Implementation of Trust reputation System (TRS) for E-commerce Applications using Data Mining

Sunil Katkar¹, Karan Pande², Pooja Patil³ & Qureshi Ahmed⁴
¹,²,³,⁴Department of Computer Engineering, Vidyavardhini’s College of Engineering and Technology

Abstract- With rapidly increasing use of internet sites mainly shopping is one of popular among all other sites. To keep it growing and stable one factor is important that is ‘trust’ hence aiming at designing a reputation system which will play crucial role in building trust. We aimed at designing TRS system which will give assurance to customers to keep trust on internet sites. TRS algorithm will not only calculate trust degree of user but also the score of the product with accuracy.

Keywords- component, Trust, Trustworthiness, Reputation, Trust Reputation Systems, e-commerce, decision-making, Textual feedbacks, semantic analysis, rating.

1. Introduction
Trust is an important term in any social relationship as well as in commerce transactions. In traditional world, a buyer can often see both the seller and the product, verify its quality of the product, and bargain with the seller. Thus, it is possible for each of them to believe the trustworthiness of each other and to be convinced about the product betterment. Various technologies available, helps users in order to make the transaction more secure and safe, [1] but they fail to construct a trustful reputation about a product or a service.

Under such circumstances, Trust Reputation systems (TRS) takes the situation by boosting trustworthiness among a group of participants according to transaction factors and to their historic path in the web transaction. In fact, e-commerce users prefer to check users review about a product, in order to develop their own trust and reputation experience. Users believe that they all of them have a common interest which is to know about the trustworthiness of the transaction and product [3], [4]. Therefore feedbacks, scores or any other information which are given by users are very important and it need to be truthful and trustful.

In TRS, users are generally considered as experienced people with interesting knowledge on the targeted product. That’s the reason why we need a TRS that can provide security and trustworthiness to the score and feedbacks associated to any product. In reality, TRS is an important mechanisms that helps to identify malicious activities of the users whose intention is to make it false and classify the score associated to the product positively or negatively.

As mentioned in the paper there are various things such as [2], [4], [6], and [7] that adopted algorithms for computing a reputation or specifying a set of possible reputations or ratings. However, few of them such as [13], [14], and [16] have been dedicated to the semantic analysis of textual feedbacks in order to generate a trust score for the product and more importantly the trust degree of the user.

In contrast to such papers, our main contribution in this work is to examine the attitude used by the user toward specific prefabricated textual feedbacks. Our motivation is to provide the user with the possibility to like or dislike – via a specific interface- some feedbacks summarizing several former users” feedbacks in addition to fake and prefabricated feedbacks. When the user gives his appreciation (a numeric value) on the product within his textual feedback, then the user’s feedback and appreciation is validated.

Our approach relies on an algorithm that includes semantic feedback analysis in order to generate most trustful reputation score for a product since feedbacks affect user’s decisions more than numeric scores alone. [15] This proposed algorithm calculates and updates also the trust degree of the user after any participation in the TRS.

The remaining of this paper is built as follows. In section 2 we remind the terminology of trust and reputation systems. Section 3 presents some related work. The architecture and the algorithm related to our TRS are detailed in section 4. Finally, we come up with some concluding remarks and an outlook on future work.

The user starts by selecting an appreciation (rating) and giving a textual feedback about a specific product. When the user clicks on submit in order to conform the given information, we are going to redirect the user to another interface showing this message for example: “please give us your opinion about the following feedbacks before conforming the information you gave below:”
In fact, in this interface we will find chosen feedbacks from the database from different types. Those feedbacks can be fabricated in order to arrange all the registered users’ feedbacks stored in the database. The generated feedbacks can be stored in another knowledge base. So the system will keep on adding feedbacks in the ordinary data base, we will fill the knowledge data base with prefabricated feedbacks using text mining and algorithms. However, some users can give already summarized feedbacks that can directly be included in the knowledge data base. Indeed, there are many text mining and data mining algorithms and tools that could search the most appropriate feedbacks that are first of all related to the product and that can recapitulate and summarize most of each type of the users’ feedbacks.

II. TRUST AND REPUTATION

BACKGROUND

Reputation and trust must be assigned closely to web content in order to estimate usefulness of web content and to use its trustworthiness. In fact, we need this estimation in order to help users react and interact with the content, share content, comment and rate. We need to create a social interaction, to develop further relations in order to collect more and more information and then filter them and decide about their reliability using TRS. Consequently, users who are ready to estimate the content can then rate the content and become useful and then “raters”. This rating is of course accompanied by a rating algorithm. In fact, the rating aims at generating a reputation. Then the assignment between the rating and the reputation must be done. A reputation algorithm calculates somehow the reputation of the content from ratings. [6]

In this work, we have added textual feedbacks analysis in the calculus of reputation score of the product. We need to focus on some definitions going to be used in the paper. We start by highlighting some definitions of trust and reputation

A. Trust definitions

Definition 1

The definition of Trust is closely related to the willingness to pay, in online markets such as eBay, the actual selling price without taking an adequate reflection of the “underlying” trust of other buyers in the seller or even the product. [17]

In fact, trust is also defined as the ability to rely on someone or something, to rely on its truthiness, on its strength to prove its reliability. In e-commerce, being trustful is a quality characterizing a product that a user claims to know either intuitively or from a past experience which is more trustful, or because other users estimate that it is a reliable product. [6], [7]

However a trust which is not based on logical evidence corroborated by real experience and analytical examination is useless and doesn’t help generating a reliable reputation

Definition 2

Trust is also considered as a subjective evaluation of the potential outcomes and risks involved by relying on a partner. [18]. Indeed, Trustworthiness refers to how much we can trust on the product or the user’s intervention concerning the product, so as to help any user construct his or her opinion and reputation image concerning the product. However, users have different intentions while sharing their “experience” by rating or writing a textual feedback. [6], [15] Then we ought not to trust the users’ honesty shown through their interventions.

Definition 3

In general, trust can be described as a willingness of an agent to be vulnerable to another agent interventions and convinced by actions of this agent based on prevision expectations that the other agent performance will be as a particular important action to the trustee, irrespective of the ability to monitor or control that other party This vulnerability has to be transformed to a convincing strength of a party that is able to give structured and logical statement accompanied with well-built arguments and proofs. [11], [12]

Actually, this trust is represented by rating and semantic feedbacks. Moreover, unreliable scores and feedbacks must be detected and examined with specific care in order to generate a most trustworthy reputation.

B. Trust Reputation Systems Definitions

Reputation is generally said or believed to be about a persons or things’ character or standing. Related to products and services, it is the subjective opinion based on feelings, past experiences and the viewpoint of a circle of “trustful” people. Reputation is often used in the sense of the community’s general reliability and trustworthiness evaluation of a service entity. [7], [8], [19]

Therefore, this trust reputation needs to be gathered, collected and filtered in order to generate the most trustworthy reputation associated to a service, a product or a user.

To do that, Trust Reputations Systems are available tools that can work on trust reputation using algorithms with the purpose to calculate the most reliable evaluation.

We give hereafter some definitions of trust reputation systems.

Definition1: TRS towards the buyer, the seller and the whole community
Trust and reputation systems (TRS) are an important class of decision support tools that can help reduce risk when dealing with in transactions and interactions online. From the individual buyer’s viewpoint, a TRS can help reduce the risk related to any particular interaction in the e-commerce application. From the service provider’s viewpoint, it represents a marketing tool attracting more buyers and convincing them to purchase. From the community viewpoint, it represents an application of social interaction, moderation and control, as well as a method to assess trust by improving the quality of online markets and communities. [1], [9], [19]

Definition 2: Robustness of TRS as a decision-making tool in e-commerce
TRS represent a decision-making mechanism that allows parties to rate each other in order to give consumers a relative clear idea whether to go through a transaction or not. They allow them to evaluate the reputation of a product, transaction, online merchant through the experience of other users. In virtual environments that apply reputation systems, users can decide whether to trust an online merchant based on the probable trust they have on the provider of the feedback. [3] Actually in e-commerce, users are obliged to trust information from unknown and unreliable sources and anonymous users. [10], [11]

That’s why, robust trust reputation systems are supposed to reduce the probability that a user is untrustworthy. Consequently, TRS reduces the probability that the user’s interaction and input in the application is untrustworthy. Reducing that risk has an efficient impact only at the moment of the user’s intervention. Then the robustness of a TRS depends on how much it reveals truthfulness of user’s recommendations to each other in a specific time and context.

III. RELATED WORK
Many authors such as the authors of [2], [4], [6], and [7] propose in their work several TRS architectures with different algorithms to calculate reputation score related to the product. Nevertheless, few academic works on TRS has been devoted to the inclusion of the semantic analysis of feedbacks in the calculation of the trust score of the product and especially the trust degree of the user. Even in studies attempting to provide more complex reputation methods such as [2], [7], [13], [14], [16], some issues are still not taken into consideration, such as the credibility of referees, the update of the trust degree of the user at any intervention, the age of the rating and the feedback or the concordance between the given rating which is a scalar value and the textual feedback related to it.

In contrast to those TRS, our proposed design treats these issues and uses an algorithm that includes semantic analysis of textual feedbacks in order to calculate the trust degree of the user and a most trustful reputation score for the product.

A. The relying party’s Credibility issue
In e-commerce, we do not only need to generate trustful scores for product, users and feedbacks but we need also to propagate them through a trust reputation network. In [2], a reputation network is defined as a network that can be represented by a graph with its nodes related with each other by an arc or multiple arcs or no one. The graph reflects the trust network. The nodes represent users or agents interacting with each other directly or indirectly by giving recommendations and rating products. Arcs can refer to the information flow between agents when propagating ratings and recommendation from one to another. [2], [3]

The authors of [2] propose a method that uses subjective logic in order to analyze trust network (TNA-SL). Hence, this method aims to model in a simple way the relationship between different agents. A single arc means a single trust relationship between two nodes A and B [A; B] meaning that A trusts B. However trust has degrees and each arc must have its trust degree. This trust relation is built thanks to a transaction or a recommendation but obviously through something tangible (a proof of trust). In fact, if A trusts B that means B has a positive reputation and does he deserve it or not. Because somehow we would have a group of agent knowing each other in real and trusting each other in the network in order to be seen as trustful by others and then falsely rate other agents or product. Then each user should be analyzed alone.

Consequently, we should take into consideration the trust degree of the arc and also the trust degree of the nodes, because the arc could be a feedback given by B and used by A to deal with a transaction. That’s why A trusts B.

In our approach, for each user who wants to leave a rating (appreciation) and a textual feedback (semantic feedback), we analyze his intervention using our algorithm. Indeed after verification, the user’s recommendation is going to be reached by any other user and then it is a recommendation for everyone interested in the product or even not. Then, we suppose that we have a path relaying every user. The most important is to analyze at any intervention the user’s attitude in order to deduce the user’s intention concerning the rating of that specific product.

B. The trust update issue
Another factor which is important in the analysis realized by a TRS is the date of the creation or the establishment of the arc. The most recent arc, which relays two nodes having the same interest on a topic or a product, is more meaningful and useful than an old one. We can add an attribute “Date” in the graph table in database of [2]. Date represents the date of the creation of the arc relating A to B on product. In addition to the age of rating issue, the trust update is a very important key contributing to the robustness of a TRS.

In fact, the authors of [3] think that the transitivity of trust is a derived trust from an existing trust between agents. However, we can find that Alice who trusts Bob doesn’t trust Charlie because she has another friend who is Ben and she trusts him more than Bob. Ben doesn’t trust Charlie then Alice will not trust Charlie. This arc is multiple and has two different statement and results with the same scope. And Bob even may change his mind after a bad transaction or situation with Charlie. In this case, we have to pay attention to the update of statement or feedbacks taken into consideration in the calculus of reputation score.

In our algorithm, at every intervention of a user, we updated his already calculated trust degree if had already participated in rating and commenting a product. If not we generate for him a new trust degree, according to his attitude modelled by “liking” or “disliking” prefabricated feedbacks concerning the same product displayed to him, in order to validate his own rating and feedback (see sect.4).

C. Reward and punishment in TRS

In fact, one of the biggest problems for reputation systems is users who falsely give ratings using many “faking identities”. However, in our TRS, even though the user has given a rating, we allow him to give ratings as much as he wants to, but at any intervention, if he changes his identity we consider him as a new user and we calculate a new trust degree which plays the role of coefficient according to his rating. A possible example for such a rating method might be school marks and coefficient (see sect.4). To demonstrate the impact of a mark, the coefficient must be higher and vice versa because it is a multiplication as an arithmetic operation. If the user is trustful then his trust degree will be higher and we can touch the impact of his intervention in the global rating of the product.

[4] Proposes a novel reputation based system on dynamic coalition formation, where buyers are rewarded with virtual credits who have similar subjects and rich experiences for helping others find trustworthy sellers to conduct business successfully. Besides, the authors of [14] use an approach that aims to calculate the trust weight. In fact, once the transaction is carried between the Web Service Providers and the Web Service Consumers, a reward or punishment is affected to users and WSPs according to the accuracy and reliability of their recommendations. [14] They focus on the punishment and reward of users (it is equal for WSPs). So WSP1 sends the satisfaction of the user who asked for the service. A simple mechanism will be established to measure the divergence between the final satisfaction of the user, Sat, and the previously given recommendation of users.

In fact, in our model we are not going to use other user’s recommendations to determine the trustworthiness of the current user since even those recommendations can be untrustworthy. Our approach aims to reward users virtual credit which represents their credibility (trust degree), if they like a trustworthy feedback or unlike an untrustworthy one. And we punish them if they like an untrustworthy feedback or unlike a trustworthy one. However the reward and the punishment have levels and degrees depending on the degree of the trustworthiness of the feedback. (See sect.4.2)

IV. OUR TRUST REPUTATION SYSTEM DESIGN

A. Algorithm Description

Actually, the trust reputation system receives the user’s feedback and ratings about the product we have to verify the concordance between them in order to avoid and eliminate contradiction or malicious programs attacking our system. In the redirected interface, we will display several prefabricated feedbacks which are stored in the system. However, the number of feedbacks to be liked or disliked can be specified by the user.

In fact, we are trying through this redirection to detect and analyze the user intention behind his intervention on the e-commerce application. Hence, we examine and evaluate user’s view using other pre-fabricated feedbacks with different types. Consequently, we use our reputation algorithm studied in section [4.2] in order to generate the user trust degree which plays the role of a coefficient and then rectify his appreciation according to his trust degree and generates the score of the feedback.

In fact, every feedback has trustworthiness in a threshold [-5, 5]. If the trustworthiness is closest to 5, the most trustworthy the feedback is. The feedback is very untrustworthy, if the trustworthiness is close to -5. If the feedback is trustworthy its score would be included in [0,5] else it would be included in [-5.0].
B. TRS algorithm

Reputation algorithm used in this TRS is using semantic feedbacks analysis in order to generate a trustful reputation score for the product. Actually, we have 3 types of feedbacks:

** Positive feedbacks: represent opinions that expressing a positive point of view about the product. Those corrective opinions contain a positive content concerning the product. Then, the adjective positive is referring to the nature of the content of the feedbacks not its trustworthiness. However, each feedback irrespective of its type can have either a positive trustworthiness or a negative trustworthiness. Either positive trustworthiness or negative one, it is gradual: it has degrees as float in a threshold of [-5.5].

**Negative feedbacks: represent opinions talking negatively about the product. Logically, the users giving such opinions are not satisfied of the commented product. This feedback could be telling the truth or a part from the truth or could be far from the truth. That’s why, each feedback has its trustworthiness represented by a float number between -5 and 5.

**mitigated feedbacks: represent feedbacks that are talking positively towards some aspects of the product and negatively towards other aspects of the product. They are also characterized by trustworthiness included in [-5.5].

**contradictious feedbacks: represent feedbacks with a contradictious content for example a feedback where the user is not talking about the specified product but another one or he/she is affirming that the camera of a mobile phone is great and later in the same opinion is saying that the camera is very bad. In fact, we have to start by detecting the contradictious feedbacks. Then we are in need of a semantic analysis algorithm and tool that can detect the contradiction in a specific content related to a product. We can personalize the analysis according to the product. For instance, if the user says that “the swimming pool of the hotel which doesn’t afford one is not clean”, the algorithm must be able to detect this great contradiction. We can give to the algorithm for each product as an input the property of the algorithm; if there is no similarity we can consider it as a contradiction. But the agreement includes the meaning of course. Because if the user writes that the negative thing about this hotel is that there is no swimming pool. He’s telling the truth then obviously the presence of an absent property in a feedback doesn’t mean that there is a contradiction.

In this paper, we will not discuss the text mining and the data mining algorithms and tools. We’re going to develop it in a future work. Actually, before sending the user’s feedback and appreciation about the product to the trust reputation system, we have to verify the concordance and the alliance between them so we don’t have contradiction.

```
Pseudo-code to verify the concordance between the rating and the textual feedback:
Boolean concordance;
Concordance =Test_concordance (int appreciation, string feedback);
If (concordance)
    URL (url_feedbacks_interface);  //redirection to the feedbacks interface
Else
    URL (url_page);  // we thank the user for his intervention and we put him temporally in a //blacklist for unconformity
```

After verifying the concordance between the appreciation and the textual feedback were going to redirect the user for selecting prefabricated feedbacks. [11]

Then the user is going to click on like or dislike according to each feedback. The event of click will be managed in order to get some information needed in the calculus of the trust degree of the user. The function uses as a parameter the id of the feedback in order to get from Knowledge base its trustworthiness. We need to get also the previous trust degree of the user if he has been already engaged in a transaction or he has used the application for rating purpose. The user choices either “like” or “dislike” is an important parameter to determine his trustworthiness.
After extracting the parameters going to be used in the next calculus, were going to calculate the trust degree of the user taking into consideration the type of the trustworthiness of the feedback and the user choice. The calculus of the trust degree of the user can be an update if the user has already a trust degree. As seen in the function below, “get_trust_degree_user (login)” uses the login of the user in order to select his last trust degree. We can add as a parameter the id of the product in order to select his trust degree by a specific product. As a result, we will select the trust degree of the user, who logged in with a specific “login”, calculated after his rating intervention on a specific product (id_product). In that case, we have to generate a global trust degree of the user by product and a global trust degree according to his general interventions. This method could be developed in a future paper.

However, our proposed algorithm rewards the user by incrementing his trust degree if he likes a trustworthy feedback or he dislikes an untrustworthy one. The incrementing value (reward) depends on the value of the feedback trustworthiness. When the user choice is a “like”, the greatest is the feedback trustworthiness, the greatest the reward would be and vice versa. And when the user dislikes a feedback, the greatest is the untrustworthiness of the feedback, the greatest the reward would be and vice versa.

However, we need to respect the threshold [-5; 5]. We can call it the values normalization or the standardization of values. In a future work, we can give a detailed method of values normalization in order to be more concise and precise, because to approach -7 to -5 is less precise than to multiply by a coefficient or a percentage.

In fact, the function returns the trust degree of the user updated according to his current participation. As a result, if his trust degree is positive we will take into consideration his rating realized in the first interface of the e-commerce application before redirection. However, if his trust degree is negative, we will not consider his appreciation in the calculation of the global trust score of the product and we can preserve his feedback in order to use to fabricate other feedbacks. And then his feedback would be considered as untrustworthy as his creator or writer and vice versa. Consequently, we affect the trust degree of the user to the degree of trustworthiness of the user’s feedback as in the last line of the pseudo-code below.

Pseudo-code for the function that gets parameters used in the calculus of the trust degree of the user:

```
Function get_infos_click (int id feedback) as list {
    double Feedtrustworth; // the feedback trustworthiness stored in knowledge base
    // its value is between -5 and 5
    String User choice; // represents the user’s choice either it is a “like” or a “dislike”
    double degree_trust_user; // a double value representing the trust degree of the user [-5; 5]
    double scorefeed; // a double value between -5 and 5
    degree_trust_user=get_trust_degree_user (login);
    Feedtrustworth= getFeedtrustworth (idfeedback); // this function gets the trustworthiness of the feedback either positive or negative value between -5 and 5
    Userchoice=getuserchoice (idfeedback); // this function get the user choice after the click // from the interface
    List listinfos= [Feedtrustworth, Userchoice, degree_trust_user];
    Return listinfos ;
}
```

Pseudo-code for the calculus of the trust degree of the user:

```
function calculate_degree_trust_user () as double{
    list listinfos;
    Int idfeedback=get_idfeedback();
    Listinfos=get_infos_click (idfeedback);
    List listScore;
    Feedtrustworth= Listinfos[0];
    Userchoice= listinfos[1];
    Degree_trust_user=Listinfos[2];
    if (0<feedtrustworth<1.5) and (userchoice=”like”) Or (-1.5<feedtrustworth<0) and (userchoice=”dislike”)
        Degree_trust_user=0.25
    If (1.5<feedtrustworth<2.5) and (userchoice=”like”) Or(-2.5<feedtrustworth<1.5) and (userchoice=”dislike”)
        Degree_trust_user=0.5
    If (2.5<feedtrustworth<3.5) and (userchoice=”like”) Or(-3.5<feedtrustworth<-2.5) and (userchoice=”dislike”)
        Degree_trust_user=0.75
    If (3.5<feedtrustworth<5) and (userchoice=”like”) Or (-5<feedtrustworth<-3.5) and (userchoice=”dislike”)
        Degree_trust_user=1
    Degree_trust_user+=1
    } // the end of the function
```

Pseudo-code for the values standardization:

```
// to respect the threshold [-5;5] {
    If (Degree_trust_user<-5)
        Degree_trust_user=-5;
    Else if (Degree_trust_user>5)
        Degree_trust_user=5;
    Return degree_trust_user;
} // the end of the function
```

```
After that, we have to generate the global trust reputation score of the product using the user’s appreciation (rating) and his trust degree. In fact, the best possible example for such a rating method can be school marks and coefficients. Actually, at school, when a course is important for a certain field, its coefficient would be great and then the effect of its mark would be greater. Similarly over here, we consider the trust degree of the user as a coefficient and his appreciation as a mark. Consequently, to calculate the global trust score of the product, we sum all the appreciation values multiplied by their respective coefficient and then divide the result of the summation on the summation of all coefficients:

\[
\frac{X + b + y}{a + h}
\]

\* \(X\) represents the summation of all users’ appreciations.
\* \(\text{“b”}\) represents the new coefficient obtained by each user.
\* \(\text{“a”}\) represents the summation of all users’ trust degrees.

In the formula below, we calculate the summation of a user’s appreciation multiplied by each user’s trust degree respectively. We divide the result by the summation of all users’ trust degree from the first until the last one for whom we have just calculated the trust degree or updated it.

**Remark:** We can store the “X” and the “a” in different areas in the database, so we can get them separately and then calculate easily:

\[
\frac{X + b + y}{a + h}
\]

**V. CONCLUSION**

TRS providing another way of trusting without interacting physically with products. Through this customer can make explicit judgement about the product. Whether the product is good to buy or not. Using algorithms like calculation of trust degree of user and product score calculation algorithm. We have implemented in order to give accurate results so that the customer will have assurance that it is the system that gives safe and reliable experience over the internet.

**REFERENCES**
