                   0.000  0.000  0.882
products_looked_available          0.000  0.000  0.823
time_keeping                       0.000  0.000  0.862
>
```

Figure 41 Three Factor Model Loadings

> semPaths(fit, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

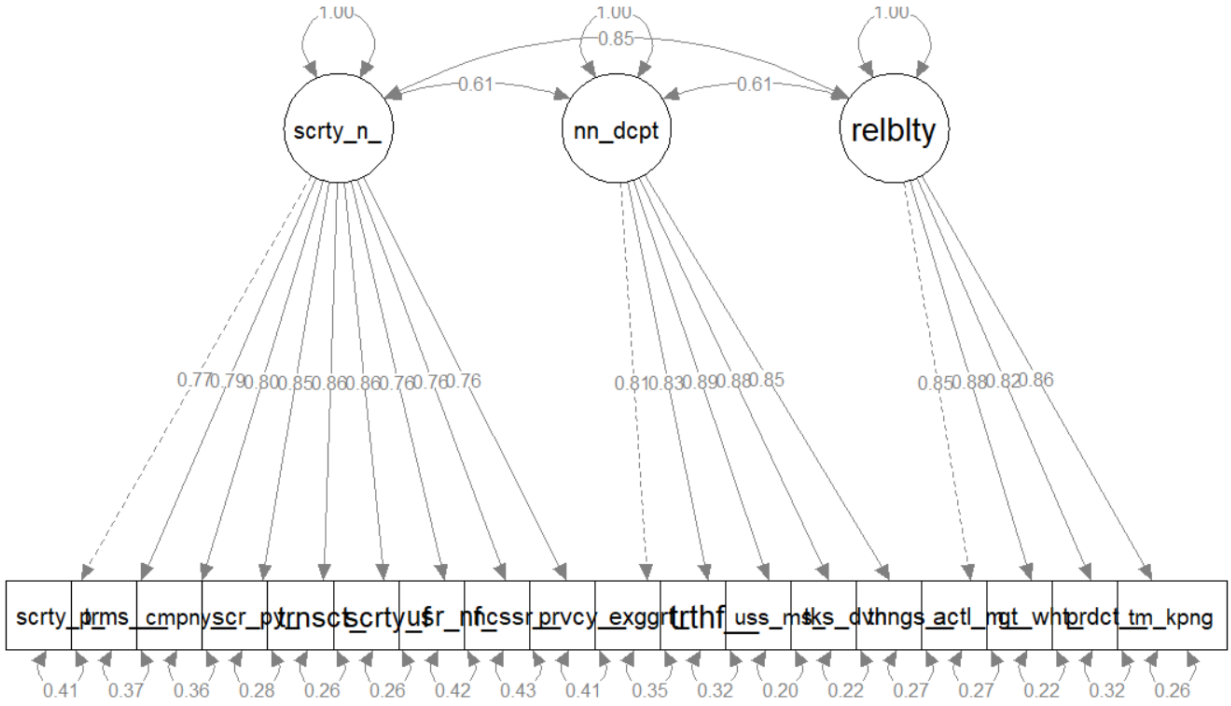


Figure 42 Three Factor Model Path Diagram

4.3.5 Two factor Trust Model

Privacy + security, fulfillment +non-deception

```
model <- '

   security_and_privacy  =~ security_policy_understandable + terms_and_conditions_displayed +
company_owner_information + secure_payment_methods + transaction_details  + security_features +
user_information_used + necessary_information_only + privacy_policy_presented

   reliability_non_deception  =~ exeggerates_benefits + truthful_about_offering +
uses_misleading_tactics + takes_advantage + things_not_needed + actual_amount_billed +
get_what_ordered + products_looked_available + time_keeping

'

> fit <- cfa(model, data = mydata)

> summary(fit, fit.measures = TRUE, standardized = TRUE)
```

```
> summary(fit, fit.measures = TRUE, standardized = TRUE)
lavaan 0.6-9 ended normally after 37 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        37

  Number of observations                          2104

Model Test User Model:

  Test statistic                              7317.158
  Degrees of freedom                               134
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             33686.452
  Degrees of freedom                               153
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.786
  Tucker-Lewis Index (TLI)                       0.755
```

Figure 43 Two Factor Model p-value, CFI, TLI

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -52403.485
  Loglikelihood unrestricted model (H1)     -48744.906

  Akaike (AIC)                              104880.970
  Bayesian (BIC)                            105090.079
  Sample-size adjusted Bayesian (BIC)       104972.526

Root Mean Square Error of Approximation:

  RMSEA                                          0.160
  90 Percent confidence interval - lower         0.157
  90 Percent confidence interval - upper         0.163
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.106

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

Figure 44 Two Factor Model RMSEA, SRMR

```
> model_loadings <- inspect(fit, what = "std")[["lambda"]]
> model_loadings
```

```
> model_loadings
                                  scrt__ rlbl__
security_policy_understandable     0.774  0.000
terms_and_conditions_displayed     0.796  0.000
company_owner_information          0.801  0.000
secure_payment_methods             0.847  0.000
transaction_details                0.856  0.000
security_features                  0.858  0.000
user_information_used              0.760  0.000
necessary_information_only         0.757  0.000
privacy_policy_presented           0.761  0.000
exeggerates_benefits               0.000  0.789
truthful_about_offering            0.000  0.774
uses_misleading_tactics            0.000  0.819
takes_advantage                    0.000  0.791
things_not_needed                  0.000  0.774
actual_amount_billed               0.000  0.725
get_what_ordered                   0.000  0.734
products_looked_available          0.000  0.711
time_keeping                       0.000  0.714
>
```

Figure 45 Two Factor Model Loadings

> semPaths(fit, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

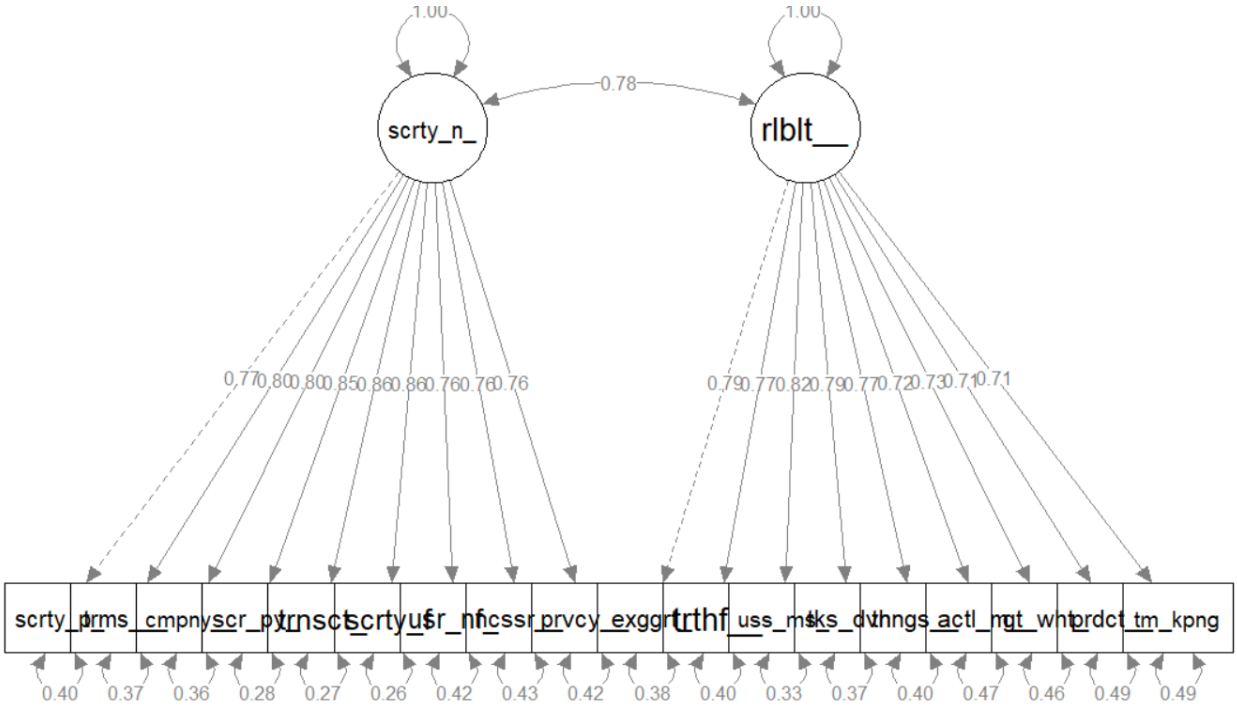


Figure 46 Two Factor Model Path Diagram

4.3.6 One factor Trust Model

Trust

> model <- '

+ trust =~ security_policy_understandable + terms_and_conditions_displayed + company_owner_information + secure_payment_methods + transaction_details + security_features + user_information_used + necessary_information_only + privacy_policy_presented + exeggerates_benefits + truthful_about_offering + uses_misleading_tactics + takes_advantage + things_not_needed + actual_amount_billed + get_what_ordered + products_looked_available + time_keeping

+ '

> fit <- cfa(model, data = mydata)

> summary(fit, fit.measures = TRUE, standardized = TRUE)

```
> summary(fit, fit.measures = TRUE, standardized = TRUE)
lavaan 0.6-9 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        36

  Number of observations                          2104

Model Test User Model:

  Test statistic                              9085.389
  Degrees of freedom                               135
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             33686.452
  Degrees of freedom                               153
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.733
  Tucker-Lewis Index (TLI)                       0.698
```

Figure 47 One Factor Model p-value, CFI, TLI

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -53287.601
  Loglikelihood unrestricted model (H1)      -48744.906

  Akaike (AIC)                               106647.201
  Bayesian (BIC)                             106850.659
  Sample-size adjusted Bayesian (BIC)        106736.283

Root Mean Square Error of Approximation:

  RMSEA                                          0.178
  90 Percent confidence interval - lower         0.174
  90 Percent confidence interval - upper         0.181
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.102

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

Figure 48 One Factor Model RMSEA, SRMR

```
> model_loadings <- inspect(fit, what = "std")[["lambda"]]
> model_loadings
```

```
> model_loadings
                                  trust
security_policy_understandable 0.736
terms_and_conditions_displayed 0.732
company_owner_information      0.757
secure_payment_methods         0.807
transaction_details            0.821
security_features              0.816
user_information_used          0.789
necessary_information_only     0.770
privacy_policy_presented       0.796
exeggerates_benefits           0.663
truthful_about_offering        0.626
uses_misleading_tactics        0.646
takes_advantage                0.601
things_not_needed              0.592
actual_amount_billed           0.794
get_what_ordered               0.813
products_looked_available      0.760
time_keeping                   0.782
>
```

Figure 49 One Factor Model Loadings

> semPaths(fit, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

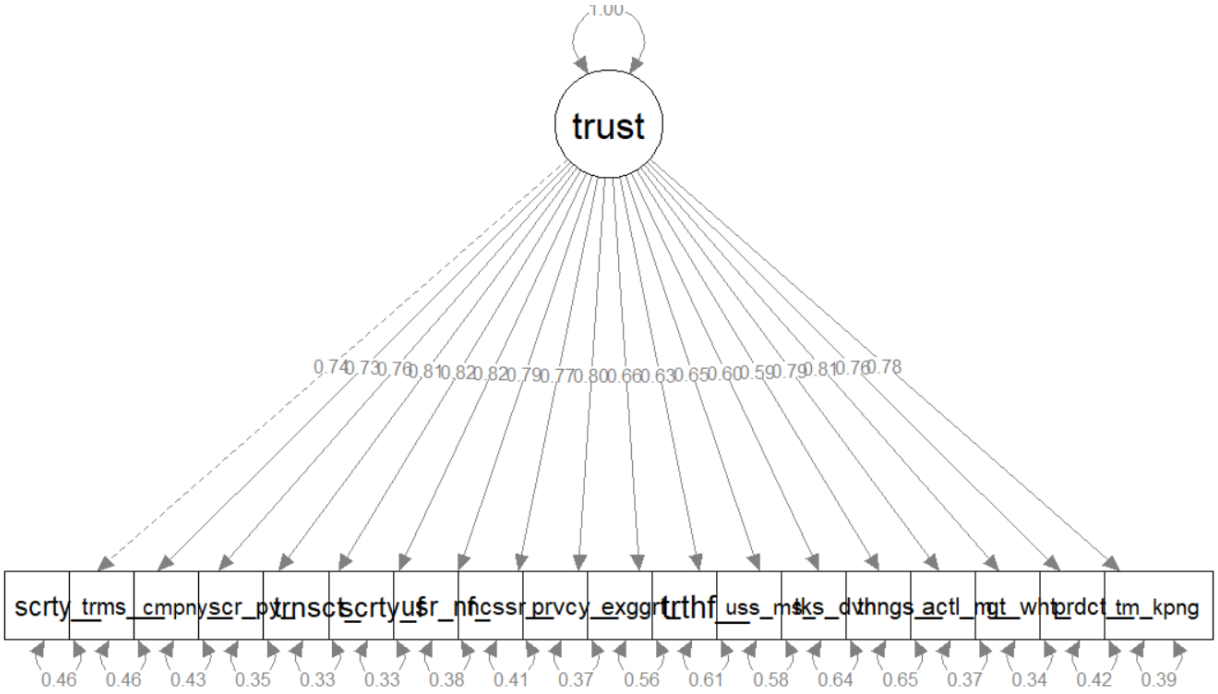


Figure 50 One Factor Model Path Diagram

4.3.7 Four factors, one second-order factor

1st order: security, non-deception, reliability, privacy

2nd order: trust

```
model <- '
  security  =~ security_policy_understandable + terms_and_conditions_displayed +
company_owner_information + secure_payment_methods + transaction_details  + security_features
  non_deception  =~ exeggerates_benefits + truthful_about_offering + uses_misleading_tactics +
takes_advantage + things_not_needed
  reliability  =~ actual_amount_billed + get_what_ordered + products_looked_available + time_keeping
  privacy  =~ user_information_used + necessary_information_only + privacy_policy_presented
  trust =~ security + non_deception + reliability + privacy
'
mydata <- read.csv("preprocessed_data_cfa.csv")
fit <- cfa(model, data = mydata)
summary(fit, fit.measures = TRUE, standardized = TRUE)
```

```
> summary(fit, fit.measures = TRUE, standardized = TRUE)
lavaan 0.6-9 ended normally after 34 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        40

  Number of observations                          2104

Model Test User Model:

  Test statistic                              1502.782
  Degrees of freedom                               131
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             33686.452
  Degrees of freedom                               153
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.959
  Tucker-Lewis Index (TLI)                       0.952
```

Figure 51 Four factors, one second-order factor p-value, CFI, TLI

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -49496.297
  Loglikelihood unrestricted model (H1)     -48744.906

  Akaike (AIC)                               99072.593
  Bayesian (BIC)                             99298.657
  Sample-size adjusted Bayesian (BIC)        99171.573

Root Mean Square Error of Approximation:

  RMSEA                                          0.071
  90 Percent confidence interval - lower         0.067
  90 Percent confidence interval - upper         0.074
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.036

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

Figure 52 Four factors, one second-order factor RMSEA, SRMR

```
> model_loadings <- inspect(fit, what = "std")[["lambda"]]
> model_loadings
```

```
> model_loadings
                                secrty nn_dcp rlblty privcy trust
security_policy_understandable  0.779  0.000  0.000  0.000     0
terms_and_conditions_displayed  0.812  0.000  0.000  0.000     0
company_owner_information       0.806  0.000  0.000  0.000     0
secure_payment_methods          0.865  0.000  0.000  0.000     0
transaction_details             0.871  0.000  0.000  0.000     0
security_features               0.877  0.000  0.000  0.000     0
exeggerates_benefits            0.000  0.808  0.000  0.000     0
truthful_about_offering         0.000  0.828  0.000  0.000     0
uses_misleading_tactics         0.000  0.892  0.000  0.000     0
takes_advantage                 0.000  0.883  0.000  0.000     0
things_not_needed               0.000  0.854  0.000  0.000     0
actual_amount_billed            0.000  0.000  0.852  0.000     0
get_what_ordered                0.000  0.000  0.881  0.000     0
products_looked_available       0.000  0.000  0.822  0.000     0
time_keeping                    0.000  0.000  0.863  0.000     0
user_information_used           0.000  0.000  0.000  0.866     0
necessary_information_only      0.000  0.000  0.000  0.839     0
privacy_policy_presented        0.000  0.000  0.000  0.852     0
>
```

Figure 53 Four factors, one second-order factor Loadings

> semPaths(fit, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

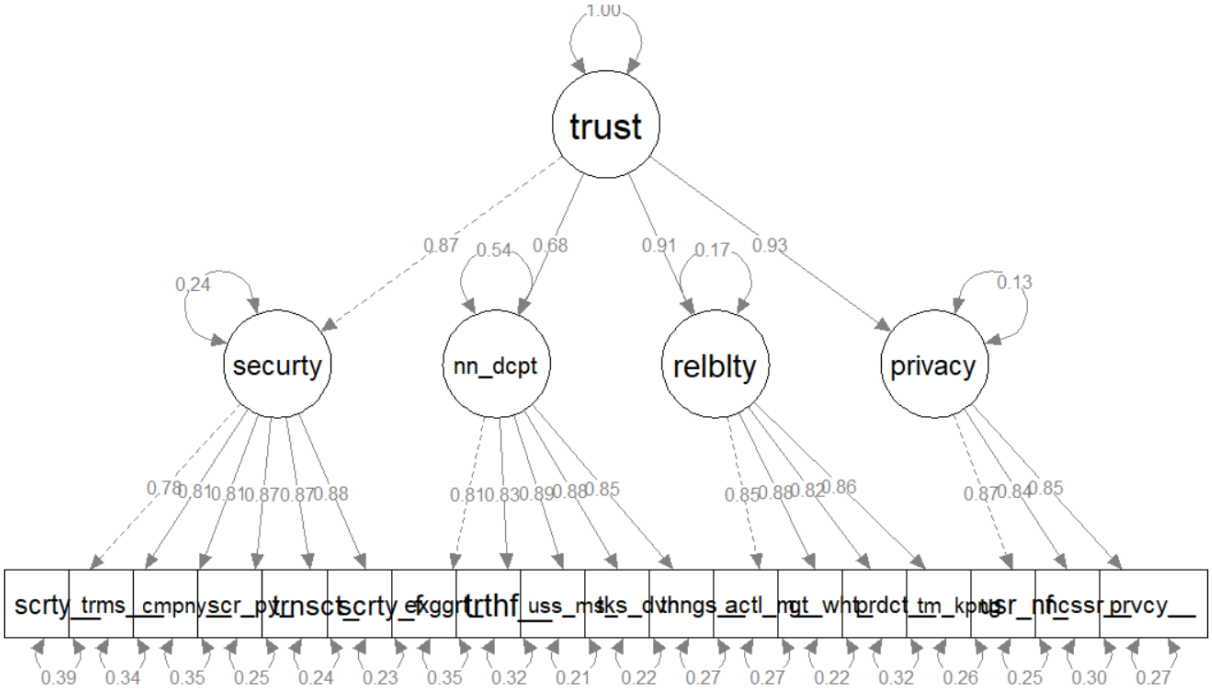


Figure 54 Four factors, one second-order model path diagram

4.3.8 Fit Statistics

Table 6 Models Fit Statistics for confirmatory factor analyses chart

Item to Measure	Description	Cut off for good model fit	1 factor Model	2 factor Model	3 factor Model	4 factor Model	4 factor Model With 2nd Order Reflective	Passed
X^2	Chi-Square	p-value > 0.05	0.000	0.000	0.000	0.000	0.000	OK (Usually sensitive to large sample size, this is large sample size)
CFI	Comparative Fit Index	CFI ≥.90	0.733	0.786	0.919	0.961	0.959	OK
(N)NFI TLI	(Non) Normed-Fit Index Tucker Lewis index	NFI ≥ 0.95 NNFI ≥ 0.95	0.698	0.755	0.906	0.953	0.952	OK
RMSEA	Root Mean Square Error of Approximation	RMSEA < 0.08	0.178	0.160	0.099	0.070	0.071	OK
(S)RMR	(Standardized) Root Mean Square Residual	SRMR <0.08	0.102	0.106	0.054	0.033	0.036	OK

4.3.9 Data Reliability

```
> library(ltm)
> cronbach.alpha(mydata)

Cronbach's alpha for the 'mydata' data-set

Items: 18
Sample units: 2104
alpha: 0.956

>
```

Figure 55 Cronbach's alpha

Table 7 Data Reliability

Measure	Cut off for reliable data	Findings	Passed
Cronbach's / Coefficient Alpha	> 0.7	0.956	OK

4.3.10 Model Validity

Table 8 Convergent Validity

Measure	Cut off for valid model	Findings	Passed
Convergent validity	Average Variance Extracted (AVE) > 0.5	Security: 0.704083333 Reliability: 0.744025 Privacy: 0.728333333 Non deception: 0.7268	OK OK OK OK

Table 9 Divergent Validity

Measure	Cut off for valid model	Construct	Security	Reliability	Privacy	Non Deception	Passed
Convergent Validity	The Square Root of Average Variance Extracted (AVE) is greater than all the correlations between a construct and its counterparts.	Security	**0.839096737**				OK
		Reliability	0.81	**0.862568838**			OK
		Privacy	0.81	0.84	**0.853424474**		OK
		Non Deception	0.55	0.61	0.67	**0.852525659**	OK

4.3.11 Choosing the model to adopt

We adopt the four factors, one second-order model presented in figure 53 above because from the fit statistics chart presented in table 3, its fit statistics meet the suggested scientific cut offs values as per literature, (Kline, 2005), and also it gives us a way of aggregating all the first order constructs into one aggregate trust value.

The factor loadings are presented in a clearer way in tables 4 to 8.

Table 10 Security Indicators and Loadings

Indicators	Factor Loadings
$\mathbf{Y_1}$: Site's Security Policy easy to understand	**0.79**
$\mathbf{Y_2}$: Site's Terms and Conditions are displayed:	**0.81**
$\mathbf{Y_3}$: Site owner's background information displayed:	**0.81**
$\mathbf{Y_4}$: The site offers secure payment methods	**0.87**
$\mathbf{Y_5}$: You can confirm the details of the transaction before paying	**0.87**
$\mathbf{Y_6}$: The site has adequate security features	**0.88**

Table 11 Reliability Indicators and loadings

Indicators	Factor Loadings
$\mathbf{Y_{10}}$: The price shown on the site is the actual amount billed	**0.85**
$\mathbf{Y_{11}}$: You get what you ordered from this site	**0.88**
$\mathbf{Y_{12}}$: The products I looked at were available	**0.86**
$\mathbf{Y_{13}}$: Promises to do something by a certain time, they do it.	**0.86**

Table 12 non deception indicators and loadings

Indicators	Factor Loading
$\mathbf{Y_{14}}$: The site exaggerates the benefits and characteristics of its offerings	**0.81**
$\mathbf{Y_{15}}$: The site is not entirely truthful about its offerings	**0.83**

$\mathbf{Y_{16}}$: The site uses misleading tactics to convince consumers to buy its products	**0.89**
$\mathbf{Y_{17}}$: This site takes advantage of less experienced consumers to make them purchase:	**0.88**
$\mathbf{Y_{18}}$: This site attempts to persuade you to buy things that you do not need	**0.85**

Table 13Privacy Indicators and Loadings

Indicators	Factor Loading
$\mathbf{Y_7}$: The site clearly explains how user information is used	**0.87**
$\mathbf{Y_9}$: Only the personal information necessary for the transaction to be completed needs to be provided	**0.84**
$\mathbf{Y_8}$: Information regarding the privacy policy is clearly presented	**0.85**

Table 14 Second Order (Trust) Factor Loadings

Indicators	Factor Loadings
$\mathbf{Y_{19}}$: Security	**0.87**
$\mathbf{Y_{20}}$: Non deception	**0.68**
$\mathbf{Y_{21}}$: Reliability	**0.91**
$\mathbf{Y_{22}}$: Privacy	**0.93**

4.4 Result on how to embed the new model as a new parameter, called trust adjustment factor, into the classical collaborative recommendation algorithm to create a new trust enhanced collaborative recommendation algorithm

4.4.1 Introduction

We compute the trust value from the model by considering the factor loadings as variable coefficients as described in section 3.6.

```
function compute_trust_score(){
    $query = "select * from wp_affiliates";
    $result = $this->select($query);
    while($row=mysqli_fetch_array($result)){
       $id = $row['id'];
       $y_1 = $row['y_1'];
       $y_2 = $row['y_2'];
       $y_3 = $row['y_3'];
       $y_4 = $row['y_4'];
       $y_5 = $row['y_5'];
       $y_6 = $row['y_6'];
       $security = (($y_1*$Cy_1/$sum_security)+($y_2*$Cy_2/$sum_security)+($y_3*$Cy_3/$sum_security)+($y_4*$Cy_4/$sum_security)+($y_5*$Cy_5/$sum_security)
              +($y_6*$Cy_6/$sum_security));
       $y_7 = $row['y_7'];
       $y_8 = $row['y_8'];
       $y_9 = $row['y_9'];
       $privacy = (($y_7*$Cy_7/$sum_privacy)+($y_8*$Cy_8/$sum_privacy)+($y_9*$Cy_9/$sum_privacy));
       $y_10 = $row['y_10'];
       $y_11 = $row['y_11'];
       $y_12 = $row['y_12'];
       $y_13 = $row['y_13'];
       $reliability = (($y_10*$Cy_10/$sum_reliability)+($y_11*$Cy_11/$sum_reliability)+($y_12*$Cy_12/$sum_reliability)+($y_13*$Cy_13/$sum_reliability));
       $y_14 = $row['y_14'];
       $y_15 = $row['y_15'];
       $y_16 = $row['y_16'];
       $y_17 = $row['y_17'];
       $y_18 = $row['y_18'];
       $non_deception = (($y_14*$Cy_14/$sum_non_deception)+($y_15*$Cy_15/$sum_non_deception)+($y_16*$Cy_16/$sum_non_deception)+
       ($y_17*$Cy_17/$sum_non_deception) +($y_18*$Cy_18/$sum_non_deception));

       $trust = ($security*$CSec/$sum_second)  - ($non_deception*$Cnd/$sum_second) +($reliability*$Crel/$sum_second) +($privacy*$Cpr/$sum_second);
       $this->write($this->log, "SECURITY: $security | PRIVACY: $privacy |  RELIABILITY: $reliability | NON DECEPTION: $non_deception | TRUST: $trust");

       $query = "update wp_affiliates set trust_score = '$trust' where id = $id limit 1";
       $this->write($this->log, "About to update trust score");
       $this->update($query);
```

Figure 21 PHP code for computation of trust value based on the four factors, one second-order trust model

Showing rows 0 - 9 (10 total, Query took 0.0010 seconds.)

```
select CONVERT(trust_score,DECIMAL(10,2)) as trust_score from wp_affiliates order by 1 desc limit 10
```

+ Options

trust_score
0.67
0.60
0.54
0.53
0.53
0.50
0.46
0.46
0.43
0.43

Top n vendors sampling formula including confidence level etc

Figure 5722 Top ten most trustworthy affiliates to arrive at trust score cut off (10 is debatable – depends on stringency considerations)

```
import pandas as pd
from surprise import Dataset
from surprise import Reader
import os

colnames=['item', 'user', 'rating', 'trust']
rating_df_with_trust_values = pd.read_csv("ratings.csv", names=colnames, header=None)

#Filter out untrustworthy ratings
rating_df_with_trust_values = rating_df_with_trust_values[(rating_df_with_trust_values['trust'] >= 0.43)]

#Remove the trust values just before feeding into the recommender system
rating_df = rating_df_with_trust_values.iloc[:, 0:3]

#Load into surprise recommender engine Dataset
reader = Reader(rating_scale=(1, 5),sep=',')
data = Dataset.load_from_df(rating_df, reader)
```

Figure 58 Python Script for loading data into the recommender system

```
# recommender.py

from surprise import KNNWithMeans

# To use item-based cosine similarity
sim_options = {
    "name": "cosine",
    "user_based": False,  # Compute  similarities between items
}
algo = KNNWithMeans(k=40, min_k=1,sim_options=sim_options)
```

Figure 59 The python recommender algorithm

```
#recommendation.py
#load modules
from load_data import data
from recommender import algo

#train algo
trainingSet = data.build_full_trainset()

algo.fit(trainingSet)

#predict rating
prediction = algo.predict(1, 2)# predict rating of user 1 on item 2
prediction.est
```

Figure 60 Sample code -python recommendation program

```
#recommendation.py
#load modules
from load_data import data
from recommender import algo

#train algo
trainingSet = data.build_full_trainset()

algo.fit(trainingSet)

#predict rating
prediction = algo.predict(1, 2)# predict rating of user 2 on item 1
prediction.est
```

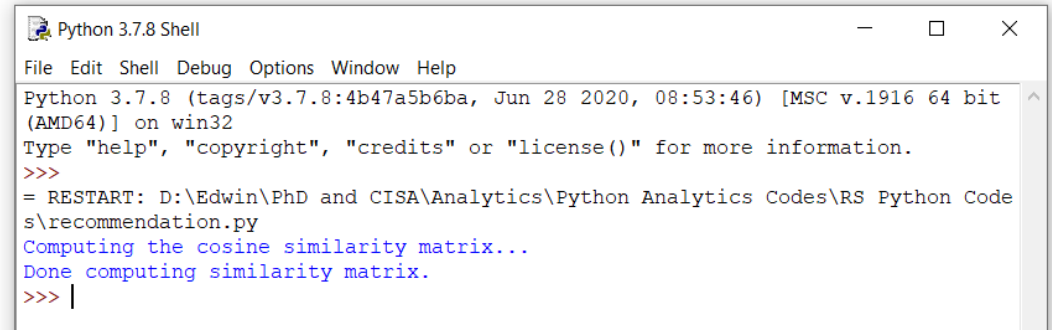


Figure 61 Sample output for computing similarity

```
#recommendation.py
#load modules
from load_data import data
from recommender import algo

#train algo
trainingSet = data.build_full_trainset()

algo.fit(trainingSet)

#predict rating
prediction = algo.predict(1, 2)# predict rating of user 2 on item 1
prediction.est
```

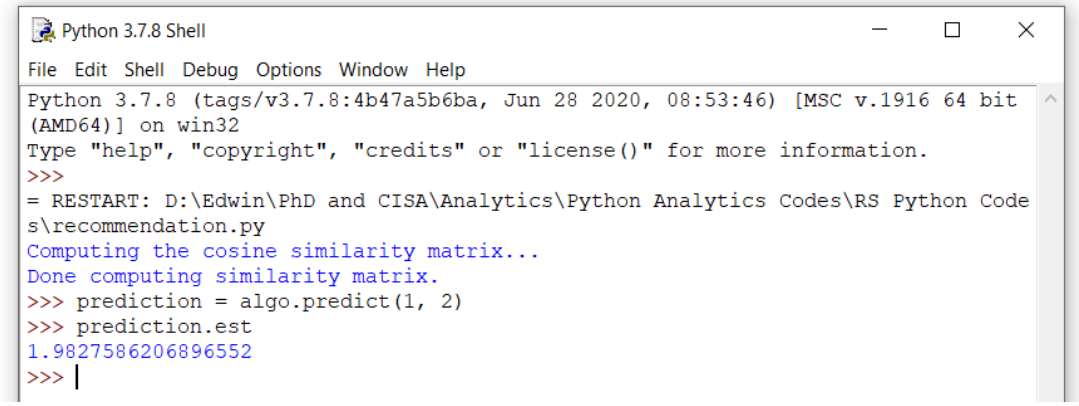


Figure 62 Sample output of a rating prediction

```
#recommendation_accuracy.py
#load modules
from load_data import data
from recommender import algo
from surprise.model_selection import train_test_split
from surprise import accuracy

#Train the algorithm on the trainset, and predict ratings for the testset
trainset, testset = train_test_split(data, test_size=.25)
algo.fit(trainset)
predictions = algo.test(testset)
# Then compute RMSE
accuracy.rmse(predictions)
accuracy.mae(predictions)
```

Figure 63 Sample python code for computing prediction accuracy of a recommendation system

```
#recommendation.py
#load modules
from load_data import data
from recommender import algo
from surprise.model_selection import train_test_split
from surprise import accuracy

#Train the algorithm on the trainset, and predict ratings for the testset
trainset, testset = train_test_split(data, test_size=.25)
algo.fit(trainset)
predictions = algo.test(testset)
# Then compute RMSE
accuracy.rmse(predictions)
accuracy.mae(predictions)
```

```
Python 3.7.8 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Edwin/PhD and CISA/Analytics/Python Analytics Codes/RS Python Code
s/recommendation_accuracy.py
Computing the cosine similarity matrix...
Done computing similarity matrix.
RMSE: 0.9220
MAE:  0.9622
>>>
```

Figure 64 Sample output for computing prediction accuracy of recommender system

4.5 Results on the deployment of the new algorithm into an empirical setup

We were able to deploy the new algorithm for a production setup empirically as described in section 3.7.

Below are a few screenshots of the store where the algorithm was residing in and operating in.

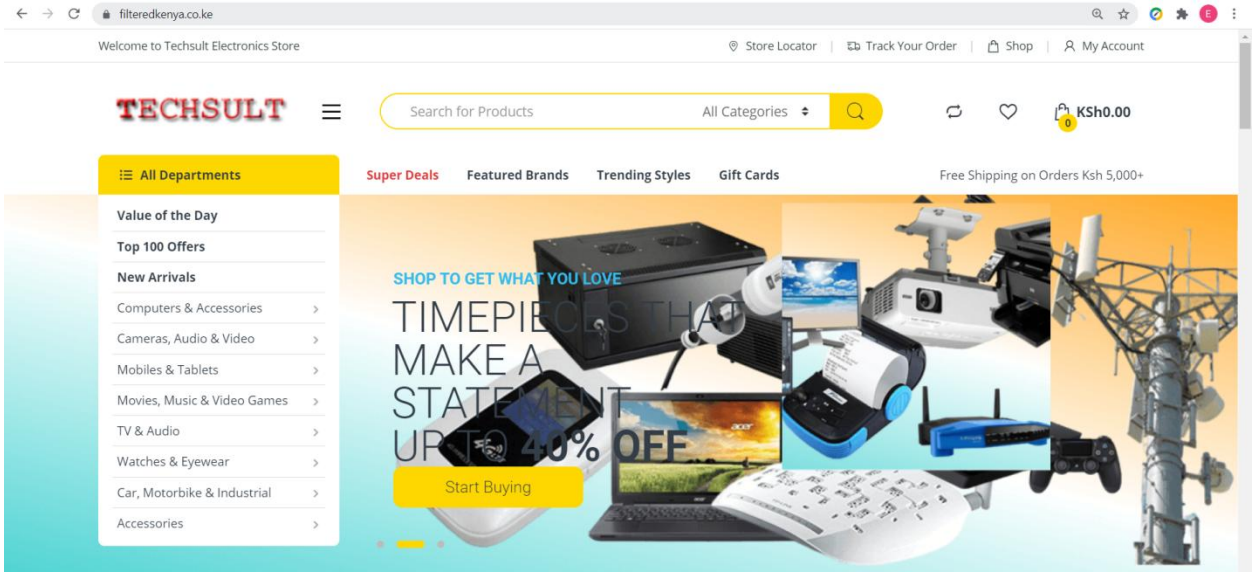


Figure 65 Prototype – user facing side: Store landing page web view

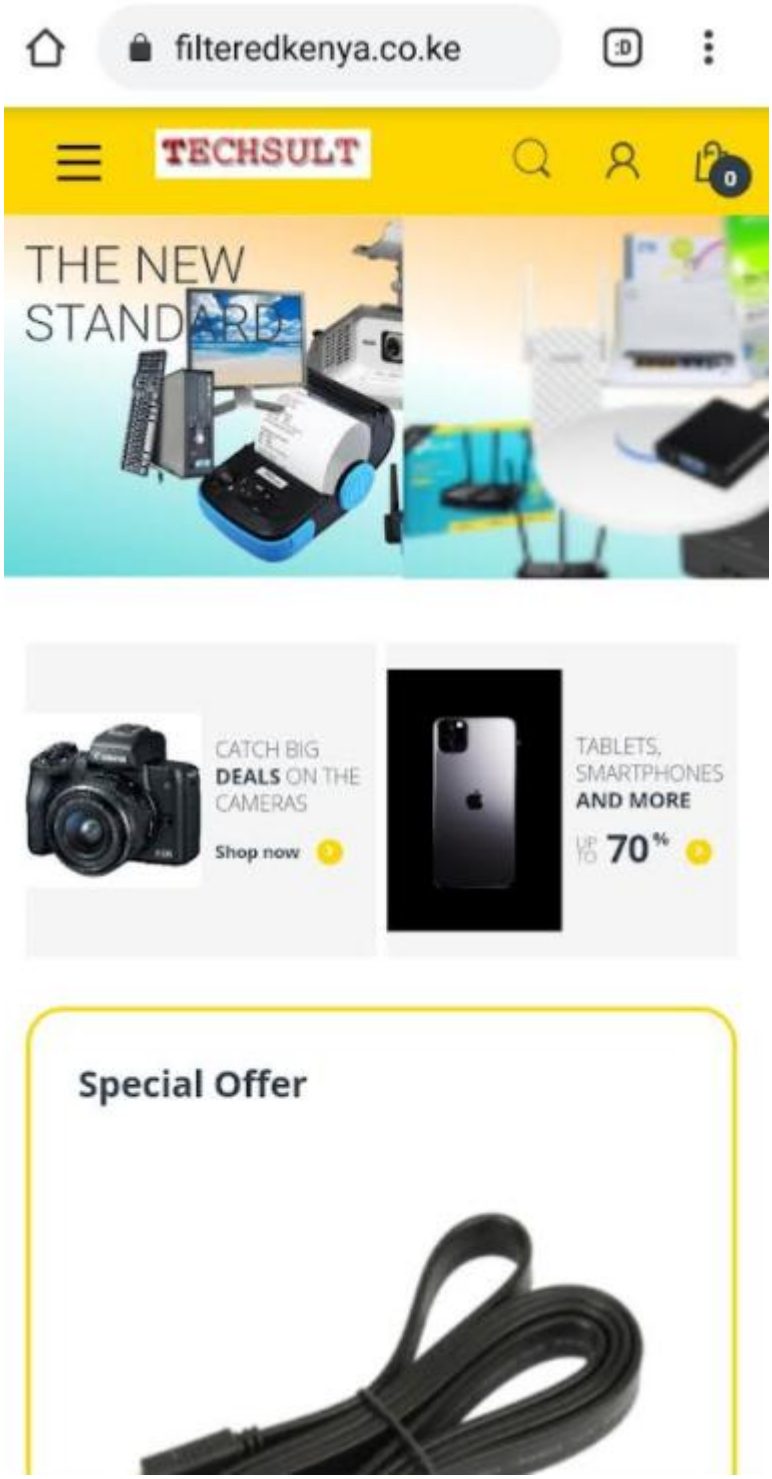


Figure 66 Prototype – user facing side: Store landing page mobile view

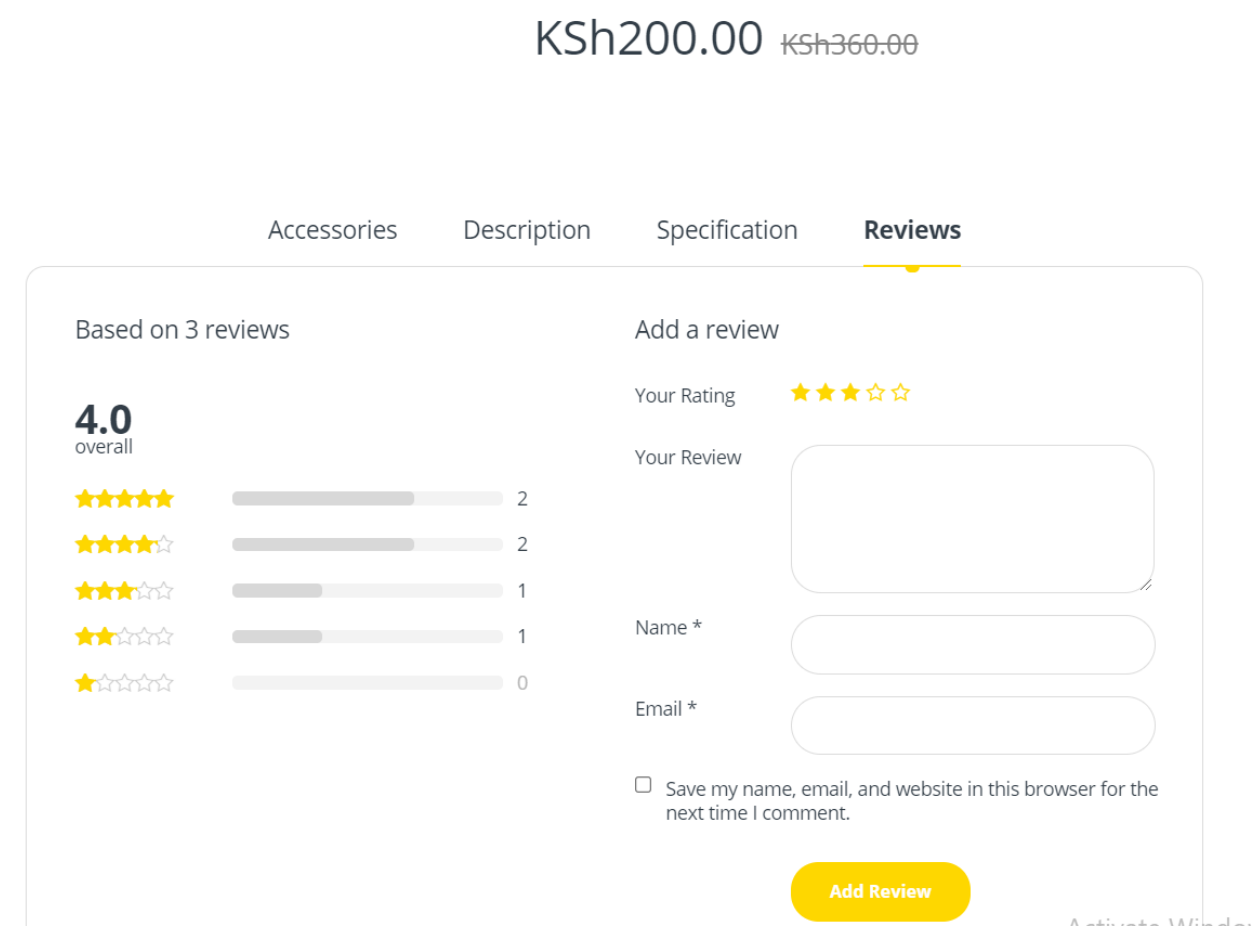


Figure 67 Prototype – user facing side: sample store item rating page

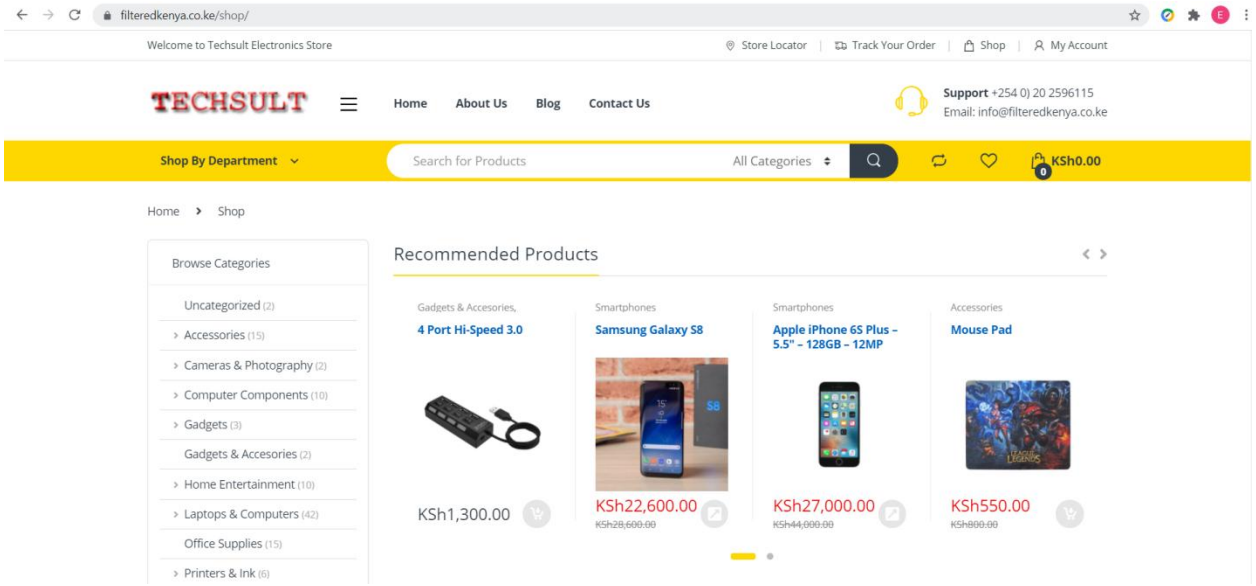


Figure 68 Prototype – user facing side: sample store recommended products output

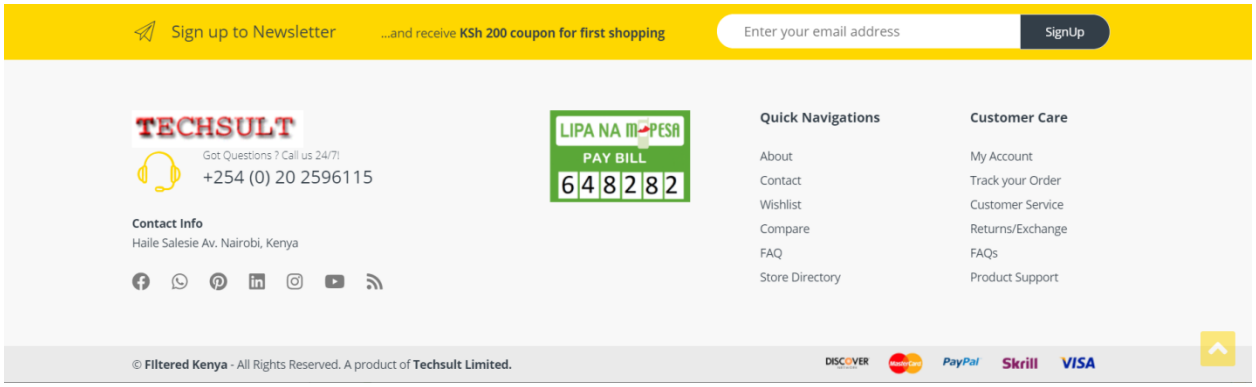


Figure 69 Prototype – user facing side: store payment options

4.6 Results on the assessment of the impact of the new trust parameter on the prediction accuracy and robustness properties of the collaborative recommendation algorithms.

4.6.1 Introduction

After constructing the trust parameter and deploying it into an empirical set up as described in sections 3.6 and 3.7, we ran comparative analysis tests on the behavior of the new algorithm as described in section 3.8.

In the next section, we present the behavior of the algorithm as far as robustness and the prediction accuracy are concerned.

4.6.2 Robustness

The goal of robust recommendation is to prevent attackers from manipulating the system through large-scale insertion of user profiles, a profile injection attack. (Burke, O'Mahony and Hurley, 2011).

We use prediction shift and hit ratio to evaluate robustness of the algorithm.

Table 15 Prediction Shift for product push attack on user based collaborative filtering algorithm

Attack size (%)	Bandwago	Average_T	Random_T	Bandwago	Average	Random
1	0.15	0.18	0.1	0.18	0.21	0.12
2	0.2	0.2	0.15	0.22	0.23	0.18
3	0.24	0.28	0.18	0.24	0.3	0.21
4	0.28	0.3	0.23	0.31	0.32	0.25
5	0.31	0.32	0.24	0.34	0.34	0.26
6	0.33	0.36	0.29	0.36	0.38	0.31
7	0.35	0.41	0.29	0.38	0.44	0.32
8	0.36	0.42	0.31	0.39	0.45	0.33
9	0.36	0.44	0.33	0.41	0.47	0.35
10	0.37	0.45	0.35	0.42	0.47	0.37
11	0.38	0.45	0.35	0.42	0.48	0.38
12	0.38	0.47	0.36	0.43	0.5	0.39
13	0.38	0.46	0.34	0.43	0.51	0.37
14	0.38	0.47	0.39	0.44	0.5	0.41
15	0.39	0.48	0.4	0.44	0.52	0.42
16	0.4	0.47	0.42	0.44	0.52	0.45
17	0.4	0.48	0.43	0.45	0.52	0.47
18	0.41	0.49	0.43	0.45	0.54	0.48
19	0.4	0.52	0.43	0.44	0.55	0.48
20	0.41	0.5	0.45	0.44	0.56	0.51

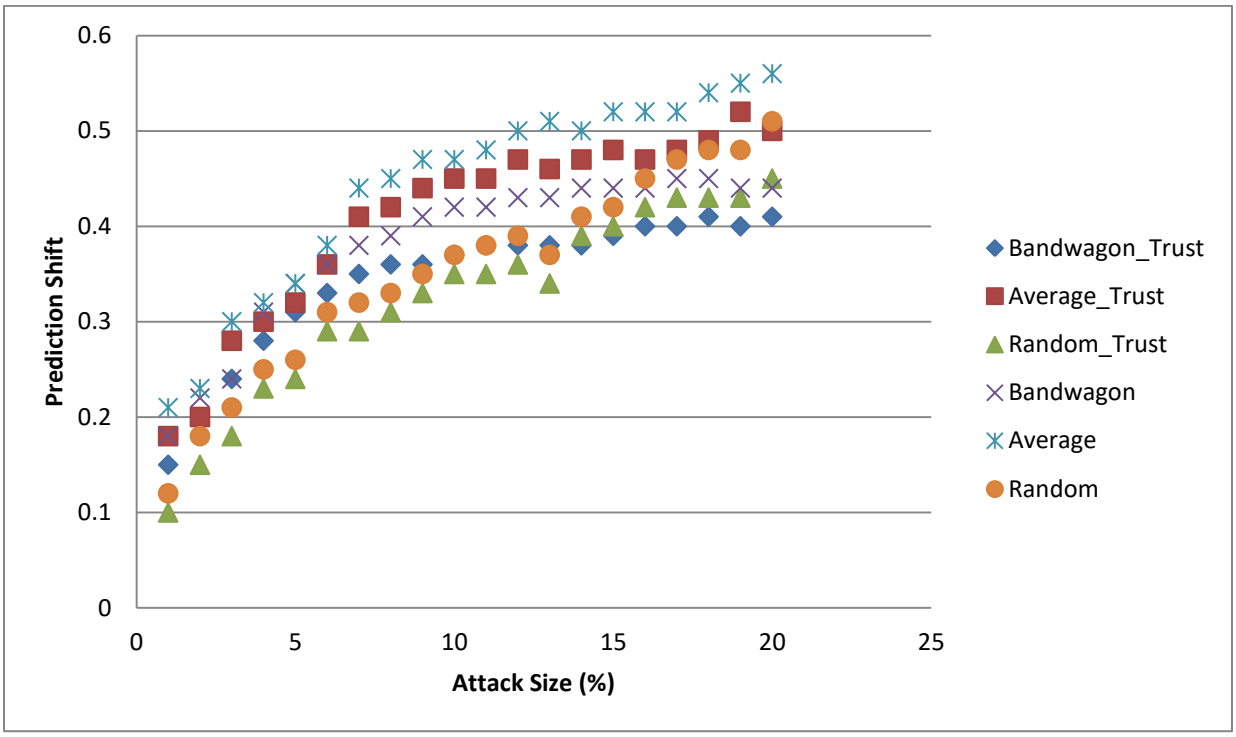


Figure 70 Prediction Shift for product push attack on user based collaborative filtering algorithm

Table 16 Hit Ratio for product push attack on user-based collaborative filtering algorithm

Number of Recommendati	Average_T	Random_T	Bandwagon_T	Baseline_T	Average	Random	Bandwago	Baseline
0	0	0	0	0	0	0	0	0
5	11	32	28	1	16	35	32	1
10	24	47	42	1	30	52	45	1
15	37	56	48	2	45	60	52	2
20	50	59	54	3	55	63	56	3
25	56	63	56	4	60	67	58	6
30	64	64	64	4	69	68	69	7
35	67	66	67	5	71	69	70	6
40	72	67	69	5	76	69	72	6
45	73	67	69	6	77	70	73	7
50	73	68	69	5	77	70	73	5

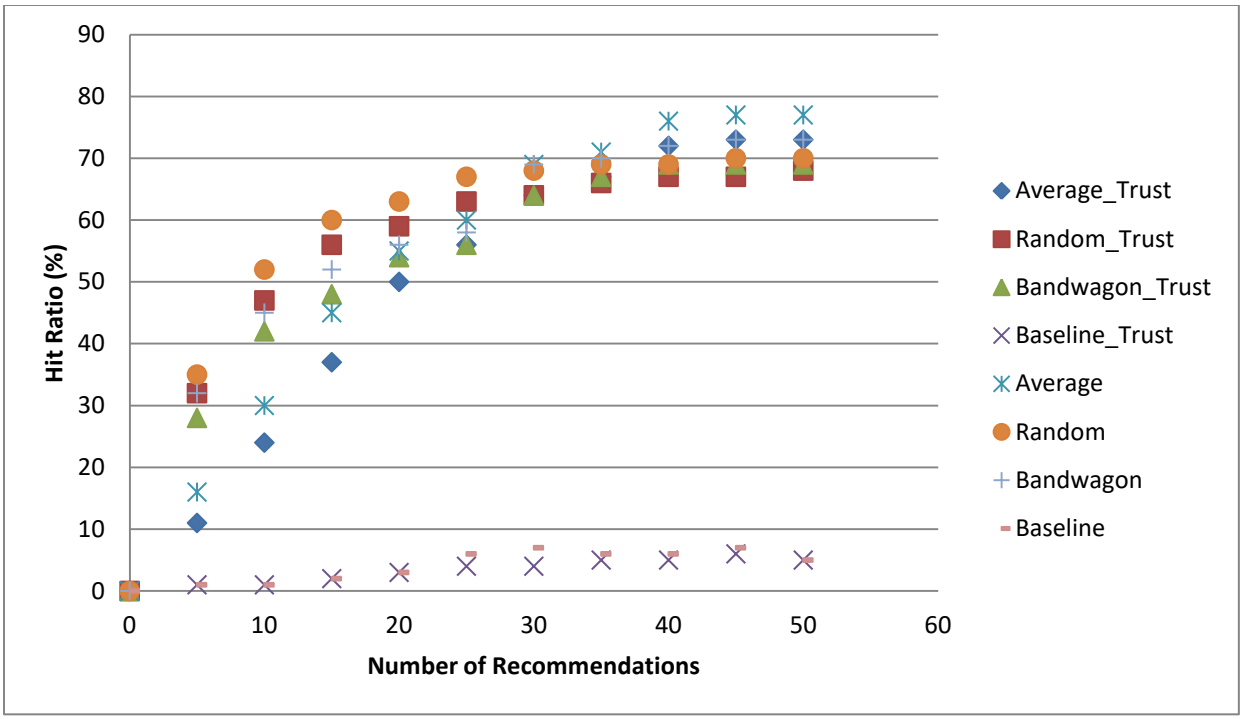


Figure 71 Hit Ratio for product push attack on user-based collaborative filtering algorithm

Table 17 Prediction Shift for product push attack on item-based collaborative filtering algorithm

Attack size	In Segmen	All User_T	In Segmen	All User
1	0.15	0.1	0.18	0.14
2	0.2	0.1	0.24	0.15
3	0.24	0.11	0.28	0.16
4	0.28	0.1	0.31	0.16
5	0.31	0.12	0.33	0.17
6	0.33	0.13	0.35	0.16
7	0.35	0.13	0.38	0.18
8	0.36	0.14	0.41	0.19
9	0.36	0.15	0.42	0.2
10	0.37	0.14	0.44	0.19
11	0.38	0.15	0.44	0.18
12	0.38	0.15	0.45	0.19
13	0.38	0.16	0.45	0.21
14	0.38	0.16	0.45	0.21
15	0.39	0.16	0.46	0.22
16	0.4	0.17	0.46	0.23
17	0.4	0.18	0.47	0.23
18	0.41	0.18	0.47	0.23
19	0.4	0.19	0.48	0.23
20	0.41	0.18	0.47	0.24

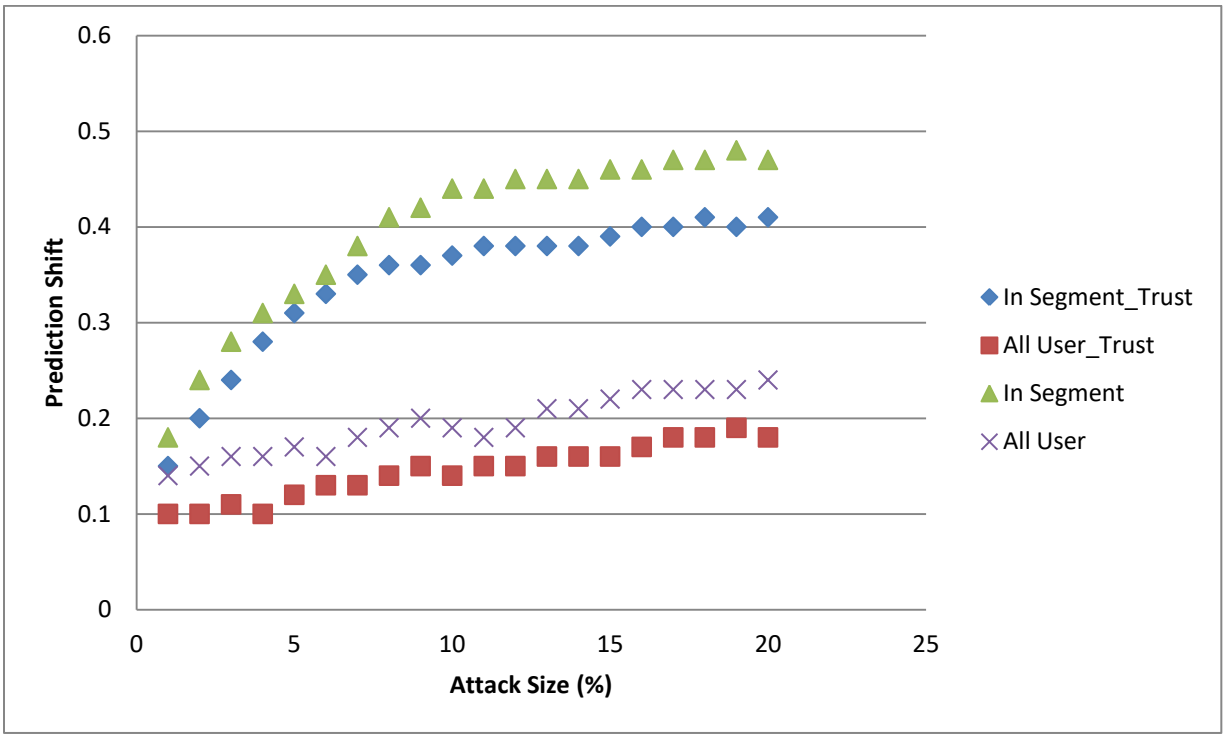


Figure 72 Prediction Shift for product push attack on item-based collaborative filtering algorithm.

Table 18 Hit Ratio for product push attack on item-based collaborative filtering algorithm

Number of Recommendations	In Segmen	All User_T	Base Line_	In Segmen	All User	Base Line
0	0	0	0	0	0	0
5	17	3	0	23	5	3
10	26	2	1	30	6	0
15	29	3	1	33	7	2
20	36	5	0	40	6	1
25	36	5	1	41	7	1
30	37	6	1	42	9	2
35	37	7	1	43	11	3
40	38	7	2	42	10	3
45	39	7	2	43	11	4
50	38	7	3	43	11	3

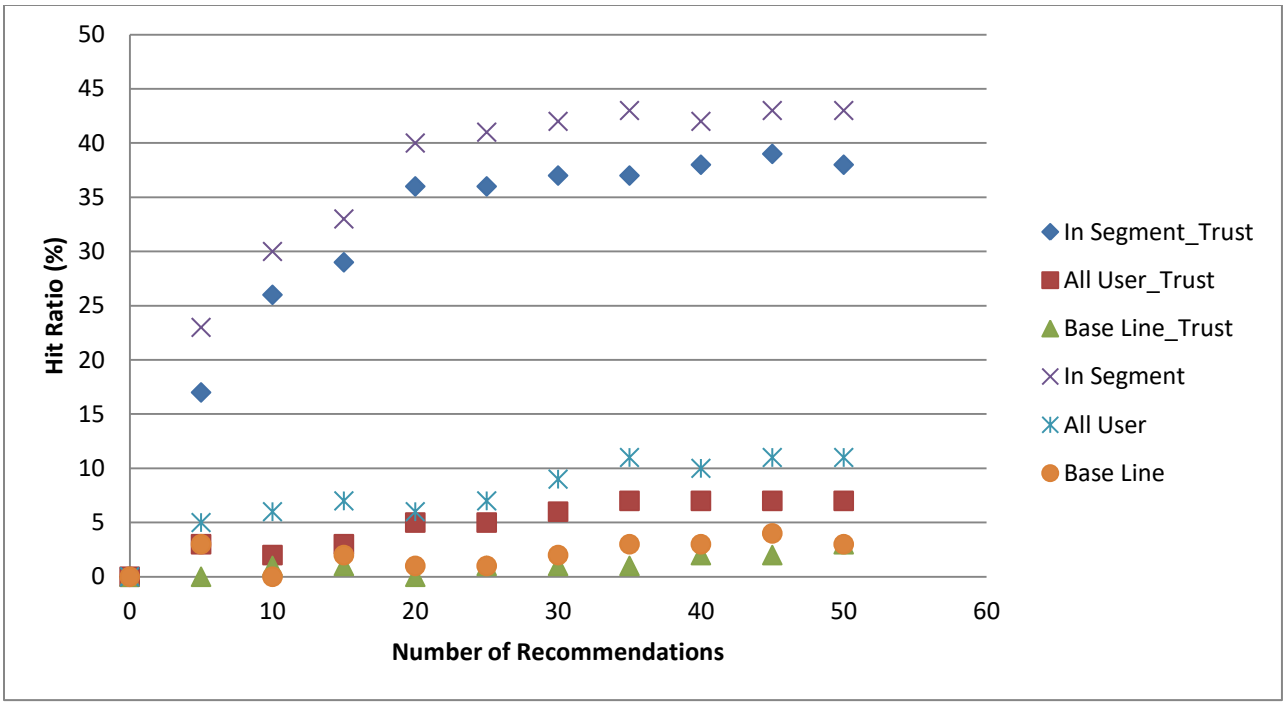


Figure 73 Hit Ratio for product push attack on item-based collaborative filtering algorithm

Table 19 Prediction shifts achieved by nuke attacks against the user-based algorithm

Attack size (%)	Average_T	Bandwago	Random_T	Love/Hate	Reverse ba	Average	Bandwago	Random	Love/Hate	Reverse bandwagon
1	-0.13	-0.11	-0.1	-0.1	-0.07	-0.18	-0.16	-0.34	-0.4	-0.1
2	-0.23	-0.22	-0.22	-0.42	-0.19	-0.26	-0.28	-0.71	-0.69	-0.61
3	-0.31	-0.34	-0.3	-0.23	-0.27	-0.37	-0.41	-0.78	-0.7	-0.7
4	-0.34	-0.44	-0.34	-0.67	-0.3	-0.41	-0.49	-0.82	-0.93	-0.73
5	-0.38	-0.51	-0.42	-0.7	-0.32	-0.43	-0.57	-0.9	-0.97	-0.74
6	-0.41	-0.56	-0.48	-0.73	-0.34	-0.46	-0.6	-0.88	-1.01	-0.78
7	-0.41	-0.61	-0.52	-0.75	-0.36	-0.47	-0.67	-0.91	-1.02	-0.81
8	-0.42	-0.63	-0.58	-0.78	-0.38	-0.47	-0.68	-0.92	-1.01	-0.82
9	-0.42	-0.62	-0.6	-0.8	-0.39	-0.48	-0.67	-0.94	-1.02	-0.86
10	-0.43	-0.63	-0.61	-0.81	-0.4	-0.49	-0.69	-0.93	-1.02	-0.87
11	-0.43	-0.64	-0.6	-0.8	-0.4	-0.49	-0.69	-0.95	-1.03	-0.86
12	-0.45	-0.65	-0.61	-0.82	-0.41	-0.48	-0.71	-0.92	-1.04	-0.87
13	-0.46	-0.66	-0.62	-0.85	-0.41	-0.49	-0.7	-0.94	-1.05	-0.88
14	-0.47	-0.67	-0.65	-0.83	-0.41	-0.5	-0.7	-0.95	-1.05	-0.91
15	-0.47	-0.67	-0.64	-0.84	-0.42	-0.5	-0.71	-0.97	-1.06	-0.82
16	-0.47	-0.67	-0.7	-0.86	-0.42	-0.52	-0.72	-0.97	-1.06	-0.87
17	-0.47	-0.68	-0.72	-0.84	-0.43	-0.53	-0.73	-0.96	-1.06	-0.9
18	-0.48	-0.7	-0.7	-0.85	-0.42	-0.53	-0.73	-0.98	-1.06	-0.88
19	-0.48	-0.68	-0.7	-0.83	-0.42	-0.54	-0.72	-0.97	-1.06	-0.87
20	-0.48	-0.67	-0.69	-0.84	-0.42	-0.53	-0.72	-0.98	-1.06	-0.87

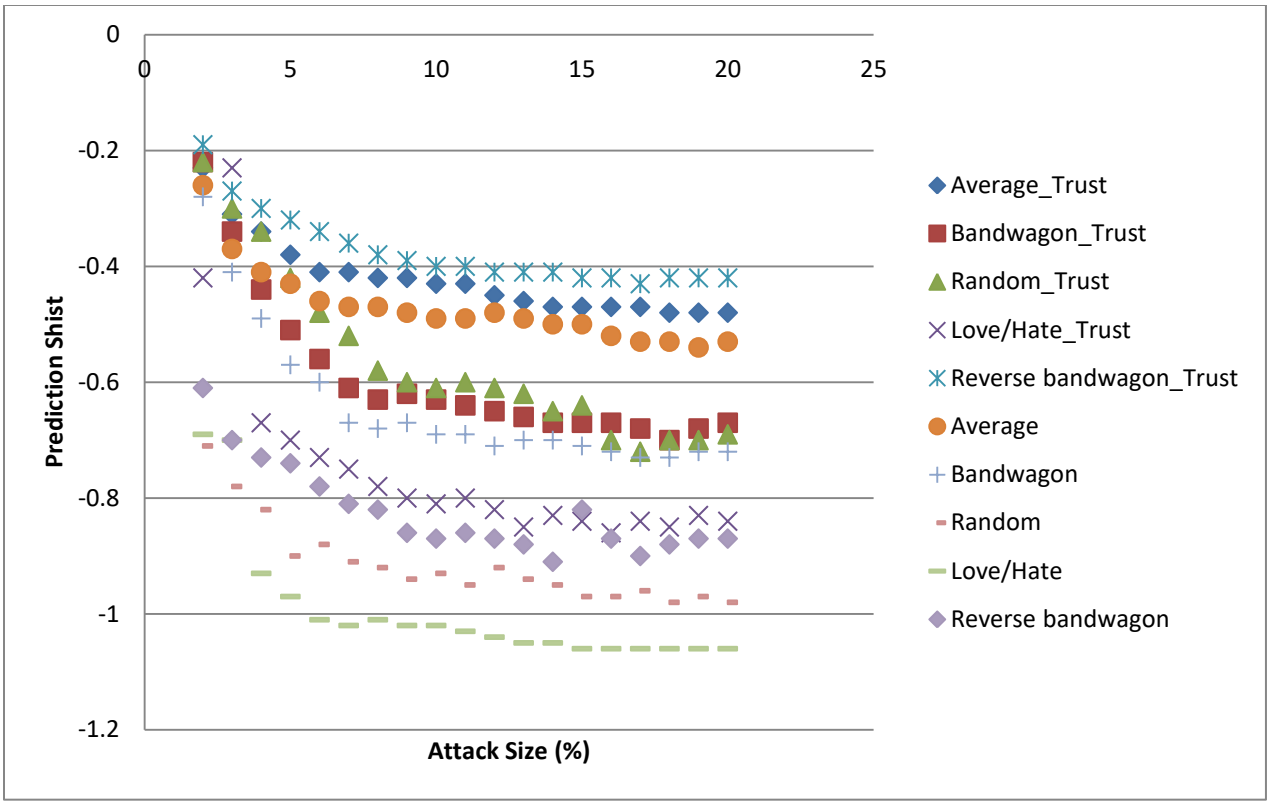


Figure 74 Prediction shifts achieved by nuke attacks against the user-based algorithm

Table 20 Prediction shifts achieved by nuke attacks against the item-based algorithm

Attack size (%)	Average_T	Bandwago	Random_T	Love/Hate	Reverse ba	Average	Bandwago	Random	Love/Hate	Reverse bandwagon
1	-0.14	0	0	0	-0.05	-0.38	0	0	0	-0.1
2	-0.24	0.01	0.01	0	-0.08	-0.42	0.03	0.02	0.01	-0.19
3	-0.37	0.02	0.03	0.02	-0.16	-0.5	0.05	0.05	0.06	-0.27
4	-0.42	0.01	0.04	0.01	-0.2	-0.58	0.04	0.7	0.08	-0.31
5	-0.44	0.03	0.06	0.04	-0.21	-0.66	0.07	0.09	0.09	-0.34
6	-0.46	0.01	0.05	0.06	-0.24	-0.7	0.06	0.1	0.08	-0.36
7	-0.49	0.019	0.06	0.07	-0.26	-0.72	0.08	0.11	0.09	-0.38
8	-0.55	0.02	0.06	0.08	-0.27	-0.76	0.09	0.12	0.1	-0.39
9	-0.58	0.021	0.07	0.07	-0.28	-0.79	0.1	0.12	0.12	-0.4
10	-0.59	0.022	0.05	0.06	-0.29	-0.81	0.09	0.13	0.12	-0.41
11	-0.61	0.023	0.06	0.03	-0.3	-0.83	0.11	0.13	0.13	-0.42
12	-0.62	0.024	0.04	0.06	-0.31	-0.84	0.13	0.13	0.1	-0.43
13	-0.63	0.025	0.07	0.05	-0.32	-0.85	0.13	0.14	0.12	-0.43
14	-0.62	0.026	0.03	0.07	-0.31	-0.86	0.13	0.14	0.11	-0.44
15	-0.63	0.027	0.06	0.03	-0.32	-0.86	0.12	0.12	0.13	-0.45
16	-0.64	0.028	0.05	0.07	-0.33	-0.87	0.13	0.14	0.13	-0.46
17	-0.63	0.029	0.07	0.08	-0.32	-0.88	0.11	0.13	0.12	-0.46
18	-0.64	0.03	0.07	0.08	-0.33	-0.87	0.13	0.13	0.13	-0.46
19	-0.63	0.031	0.08	0.07	-0.33	-0.89	0.14	0.14	0.13	-0.46
20	-0.63	0.032	0.08	0.06	-0.32	-0.87	0.13	0.13	0.13	-0.46

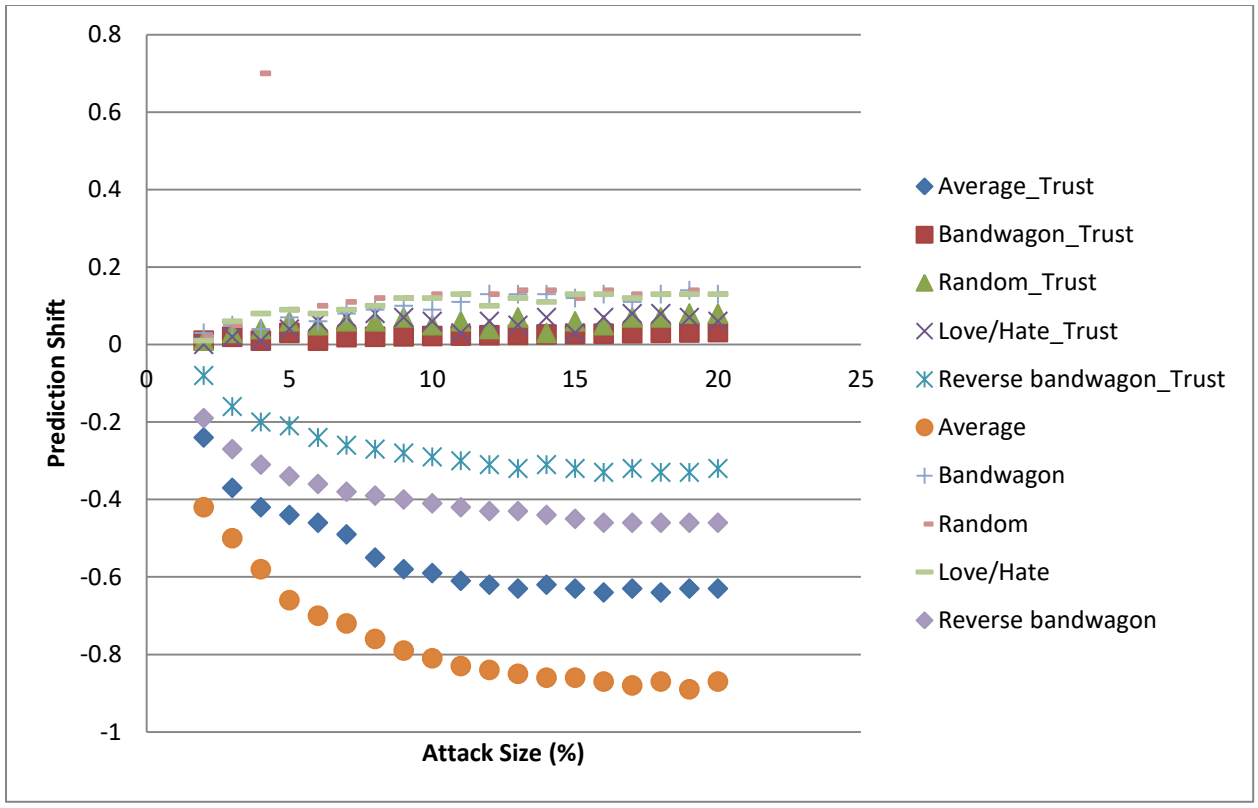


Figure 75 Prediction shifts achieved by nuke attacks against the item-based algorithm

Table 21 Hit ratios achieved by the popular, probe and average nuke attacks against the user-based algorithm.

Number of Recommendations	Popular_T	Probe_Tru	Average_T	Popular	Probe	Average
0	0	0	0	0	0	0
5	22	15	1	24	20	2
10	29	21	1.5	28	24	2.1
15	37	26	1.8	42	30	2.6
20	41	29	1.9	44	34	3.1
25	43	31	2.2	46	36	3.3
30	44	32	2.6	47	35	3.4
35	44	30	2.7	47	36	3.5
40	44	33	2.9	48	36	3.4
45	45	34	3.2	47	37	3.5
50	44	33	3	48	36	3.5

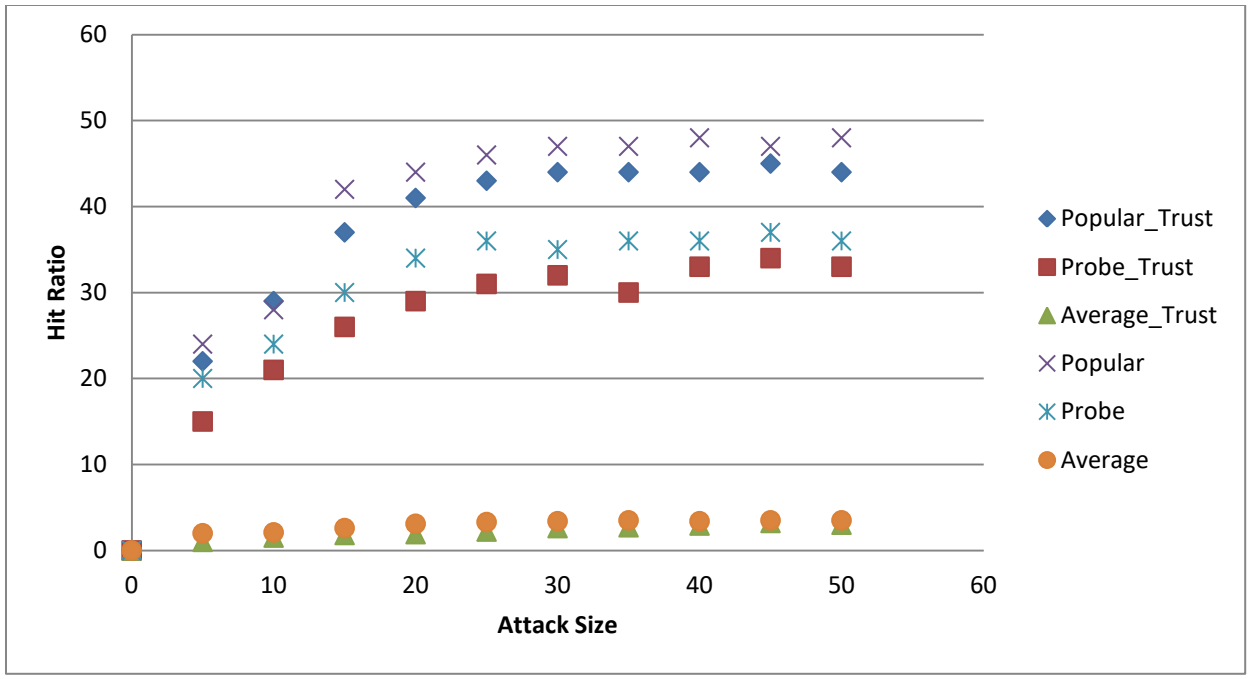


Figure 76 Hit ratios achieved by the popular, probe and average nuke attacks against the user-based algorithm.

4.6.2 Prediction accuracy

Table 22 Prototype Empirical Results – Root Mean Square Algorithm Accuracy

Neighbors	5	10	15	20	25	30
CFRA	0.9721	0.96464	0.9615	0.9604	0.96	0.9595
CFRAT	0.9683	0.9621	0.9581	0.9567	0.956	0.9559

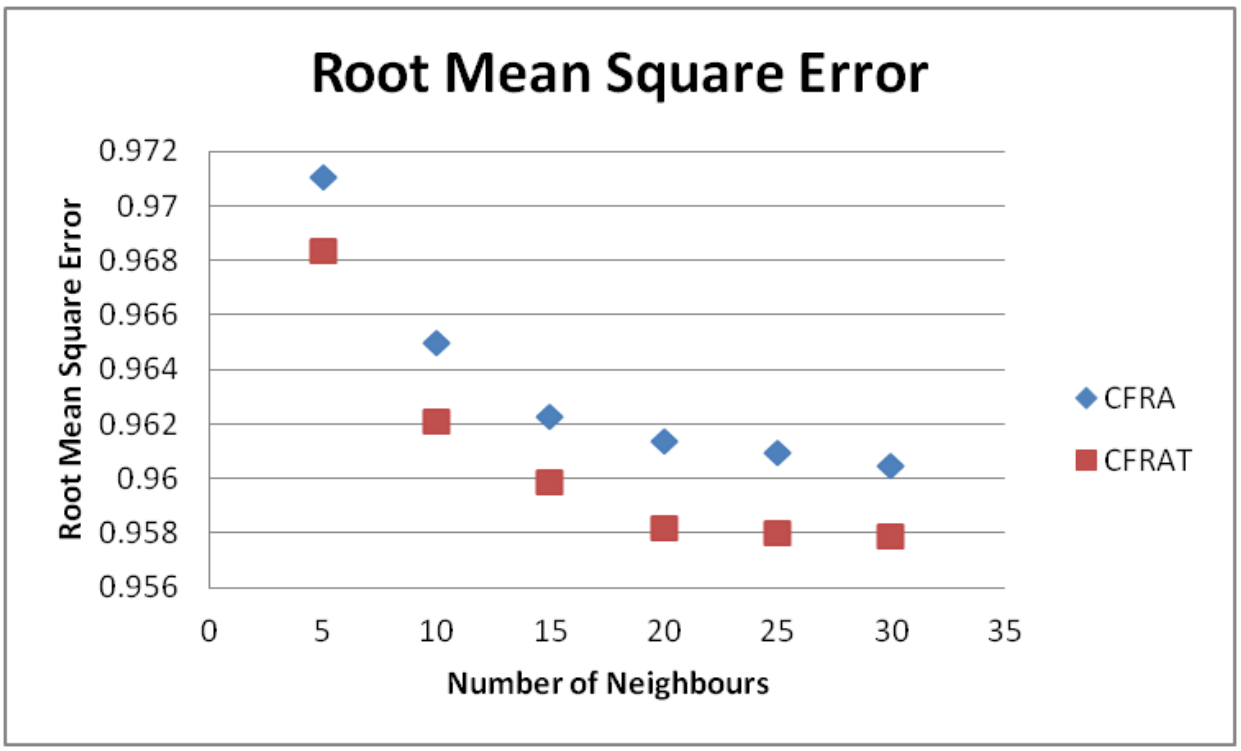


Figure 77 Prototype Empirical Results – Root Mean Square Algorithm Accuracy

Table 23 Prototype Empirical Results MAE Curve

Neighbors	5	10	15	20	25	30
CFRA	0.9711	0.965	0.9623	0.9614	0.961	0.9605
CFRAT	0.9684	0.9621	0.9599	0.9582	0.958	0.9579

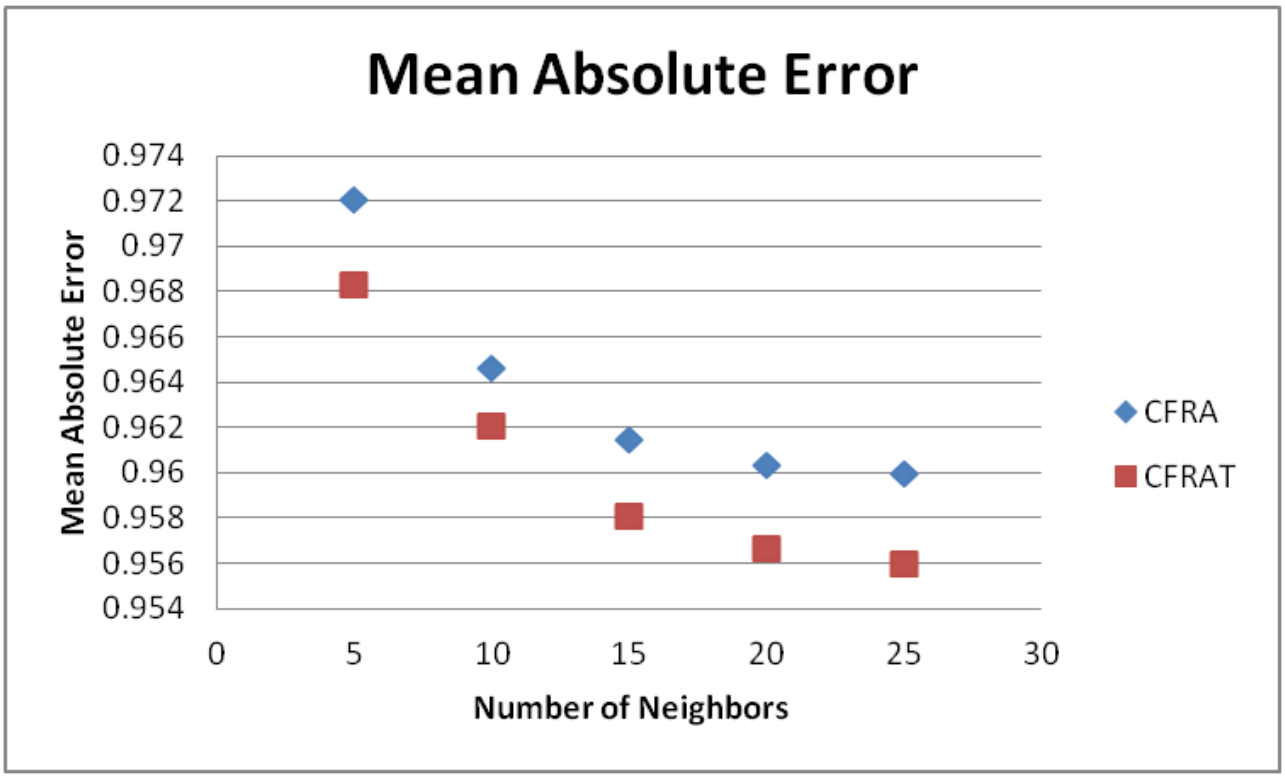


Figure 78 Prototype Empirical Results MAE Curve

4.7 Hypothesis Testing Results

The results in this section can be interpreted by looking at (Dorfman, 2019).

4.7.2 Robustness Hypothesis Testing Decision Making

Table 24 Robustness Hypothesis Testing Results

S/N	Description	Sub Hypothesis	P - value (σ)	T-stat	t-critical one tail	Number of observations (n)	Remark	Reject Null?
1	Prediction Shift for product push attack on user based collaborative filtering algorithm	Average	5.91139E-11	12.56767938	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Bandwagon	9.44649E-11	12.22601834	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Random	4.94171E-10	11.07731306	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
2	Hit Ratio for product push attack on user-based collaborative filtering algorithm	Average	3.73515E-07	12.04270923	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES

		Bandwagon	7.29474E-07	11.1291124	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Random	7.29474E-07	11.1291124	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Baseline	0.018393749	2.449489743	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
3	Prediction Shift for product push attack on item-based collaborative filtering algorithm.	All Users	3.12212E-16	24.78574859	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		In Segment	6.38908E-11	12.51052343	1.729132792	20	The T-value is highly significance compared to critical value of and also p value	YES

							is negligible.	
4	Hit Ratio for product push attack on item-based collaborative filtering algorithm	All User	4.64392E-06	8.907784453	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		In Segment	1.11802E-08	18.05320007	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Baseline	0.011449747	2.738612788	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
5	Prediction shifts achieved by nuke attacks against the user-based algorithm	Average	2.08517E-13	-17.34469513	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Bandwagon	6.09932E-15	-21.08166868	1.729132792	20	The T-value is highly significance	YES

							compared to critical value of and also p value is negligible.	
		Random	4.05566E-14	-18.99524052	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Love/Hate	6.33388E-14	-18.53242445	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Reverse Band Wagon	1.80508E-14	-19.86250197	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
6	Prediction shifts achieved by nuke attacks against the item-based algorithm	Average	2.98625E-18	-31.83765479		20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Bandwagon	4.49569E-09	9.671342365	1.729132792	20	The T-value is highly	YES

							significance compared to critical value of and also p value is negligible.	
		Random	0.005577841	2.810891842	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Love/Hate	7.08453E-08	8.090729559	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Reverse Band Wagon	2.39677E-17	-28.46979978	1.729132792	20	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
7	Hit ratios achieved by the popular, probe and average push attacks against the user-based algorithm.	Popular	0.00019816	5.467934261	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES

		Probe	8.33625E-07	10.95445115	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
		Average	7.78135E-06	8.358885556	1.833112923	10	The T-value is highly significance compared to critical value of and also p value is negligible.	YES

4.7.3 Prediction Accuracy Hypothesis Testing Decision Making

Table 25 Prediction Accuracy Hypothesis Testing Results

S/N	Description	Sub hypothesis	P - value (σ)	T - stat	t-critical one tail	Number of observations (n)	Remark	Reject Null?
1	Measuring prediction accuracy through Mean Absolute Error	Mean Absolute Error	7.01464E-06	16.70894067	2.015048372	6	The T-value is highly significance compared to critical value of and also p value is negligible.	YES
2	Measuring prediction accuracy through Root Mean Square Error	Root Mean Square Error	1.25467E-06	23.66431913	2.015048372	6	The T-value is highly significance compared to critical value of and also p value	YES

							is negligible.	

CHAPTER 5: DISCUSSION

5.1 Introduction

In this section, we intend to describe the significance of our results, in terms of who cares about what, how and why as far as our results are concerned.

Again for this section, we have categorized the discussion in terms of the research project objectives.

5.2 Determining the indicators of trust in online services

For this deliverable, we used exploratory factor analysis, which traditionally has been used to explore the possible underlying structure of a set of measured variables without imposing any preconceived structure on the outcome as well as the principal component analysis which reduces the number of observed variables to a smaller number of principal components which account for most of the variance of the observed variables,(Suhr, 2009).

In this section of our research project, we had an objective of determining the indicators of trust in an online service. As described by, trust is an important parameter which improves collaboration; lowers cost of transactions in a business and also ensures harmony in an association. Participants in any exercise always look for indicators of trust of trust and until they are convinced that there is trust is when they commit to participate, these include online shoppers who are seeking to purchase an item online.

It is therefore paramount for the online service platform developers and online shop owners to be privy of what the end user will look at when evaluating his shop online if he wants to be competitive against millions of other shops online.

This is because in the case of online shopping, the seller is not there to instantly clarify the concerns that a customer has like the way it is in a physical shop. A small misconception from the customer is enough to make the shop owner miss a sale worth millions and this works against the core objective of owning a shop online, which is to improve sales by reaching customers who would not otherwise reach a physical shop.

From the opposite side, also naïve shoppers can know how to check a trustworthy shop by relying on what has been provided as the key indicators and this will improve their shopping

experience because they will not need to go through a lot of pain in the process of thinking about which shop to purchase from and neither will their end decision be exposed to potential fraud which may lead to lose of livelihood or to a far extent, lose of life.

Assurance that they have a means to help them choose a proper shop online will also improve their trust in online shops. This will suppress the urge to go and shop physically when they fear risking.

Going to shop out physically not only adds additional locomotive costs to the budget but also there is an element of wasting valuable time in the process.

Saving the locomotive energy spent on physical shopping is not just a green energy way of doing things but also on aggregate leads to economic benefits at the national level.

In some cases, buying items physically also has its security challenges such as exposure to repudiation of a transaction or deliberate conman ship from the physical shop owners and in many cases the physical shop owners might have made a deliberate effort to obscure the evidence of the transaction, unlike in an online shopping whereby by the time the shopper makes a transaction payments, they leave behind are so many digital footprints of the said transaction and therefore making repudiation process either inefficient resource wise to the seller, or simply impossible, so any parameter that helps to contribute to online shopping is a parameter of value.

The governments will also benefit when the population is enlightened on how to shop online since shopping online provides business data both for tax collection more easily than when people are transacting physically and maybe recordings transactions on paper books or simply not recording the business transactions. This also applies to cases of law enforcements such as tracking of counterfeits and also tracking of contrabands.

In figure 5, we have established that that majority (38.6%) spend only between KES 1 and KES 10000. This is paltry as compared to the physical shopping as per personal observation considering how people shop in physical outlets. We then thought that maybe it is due to insufficient sensitization or lack of awareness as to the benefits of online shopping, but again this line of thinking was proved wrong by the findings shown in figure 6.

In figure 6, it is evident that the shoppers understand the benefits of shopping online and especially the key ones, which include time saving (43.2%), ability to track a purchases (35.7%) as well as relatively cheaper prices (26.1%). The looming question then becomes why then don't they shop online?

Results in figure 7 show that most of the shoppers have concerns related to online security and mistrust of online shops. In figure 6, we realize that 40.1% fear being deceived when shopping online, 22.8% perceive online shopping as unreliable, 19.5% thinks that it is insecure to shop online and 12.9% believe that shopping online leads to a breach of their privacy.

The key factors in the findings shown in figure 7 can be summarized as lack of trust in online shopping and this section gives impetus to the need to find a solution and confirms the relevance or significance of our study.

The four key factors in figure 7 are latent variables, in the sense that they cannot be measured directly. Figures 8,9,10,11 provide the empirical indicators for the said latent variables in figure 7.

We have also presented the indicators that indicate trust in ecommerce platforms from consumer perspective in developing country context in table 5.

We used EFA and PCA tests as the key statistical tests.

For Exploratory Factor Analysis, we obtained uniqueness, factor loadings, scree plot, eigen values, parallel analysis, optimal coordinates, acceleration factor.

For Principal Component Analysis, we obtained PCA Importance of Components, Loadings, Scree Plot and the distance biplot.

In EFA, we were keen to find underlying factors that contribute to certain observed variables. In turn, these variables can be used to predict the presence of such factors in a reversed manner. The variables are therefore called indicators of the underlying factor and what this means is that varying the indicator contributes to some variation or variance in the underlying factor which it indicates. Indeed the more variance in the underlying factor is observed by a variation in the

surface variable/indicator, the more significant that indicator is. As a result, the total variance of an underlying factor can be represented by the equation 12 below:

$$Total\ Variance = Common\ Variance + Unique\ Variance + erro \qquad (12)$$

In this equation, common variance is the variance accounted for by many indicators combined. The Unique variance is the variance which is unique to a particular indicator where as error term is the term associated by some error not necessarily as a result of actual relationship between the underlying factor and its indicator, such as error in measurements. In our results, this unique variance is represented by Figures 13 and 20. As can be seen in figure 13, the uniqueness of the indicator "abnormal pricing" is indeed 0.00, and is very much close to that of use of misleading tactics, which indicates that indeed the two are very high correlated and an indicator that one of them is not as significant when estimating trustworthiness of an ecommerce platform, an indicator that one might need to be done away with.

We also have factor loadings for EFA in figures 14 and 21. As can be seen in figure 7, there is still a problem of cross loading, which does not occur in figure 21.

The next parameter we report under EFA is the scree test in figures 12 and 19. These indicate the eigen values, parallel analysis, optimal coordinates, acceleration factor. As can be seen in figure 12, there is really no clear point of inflection in the plot as can be seen clearly in figure 13 that the point of inflection is indeed at four factors. This is consistent with literature. The eigen values, parallel analysis, optimal coordinates and the acceleration factor are however the same in both figures, which paints some element of greyness and necessitates further research as some analysts may argue that both figures are meaningful. We however go with the results of figure 19 because indeed it agrees with literature and also with the opinion of the thematic review panel, both of which suggest that four factors are adequate for this data. Scree test helps a researcher to estimate the number of factors to retain an exploratory factor analysis, however due to the subjectiveness about the actual or acceptable point of inflection, (sometimes the graph has multiple points of inflection!), non – graphical solutions to scree test have been suggested and these include the eigen values, parallel analysis, optimal coordinates and the acceleration factor. We do not discuss these in this paper because they are all consistent and their meanings can be inferred from literature, (Ledesma, Valero-Mora and Macbeth, 2015), (Ruscio and Roche, 2012).

About Principal component analysis, even though most studies choose to either carry out either PCA or EFA because the two scientific/statistical procedures usually talk to answer different research questions, the two procedures are not mutually exclusive in the sense that a researcher may want to excavate underlying factors which are contributing to the values of some set of observed variable, which is a structural questions while still maintaining the desire to exercise dimensionality reduction, which is a measurement question, on the observed variables for certain reasons such as to reduce the length of a questionnaire or a need which is related to resource constraint whatsoever, then the researcher will perform principal component analysis order to determine the most important variables/components to retain. As such we sought to carry out both procedures and report both the results in one go.

For the principal components analysis, we obtained PCA Importance of Components, Loadings, Scree Plot and the distance biplot.

Figures 15 and 22 shows the PCA Importance of components or Communality (variance accounted for), with abnormal pricing and without abnormal pricing respectively. As can be seen in figure 15, the variance accounted for by component 19 is indeed negligible as compared to the rest of the components. This means that component 19 is not contributing meaningful amount of variance and therefore dropping it from the list of variables does not result in losing a meaningful amount of information present in the original data. Figure 22 looks better in terms of how each of the 18 components contributes relatively a balanced amount of variance and therefore all are worth retaining.

Another indicator of weight of component, in absolute value is the PCA loadings, shown in figures 16 and 23. PCA loadings are equivalent to correlations between observed variables and components. As can be seen in figure 16, component 19 has only two items loading onto it and these are abnormal pricing and use of misleading tactics. In Principal Component Analysis, negative loading implies a negative correlation. It can be seen that component 19 is problematic since only two variables load onto it, which are the use of misleading tactics and the use of abnormal pricing. From the thematic expert review panel, these two items are supposed to be scored in the same direction since in their view, use of abnormal pricing to tease buyers is indeed a manifestation or an instance of or a case of use of misleading tactics. We therefore dropped the

use of abnormal pricing from the list of variables and after running the analysis again, figure 23 gives a better output where all items have meaningful loadings to all the remaining components.

The graphical PCA scree plots in figures 17 and 24 also agree with the EFA scree plots in figures 12 and 19, that four components can adequately represent the information contained in the original data. For the same reason as that already discussed for the EFA scree plot, again it is not very clear where the graph screes at for figure 17 but it is clear at figure 24 that it screes at four components.

Another parameter reported here in is the distance biplot shown in figures 17 and 25. This is a type of scatter plot which graphically represents the position of each variable score on a two dimensional axis of the first two principal components. Each score is represented by a vector representing the direction and the magnitude of effect that a variable has on the final estimation. The visual biplot is a tool which can be used to quickly get a glance of the most important variable that contributes to a certain direction, just by looking at the variable with the longest vector whose direction is towards the desired direction.

We used cronbach's alpha, shown in figure 26, to ascertain the reliability of the data used in this study. The cronbach's alpha cut off for reliable from the literature is 0.7. Figure 26 shows a cronbach's alpha value of 0.959 which confirms that we used reliable data in the study.

We therefore present the residual elements of trust in table 5, which describes the trust constructs, their indicators, when to measure the said indicator and also how to measure the indicator.

5.3 Developing a model for estimation of trustworthiness of an online shop

We have also presented models for estimating trust in ecommerce platform. We used Structural Equation Modeling (SEM) (Hox and Bechger, 2014),(Stein, Morris and Nock, 2012).

We perform Confirmatory Factor Analysis test as an ongoing work from the Exploratory Factor Analysis (EFA) study done earlier, partly published in (Ngwawe, Abade and Mburu, 2020) and resonating with (Roman, 2007) in terms of methodology, save for the context.

Here we produce path diagrams for models as follows:

i. One factor trust model, figure 50

ii. Two factor trust model, figure 46

iii. Three factor trust model, figure 42

iv. Four Factor trust model, figure 38

v. Four factor with a second order factor trust model, figure 54

We produce path diagrams for different number of factors because during EFA, there was some grey area in determining the correct number of factors to consider as can be seen on the scree test, where from the graphical solution was not agreeing with non graphical solutions to scree test in the sense that the point of inflection in the graph was at four factors where as the non graphical solutions such as parallel analysis, optimal coordinates, acceleration factor suggested two, two and one respectively. As a result, during this stage, we test all of the four possible cases and therefore we have here the path diagrams in figures 38, 42, 46, 50, 54 as shown above.

We then use statistics presented in table 6, in reference to the cutoffs suggested in (Kline, 2005) and determine that four is the number of factors to go with.

About the reliability of data used in the study, figure 55 shows the output of cronbach's alpha test for data reliability which is 0.956 and this is above the suggested cut of 0.7 for reliable data as summarized in table 7. We also present in the tables 8 and 9 the convergent reliability and divergent reliability and demonstrate how they pass the minimum requirements (Carlson and Herdman, 2012), (Zait and Bertea, 2012).

5.4 Augmenting the new trust model as a new parameter, called trust adjustment factor, into the classical collaborative recommendation algorithm to create the new trust enhanced algorithm.

The term "augment" is an English word which means to make greater, more numerous, larger, or more intense. In this context, we use it to mean adding a new parameter to the classical algorithm to make it more effective.

We describe the process of augmenting the new trust model into the classical collaborative filtering recommendation algorithm in section 3.6.5 and the results in section 4.4, in a self explanatory manner.

5.5 Deploying the new algorithm into an empirical setup for proof of concept

The output of this objective was an engineering design on how to deploy the new trust enhanced algorithm for production purpose.

In order to prove our concept and demonstrate that the trust model we have constructed above is not just pure logic but can stand the test of waters in a practical world, we designed an prototypical empirical setup for the trust enhanced algorithms and observed the performance results which we analyzed and then we performed comparative analysis by benchmarking these results against the results of the classical algorithm in the same context in order to prove our concept.

The Engineering design work described in section 3.7 and the screenshots presented figures 65 to 69 in section 4.5 are self explanatory.

5.6 Assessing the impact of the new trust parameter on collaborative recommendation on properties of the recommender algorithm.

This section discusses the effect of the new trust parameter on the effectiveness of the artificial intelligence driven common filtering recommender algorithm performance in terms of it prediction accuracy and also robustness properties.

The procedure of evaluation has been presented in section 3.8 and how to test the hypothesis described in section 3.9. The results of the two sections are presented in section 4.6 and 4.7.

We realize that the new trust parameter improves the robustness of both user based recommender algorithm as well as item based recommender algorithm as measured by both the prediction shift and also hit ratio for all modes of profile injection attacks as shown by the tables 15 - 21 and figures 70-76 in section 4.6.2.

We also realize that the new trust parameter improves the prediction accuracy, as measured by both Mean Absolute Error (MAE) and also by Root Mean Square Error (RMSE) as shown by the figures 77 and 78 and tables 22 and 23 in section 4.6.1.

Tables 24 and 25 also confirm that these impacts are not out of mere chance but have a statistical significance as measured by the t-test.

CHAPTER 6: CONCLUSION

6.1 Introduction

In this work, we identified a problem as stated in section 1.1, that the traditional collaborative recommendation algorithm is susceptible to manipulation using profile attacks .

We then posed five research questions and sought to seek an answer through research.

The research questions were:

6) What are the indicators of trustworthiness in online shop from the perspective of a Kenyan online shopper?
7) How can we estimate the trustworthiness of an online shop beforehand?
8) How can incorporate the estimated trust parameter into existing collaborative recommendation algorithm to make it more robust and accurate?
9) How can we deploy the new trust enhanced algorithm into an empirical set up for production purposes?
10) What is the impact of the new trust parameter on the robustness and prediction accuracy properties of collaborative recommendation algorithm?

We therefore wish to conclude the research project by providing the answers to the research questions.

6.2 What are the indicators of trustworthiness in online shop from the perspective of a Kenyan online shopper?

To respond to the question of the indicators of trustworthiness of an ecommerce platform, we used Exploratory Factor Analysis as described in section 3.4 - to determine the indicators of trust in online services.

We report the results in table 5.

6.3 How can we estimate the trustworthiness of an online shop beforehand?

To respond to the question of how to estimate the trustworthiness of an ecommerce platform, we created a model as described in section 3.5, results reported in section 4.3 and discussed in section 5.3.

6.4 How can incorporate the estimated trust parameter into existing collaborative recommendation algorithm to make it more robust?

The procedure to incorporate the estimated trust parameter in to an artificial intelligence driven collaborative filtering recommender system pipeline was engineered following the work that already existed in literature and is presented in section 3.6 - Augmenting the trust model as a new parameter, called trust adjustment factor, into the classical collaborative recommendation algorithm to create a new trust enhanced collaborative recommendation algorithm.

6.5 How can we deploy the new trust enhanced algorithm into an empirical set up for production purposes?

To respond to the question of how to deploy the new trust enhanced algorithm into an empirical setup for production purposes, we engineer a system based on knowledge on literature as presented in section 3.7 and provide screenshots in section 4.4.

6.6 What is the impact of the new trust parameter on other properties of collaborative recommendation algorithm?

To assess the impact of the new trust parameter on the effectiveness of a collaborative recommender algorithm, we run a comparative analysis between the new trust enhanced collaborative filtering recommendation algorithm against the classical collaborative filtering recommendation algorithm in terms of their prediction accuracy and robustness properties as described in section 3.8. and results reported in section 4.6.

We then test the hypothesis using t-test as described in the procedure in section 3.9 and results reported in section 4.7, to ascertain that the positive results in section 4.6 are not just out of mere chance but are of statistical significance.

We therefore reject the null hypotheses that autonomous trust model has no significant effect on the robustness of a collaborative recommendation algorithm and also reject the null hypothesis that autonomous trust model has no significant positive effect on prediction accuracy of a collaborative recommendation algorithm.

Our research contribution therefore is the Context Aware Computational Trust Model for Robust and Accurate Recommender Systems Algorithms for Ecommerce Platforms as well as the technique to embed it into the artificial intelligence or data driven recommender system autonomously.

CHAPTER 7: FUTURE RESEARCH DIRECTION

An algorithm is a mechanism of doing things in a finite way. It is therefore considerable that changing the way things work will definitely shift the output of the process by some margins. In order to help the application developers to make a decision on how to use our new algorithm, we intend to carry out further scientific studies on the impact of trust parameter on other properties of collaborative recommender algorithms, other than robustness and prediction accuracy as listed in section 2.1.4 and described in the work of (Shani and Gunawardana, 2011).

References

Aggarwal, G., S, M., Pál, D. and Pál, M. (2009) 'General auction mechanism for search advertising', WWW '09: Proceedings of the 18th international conference on World wide web, 241-250.

Akaike, H. (1987) 'Factor Analysis and AIC', in Parzen, E., Tanabe, K. and G., K. (ed.) *Selected Papers of Hirotugu Akaike. Springer Series in Statistics (Perspectives in Statistics).*, New York, NY.: Springer.

Athey, S. and Nekipelov, D. (2010) 'A Structural Model of Sponsored Search Advertising Auctions', Sixth ad auctions workshop, New Haven.

Bartlett, M.S. (1950) 'TESTS OF SIGNIFICANCE IN FACTOR ANALYSIS', *British Journal of Statistical Psychology*, vol. 3, no. 2, June, pp. 77-85.

Bartlett, M.S. (1951) 'A FURTHER NOTE ON TESTS OF SIGNIFICANCE IN FACTOR ANALYSIS', *British Journal of Mathematical and Statistical Psychology*, vol. IV, no. 1, March, pp. 1-2.

Bart, Y., Shankar, V., Sultan, F. and Urban, G.L. (2005) 'Are the Drivers and Role of Online Trust the Same for All Web Sites and Consumers? A Large-Scale Exploratory Empirical Study', *Journal of Marketing*.

Belanger, F., Hiller, J.S. and John, W.S. (2002) 'Trustworthiness in electronic commerce: the role of privacy, security, and site attributes', *The Journal of Strategic Information Systems*, pp. 245-270.

Beth, T., Borcherding, M. and Klein, B. (1994) 'Valuation of Trust in Open Networks', Proceedings 1 European Symposium on Research in Security (ESORICS), Berlin, 3-18.

Burke, R., Mobasher, B. and Bhaumik, R. (2005) 'Limited knowledge shilling attacks in collaborative filtering systems', Workshop on Intelligent Techniques for Web Personalization.

Burke, R., Mobasher, B., Zabicki, R. and Bhaumik, R. (2005) 'Identifying attack models for secure recommendation', Beyond Personalization: A Workshop on the Next Generation of Recommender Systems.

Burke, R., O'Mahony, M.P. and Hurley, N.J. (2011) 'Robust Collaborative Recommendation', in Ricci, F., Rokach, L. and Bracha, S. (ed.) *Recommender Systems Handbook,* New York: Springer.

Burke, R., O'Mahony, M.P. and J., H.N. (2011) 'Robust Collaborative Recommendation', in Ricci, F., Rokach, L., Shapira, B. and Kantor, P.B. (ed.) *Recommender Systems handbook*, New York: Springer Science+Business Media, LLC.

Burt, C. (1952) 'Tests of significance in factor analysis', *British Journal of Psychology*, vol. 5, pp. 109-133.

Carlson, K.D. and Herdman, A.O. (2012) 'Understanding the Impact of Convergent Validity on Research Results', *Organizational Research Methods*, pp. 17-32.

Chai, T. and Draxler, R.R. (2014) 'Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature', *Geosci. Model Dev.*, no. 7, pp. 1247-1250.

Cornière, A. (2016) 'Search Advertising', *American Economic Journal: Microeconomics*, vol. 8, no. 3, pp. 156-88.

Cpanel, LLC (2020) *Create an exceptional hosting experience*, 26 January, [Online], Available: https://cpanel.net/ [26 January 2020].

Dorfman, K. (2019) *Comparing Means: The t-Test*, UMass biology Department.

Fraering, M. and S. Minor, M. (2013) 'Beyond loyalty: customer satisfaction, loyalty, and fortitude', *Journal of Services Marketing*, vol. 27, no. 4, pp. 333-344.

Ghose, A. and Yang, S. (2009) 'An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets', *Marketing Science*, vol. 55, no. 10, pp. iv-1753.

Gorsuch, R.L. (1983) *Factor Analysis, 2nd Edition*, 2nd edition, Mahwah: Lawrence Erlbaum Associates.

Grewala, D., Iyer, G.R. and Levya, M. (2004) 'Internet retailing: enablers, limiters and market consequences', *Journal of Business Research*, vol. 57, no. 7, pp. 703-713.

Hox, J. and Bechger, T. (2014) 'An Introduction to Structural Equation Modeling', *Family Science Review*, pp. 354-373.

IEEE (2011) *Bandwidth Trends on the Internet. A Cable Data Vendor's Perspective*, 1 September, [Online], Available: http://www.ieee802.org/3/ad_hoc/bwa/public/sep11/cloonan_01a_0911.pdf [30 November 2014].

Israel, G.D. (1992) *Determining Sample Size*, Florida: University of Florida.

Jiang, W., Wang, G., Bhuiyan, M.Z.A. and Wu, J. (2016) 'Understanding Graph-Based Trust Evaluation in Online Social Networks: Methodologies and Challenges', *ACM Computing Surveys*, vol. 49, no. 1, May, pp. 1-35.

Jones, J. and Barry, M.M. (2011) 'Developing a scale to measure trust in health promotion partnerships', *Health Promotion International*, vol. 16, no. 4, February.

Jumia KE (2021) *Rate & Review*, 29 March, [Online], Available: https://www.jumia.co.ke/customer/reviewsratings/detail/?sku=.

Keith, R.J. (1960) 'The Marketing Revolution', *Journal of Marketing*.

Kenya National Bureau of Statistics (KNBS) (2019) *Launch of the Gross County Product 2019 Report*, 13 February, [Online], Available: https://www.knbs.or.ke/ [06 January 2020].

Kline, R. (2005) *Principles and Practice of Structural Equation Modeling, Fourth Edition*, New York, London: The Guilford Press.

Koren, Y. and R, B. (2011) 'Advances in Collaborative Filtering', in Ricci, F. and Rokach, L. *Recommender Systems Handbook*, New York,: Springer.

Kotler, P. and Keller, K.L. (2006) 'Defining Marketing for the 21st Century', *Marketing management*, pp. 3-33.

Kwak, S.G. and Kim, J.H. (2017) 'Central limit theorem: the cornerstone of modern statistics', *Korean Journal of Anesthesiology*, vol. 2, no. 70, April, pp. 144 - 156.

Lahitani, A.R., Permanasari, A.E. and Setiawan, N.A. (2016) 'Cosine similarity to determine similarity measure: Study case in online essay assessment', 4th International Conference on Cyber and IT Service Management, 1-6.

Lake, J., Gerrans, P., Sneddon, J., Attwell, K., Courtenay, B.L. and Anne, L.J. (2021) 'We're all in this together, but for different reasons: Social values and social actions that affect COVID-19 preventative behaviors', *Personality and Individual Differences*, vol. 178, March.

Lam, S.K. and Riedl, J. (2004) 'Shilling recommender systems for fun and profit', The 13th International World Wide Web Conference, 393-402.

McIntyre, F.S., L., T.J.J. and Gilbert, F.W. (1999) 'Consumer Segments and Perceptions of Retailing Ethics', *Journal of Marketing Theory and Practice*, pp. 43-53.

McLeod, S. (2018) 'Maslow's Hierachy of Needs', *SimplePsychology*, May.

Mobasher, B., Burke, R. and Bhaumik, R.S.J.J. (2007) 'Attacks and Remedies in Collaborative Recommendation', *IEEE Computer Society*, Available: www.computer.org/intelligent.

Mobasher, B., Burke, R., Bhaumik, R. and Williams, C. (2005) 'Effective attack models for shilling item-based collaborative filtering system', 2005 WebKDD Workshop.

Narayanan, S. and Kalyanam, K. (2015) 'Position Effects in Search Advertising and their Moderators: A Regression Discontinuity Approach', *Marketing Science*, pp. 309-472.

Ngwawe, E.O., Abade, E.O. and Mburu, S.N. (2020) 'Context-Aware Computational Trust Model for Recommender Systems', *EJECE, European Journal of Electrical Engineering and Computer Science*, vol. 4, no. 6.

Ngwawe, E.O., Abade, E.O. and Mburu, S.N. (2021) 'Improving Online Experience Using Trust', *Adjustment Factor for Recommender Systems*, vol. 63, pp. 83-89.

Niwattanakul, S., Singthongchai, J., Naenudorn, E. and Wanapu, S. (2013) 'Using of Jaccard Coefficient for Keywords Similarity', International MultiConference of Engineers and Computer Scientists, Hong Kong.

O'Donovan, J. and Smyth, B. (2005) 'Trust in recommender systems', *Proc. of the 10th International Conference on Intelligent User Interfaces*, pp. 167-174.

O'Mahony, M., Hurley, N., Kushmerick, N. and Silvestre, G. (2004) 'Collaborative recommendation: A robustness analysis', *ACM Transactions on Internet Technology*, vol. 4, no. 4, November, pp. 344-377.

Oxford University Press (2018) *English Oxford Living Dictionaries*, 19 November, [Online], Available: https://en.oxforddictionaries.com/definition/mall [19 November 2018].

Parasuraman, A., Zeithaml, V.A. and Malhotra, A. (2005) 'E-S-QUAL A Multiple-Item Scale For Assessing Electronic Service Quality', *Journal of Service Research*, vol. 7, pp. 213–233.

Rao, C.R. (1955) 'Estimation and tests of significance in factor analysis', *Psychometrika 20*, pp. 93-111.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and J.Riedl (1994) 'Grouplens: An open architecture for collaborative filtering of netnews', 175-186.

Ricci, F., Rokach, L. and Shapira, B. (2011) *Introduction to Recommender Systems Handbook*, New York: Springer Science+Business Media, LLC.

Roman, S. (2007) 'The Ethics of Online Retailing: A Scale Development and Validation from the Consumers' Perspective', *Journal of Business Ethics*, vol. 72, pp. 131-148.

Román, S. and Munuera-Alemán, J.-L. (2005) 'Determinants and consequences of ethical behaviour: An empirical study of salespeople', *European Journal of Marketing*, pp. 493-495.

Rosseel, Y. (2012) 'lavaan: An R Package for Structural Equation Modeling', *Journal of Statistical Software*, vol. 48, no. 2, May, Available: http://www.jstatsoft.org/.

Rust, R.T. and Oliver, R.L. (1994) 'Service Quality: Insights and Managerial Implications from the Frontier', in Rust, R.T. and Oliver, R.L. (ed.) *Service Quality: New Directions in Theory and Practice*, Thousand Oaks: SAGE Publications, Inc.

Sergio, R. (2003) 'The Impact of Ethical Sales Behaviour on Customer Satisfaction, Trust and Loyalty to the Company: An Empirical Study in the Financial Services Industry', *Journal of Marketing Management*, vol. 19, no. 9-10, pp. 915-939.

Sergio, R. (2007) 'The Ethics of Online Retailing: A Scale Development and Validation from the Consumers' Perspective', *Journal of Business Ethics*, no. 72, pp. 131-148.

Shafer, D.S. and Zhang, Z. (2012) 'Hypothesis Testing', in Shafer, D.S. and Zhang, Z. *Beginning Statistics*, Charlotte: Creative Commons licensed.

Shani, G. and Gunawardana, A. (2011) 'Evaluating Recommendation Systems', in Ricci, F., Rokach, L., Shapira, B. and Kantor, P.B. (ed.) *Recommender Ystems handbook*, New York: Springer.

Shani, G. and Gunawardana, A. (2011) 'Evaluating Recommendation Systems', in Ricci, F., Rokach, L., Shapira, B. and Kantor, P.B. (ed.) *Recommender Systems Handbook*, New York.

Sheugh, L. and Alizadeh, S.H. (2015) 'A note on pearson correlation coefficient as a metric of similarity in recommender system', AI & Robotics (IRANOPEN), 1-6.

Statistics Solutions (2020) *Factor Analysis*, 06 June, [Online], Available: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/factor-analysis/ [06 June 2020].

Stead, B.A. and Gilbert, J. (2001) 'Ethical Issues in Electronic Commerce', *Journal of Business Ethics*, pp. 75-85.

Stein, C.M., Morris, N.J. and Nock, N.L. (2012) 'Structural Equation Modeling', *Methods in molecular biology*.

Suhr, D.D. (2009) 'Principal Component Analysis vs. Exploratory Factor Analysis', SUGI 30 Proceedings.

Tavani, H.T. (2000) 'Privacy and Security', in Langford, D. *Internet Ethics*, New York: St Martin's Press.

The R Foundation (2021) *The R Project for Statistical Computing*, 29 March, [Online], Available: https://www.r-project.org/.

Themeforest (2015) *Electro Electronics Store WooCommerce Theme*, 22 January, [Online], Available: https://themeforest.net/item/electro-electronics-store-woocommerce-theme/15720624 [29 January 2016].

Victor, P., Cock, M. and Cornelis, C. (2011) 'Trust and Recommendations', in Ricci, F., Rokach, L., Shapira, B. and Kantor, P.B. (ed.) *Recommender Systems Handbook*, New York: Springer Science+Business Media, LLC 2011.

Victor, P., Cornelis, C., De Cock, M. and Teredesai, A.M. (2009) 'Trust- and distrust-based recommendations for controversial reviews', *IEEE Intelligent Systems*.

Wang, Y., Cai, Z., Yin, G., Gao, Y., Tong, X. and Han, Q. (2016) 'A game theory-based trust measurement model for social networks', *Compu Social Networks* , vol. 3, no. 1, May.

Welch, E.W., Hinnant, C.C. and Moon, M.J. (2005) 'Linking Citizen Satisfaction with E-Government and Trust in Government', *The Journal of Public Administration Research and Theory*, vol. 16, no. 3, July, pp. 371-391.

White, T. (2012) 'Hadoop: The Definitive Guide', in Blanchette, M.L.a.M. (ed.) *Hadoop: The Definitive Guide*, Third Edition edition, Carlifornia: O'Reilly Media, Inc.

Wikipedia, the free encyclopedia (2018) *Shopping Mall*, 19 November, [Online], Available: https://en.wikipedia.org/wiki/Shopping_mall [19 November 2018].

Woocomerce (2016) *Build exactly the eCommerce website you want*, 24 January, [Online], Available: https://woocommerce.com/ [29 January 2017].

WordPress.com (2016) *Create a website in minutes.*, 01 January, [Online], Available: https://www.wordpress.com/ [01 January 2016].

Yasmin, A., Tasneem, S. and Fatema, K. (2015) 'Effectiveness of Digital Marketing in the Challenging Age: An Empirical Study', *International Journal of Management Science and Business Administration*, vol. 1, no. 5, pp. 69-80.

Yin, C., Wang, J. and Park, H.J. (2017) 'An Improved Recommendation Algorithm for Big data Cloud Service based on the Trust in Sociology', *Neurocomputing*, July.

Yin, C., Wang, J. and Park, J.H. (2017) 'An Improved Recommendation Algorithm for Big data Cloud Service based on the Trust in Sociology', *Neurocomputing*.

Zait, A. and Bartea, E.P. (2011) 'METHODS FOR TESTING DISCRIMINANT VALIDITY', *Management and Marketing*, vol. IX, no. 2.

Zait, A. and Bertea, P.E. (2012) 'Methods for Testing Discriminant Validity', *Research Papers in Economics*.

Zeithaml, V.A. and Bitner, M.J. (2003) *Services Marketing: Integrating Customer Focus across the Firm*, New York: Irwin McGraw-Hill.