ASHESI UNIVERSITY

TrustHop: Building a Social Trust Network

Undergraduate dissertation submitted to the Department of Computer Science,

Ashesi University. Submitted in partial fulfillment of the requirements for the award of Bachelor of Science Degree in Management Information Systems

B.Sc. Management Information Systems

Papa Kwame Twumasi-Ntiamoah

DECLARATION

I hereby declare that this dissertation is my original work and that no part of it has been presented for another degree at this university or elsewhere.

Candidate's name: Papa Kwame Twumasi-Ntiamoah

Student ID: 37352022

Date: 25th April 2022.

I hereby declare that the preparation and presentation of this dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University.

Supervisor's name: Dennis Asamoah Owusu

Acknowledgement

I would like to express my sincerest gratitude to the Almighty God for directing and guiding me through this project because I could not have done it without Him. I would also like to thank my supervisor, Dennis Asamoah Owusu for his undying dedication and constructive feedback throughout this project.

I would also like to express my deepest appreciation to my family for their endless support, encouragement, and prayers throughout my journey in Ashesi. Finally, to my friends who contributed to making this paper a success. A big shoutout to “Valentinesss” and Nana Ekow Korsah. God continue to bless you all.

Abstract

In recent online social networks, each user can often assign a value to their immediate friends' level of trustworthiness. Understanding a social trust value between any two nodes in an online social network is beneficial in a range of applications, like online marketing and recommendation systems. However, assessing social trust between two members in an online social network is difficult and time-consuming. This is because existing work either created handcrafted rules based on specialized domain knowledge or required a large number of computational resources, limiting its scalability. Graph-based techniques have recently been proved to be effective at learning from graph data. Even though social trust may be represented as graph data, its advantages have a lot of potential for trust evaluation. Therefore, we begin by reviewing the characteristics of online social networks and the properties of trust. After which the two types of graph-simplification and graph-analogy methodologies would be compared and contrasted as well as their respective problems and obstacles. We then conduct a quick examination of its pre- and post-processes to present an integrated view of trust evaluation. Finally, we discuss some unresolved issues that all trust models face.

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Introduction

With the enormous growth of social network services, identifying trustworthy people has become a top priority in order to protect users' vast volumes of personal information from being tarnished by untrustworthy users. Trust is an essential component of everyday human life. It is utilized on a daily basis in some form or another. For example, when one buys food from a roadside vendor and trusts that the food will not be poisoned, or even when one trusts that televisions will function every time they are turned on.

Furthermore, someone occasionally does business with strangers and is presented with the difficult challenge of making risky decisions in an online environment. Customers or consumers on any e-commerce site, for example, read reviews of things they want to buy and are frequently faced with the decision of whether the reviews were written by website representatives posing as customers or by actual buyers. As a result of this, the question of internet trust is gaining attraction in social media. The foundation of any functioning society is trust among its members, so it is only logical to expect the same in online groups.

Internet communications are complicated because they involve contacts with persons who may or may not be strangers. As a result, a person on the internet is frequently faced with the decision of how much to trust another person for personal or professional reasons. In real life, people may ask their friends or friend's friend for information about a stranger's trustworthiness, but when it comes to online dealings, a stranger may be socially distant, and finding people to question about trustworthiness might be difficult. As a result, a methodology that can precisely predict how much one person will trust another will be extremely beneficial.

The foundation for analyzing trust in this paper is web-based social networks. We examine situations in which trust is built into a social network. The purpose of this research is to figure out how to make use of the structure of social networks and the trust connections that exist within them. In the sense that, if two people are not directly connected, a trust inference mechanism uses the connections, to create a suggestion about how much two people who are not directly connected should trust each other. We conduct a real-life survey at Ashesi University, in which the class of 2022 participated, to prove the correctness of our algorithm.

The rest of this paper is laid out as follows: The next chapter covers the prior literature review work. The formulation of the problem, as well as our contribution, is described in chapter three. Chapter four describes the experimental design and contains the detailed experimental data and analyses. Chapter five presents the conclusion and future work.

Chapter 2: Literature Review

2.1 Background

The concept of "trust" has been explored in a range of social scientific disciplines, which has resulted in a plethora of definitions [14]. Several types of trust relationships as they arise in the offline world have been discussed by psychologists, sociologists, and others [15]. Moreover, the majority of these definitions are based on traditional offline interpersonal trust. Although offline and online settings have so much in common, previous offline trust studies appear to be comparable to an online environment. Exchange is one clear commonality in this instance. Risk, fear, complexity, and price limit exchange in such situations also. Furthermore, it appears that people's social rules of interaction work in both the offline and online worlds. As a result, research on offline trust is important to online trust [16]. Therefore, our understanding of online trust should influence the development of offline trust definitions.

Josang et al. (2007) define trust as "a subjective probability by which one user expects that another user performs a given action" [17]. This definition is primarily focused on interpersonal trust, with little awareness of the internet's risk characteristics, which are especially important in the online context, therefore, it is unable to fully describe online trust. Online trust, according to Corritore et al. (2003), is "a confident anticipation in an online environment of risk that one's vulnerabilities will not be exploited" [16]. Risk, vulnerability, expectation, confidence, and exploitation are crucial ideas in this definition, and they provide a deeper understanding of online trust. Online trust with websites, internet vendors, and virtual community members has been the focus of previous studies. In contrast to previous studies on online trust, there is less research on social network trust amongst members, which is what will be primarily focused on in this study.

How can one clearly define social network trust? Utilizing Yu's amended definition, "social network trust can be defined as an individual or a group's expectation that another individual or group's word, promise, or written or verbal statement may be relied upon in the course of contact with individuals in a social network situation of risk" [13].

Studies have shown that trust is multidimensional, with ability, honesty, and compassion being the most commonly recognized three characteristics of trust [7] [13]. Certain academics, such as McKnight and Chervany (2001), feel that trust is made up of four elements. These are ability, compassion, honesty, and predictability [18]. Also, different academic groups have differing opinions on the aspects of trust; nonetheless, the previously described trust component is significantly considered.

In addition to the above, according to Merriam-Webster (nd), abilities refer to the competencies that one is able to do something in his or her specific field. Ability is "domain-specific", therefore, people with certain abilities are more trustworthy than others. Members who display the expertise and skills to give high-quality services, for example, will attract more fans and demonstrate their ability to detect member wants; additionally, because they have earned members' confidence, fans will want to follow their recommendations regarding e-commerce services.

Benevolence, on the other hand, refers to the disposition to do good (Merriam-Webster, nd). In the sense that, trusted parties will act in a positive manner to achieve the desired outcome in a relationship without bringing any benefits to the trustee. Kindness and altruism are exemplified by benevolence. When it comes to social networking, the trustee responds with appropriate advice and assistance, such as participating in continuous discussions to assist, support, and care for people. Hence, the benevolent members should actively react to consumer questions [13].

Further, according to Cambridge Dictionary (nd), “integrity refers to the quality of being honest and having strong moral principles that one refuses to change”. In other words, it is the assumption that someone will act in line with widely held beliefs, principles, and rules such as not telling lies and delivering reasonable verifiable information. As a result of this, social networking trust can help people acquire a sense of justice by enforcing an ethical norm.

However, trust is the main issue when it comes to social networks, encapsulated brilliantly by Isaac Watts who says, “Learning to trust is one of life’s most difficult tasks” (nd). People experience trust issues whenever they are making decisions in their day-to-day life, and this is worsened in online social networks owing to a lack of actual human interactions and mutual experience. Therefore, our everyday social life, which we overlook, would not be feasible without trust.

Also, according to Wang, customers globally are more likely to trust trusted sources, such as guidance from best friends and family recommendations [13]. Regardless of its simplicity and convenient functionalities, the social network has undeniably high motivational potential. Moreover, in social sciences, social trust is an important term that is closely linked to other notions such as social networks [3]. This is because social network trust is directly tied to societal goods like minimal corruption.

The foundations and sources of confidence in social network trust are numerous. To describe the sources and processes relating to the formation of trust, academics have adopted a variety of approaches [3] [13]. Nevertheless, due to the lack of a clear definition of social network trust, it is widely accepted that it encompasses essential concepts of social engagement such as reciprocity, solidarity, and fraternity [3]. Besides, the phrase “social network trust” does not suggest that individuals trust one another on a personal mere level because they are

acquainted whether particularized trust or specific trust. Rather, social network trust refers to a far larger appraisal of people's overall trustworthiness.

Despite this, social network trust reflects a person's more or less positive expectation of the outcome of their interactions with others, as well as their fundamental grasp of how society's social fabric works. However, risk, resilience, confidence, exploitation, and expectations are crucial ideas in finding a definition for social network trust and they provide a deeper understanding of online social trust. Hence, it will be especially useful for assessing the pairwise trust connection between two individuals who are not directly connected on a social network. This could be through a thorough analysis of social networks by providing the challenges and methods of a trust graph-based assessment to gain a better understanding of social networks' trust models. These kinds of trustworthiness estimations can assist users to figure out how much they can trust someone else to do certain activities.

2.2 Trust Concepts and Categories

In the section above, a few key principles were addressed. To properly describe trust and trust evaluation, more precise definitions are provided here. First, the trust definition utilized in this survey will be gone over again. As previously stated, numerous definitions of trust have been proposed, all of which apply to online social networks with graph-based trust models. As a result, rather than presenting a new concept, we refer to Josang et al. (2007).

Trust is a subjective probability by which one user expects that another user performs a given action [17].

Trustor: A trustor is a user who is attempting to determine the trustworthiness or a trust degree of another user.

Trustee: A trustee is a user whose trustworthiness or trust degree is being assessed.

Recommender: A recommender is an intermediate user who assists the trustor in evaluating the trustee's trustworthiness or trust degree.

Trusted Path: A trustworthy path from a trustor to a trustee is one that includes a trustor (who is the source), many recommenders, a trustee (who is the target), and a trust relationship among them.

Trusted Graph: A trustworthy graph from a trustor to a trustee is formed by all the trusted paths that start with a trustor and terminate with a trustee.

According to Josang (2006), there are two sorts of trusts utilized in a trusted path. These are referral trust and functional trust.

Referral Trust: Referral trust is based on a recommender's ability to recommend a good service provider (that is, a trustee).

Functional Trust: Functional trust is trust in oneself to be a good service provider (that is, trustee).

In addition to the above, referral trust normally starts from a source (a trustor) to a recommender to another recommender, according to these two definitions. Meanwhile, functional trust usually starts from a directly connected neighbour to the target (a trustee).

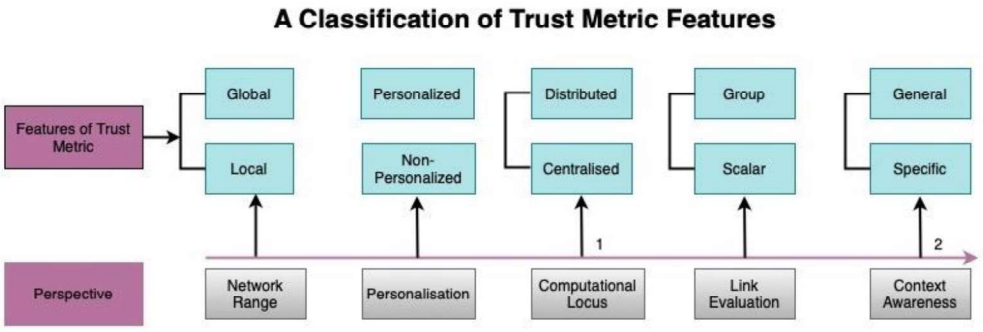


Figure 1. *Labels 1 and 2 indicate both features are suitable*

A classification of trust metric features is illustrated in Figure 1 based on previous work by Jiang et al. 2016 and Josang et al. 2007. This illustration according to Jiang, takes into account a variety of factors, such as network range (local or global metrics), personalization (personalized or not personalized), computational locus (distributed or centralized), link evaluation (group or scalar), and context awareness (general or specific) (2016). Following this, many new notions based on this classification are presented.

Local Trust Metric: takes into account the opinions of partial users, usually from the trustor's neighbourhood.

Global Trust Metrics takes into account the opinions of all users and all trust relationships among them, and it is calculated using all of the network's trust information.

Personal Trust refers to a user's trustworthiness from the perspective of another user.

General Trust refers to a user's trustworthiness without regard to any specific issue.

Specific Trust refers to a user's trustworthiness on a specific issue.

The arrows linking these different features in figure 1 illustrate their associations. To be more explicit, the computation for global trust can be done in a distributed or centralized manner, it also analyzes groups of trust claims at once, it is, however, non-personalized, and it can be general, or it can follow some specific issues. Local trust, on the other hand, is typically calculated in a centralized manner and analyzes each trust individually. It is usually personalized, unlike the global trust, and it can also be either general or specific in issues.

2.3 Related Works on Online Social Network Trust

Since the introduction of the internet, there has been substantial evidence that creating supportive interpersonal bonds online is beneficial [13]. Interpersonal trust between people online is the most crucial factor. As a result, trust has been a major issue, not only in the physical world but also in the virtual world. Recently, people are increasingly using the internet to make purchases and interact with friends.

For many businesses in e-commerce, internet retailing or marketing has been a significant channel or business strategy [2]. The main objective of online sellers is to encourage customers to make recurring purchases through their websites. Maintaining continuity in the buyer-seller relationship requires trust in sellers. As a result of this, trust has been regarded as a vital success factor and substantial research has been done [2] [3] [13]. Moreover, other research suggests that customers are unlikely to buy things from online vendors unless they have good thoughts and intentions [5].

Researchers have proposed various classification techniques for online social networks. Nepal and others (2013) used the social trust model, to examine the long-term viability of social networks to incorporate the concept of engagement trust and combine it with popularity trust to calculate the social trust of the community as well as an individual member [9]. Lin, Gao, and Li (2020) used an end-to-end system called *Guardian*, to evaluate social trust between any two users by incorporating social network structures and trust linkages [11] whereas Jiang, Wang, and Wu (2016) reviewed the characteristics of online social networks and the properties of trust by comparing and contrasting two types of graph-simplification and graph-analogy methodologies, as well as their respective problems and obstacles [11].

For these reasons, the propagative and composable character of social trust in online social networks has led to existing trust evaluation methodologies. The propagative nature of social trust, in particular, relates to the fact that trust can be conveyed from one user to another, resulting in social trust chains that connect users who are not officially connected [7]. The composable character of social trust refers to the notion that if many chains of social trust exist, trust must be aggregated [7]. In a nutshell, the essentials to efficiently evaluate social trust in online social networks are trust propagation and aggregation procedures.

Nonetheless, in present-day, deep graph convolutional neural networks for graph-structured data have made significant progress [7]. Graph convolutional neural networks gradually integrate feature information from local graph neighbours. Local information can be spread throughout the graph by stacking numerous convolutions and transformations. Social trust can be represented as graph data in online social networks, containing both social network topologies and associated trust relationships between members [1] [7]. As a result of its benefits, graph convolutional neural networks may offer excellent chances for capturing trust propagation and aggregation criteria for analyzing social trust relationships between pairs of users.

However, assessing social trust with graph convolutional neural networks is quite difficult. The first difficulty in this setting is to figure out how to express social connections and associated trust relationships together so that the propagative and composable natures of social trust may be recorded at the same time. Furthermore, social trust is frequently asymmetric; one user may trust another more than they are trusted in return. As a result, the second challenge is defining an asymmetric quality in social trust.

However, the notion of relationship communities in online social networks blurs the line between measuring trust in a postdictive manner and a predictive manner. According to

previous research, experts use social networks as virtual communities directly and believe that there is no difference in relevant online trust studies [13]. Moldoveanu and Baum (2011) propose that describing the essential parts of trust within dyads and larger groups, as well as a method for measuring trust in a noncircular and predictive manner, will help focus on the beliefs that are important to mobilization and coordination and show how trust functions to influence social capital arising from network structure after advancing arguments for the importance of interactive belief systems to successful behaviour coordination [5].

In this paper, graph convolutional neural networks are suggested to address these issues in social trust evaluation. More specifically, given the social network topology and associated trust relationships between users, an evaluation is intended to evaluate the value of trustworthiness between any two users who are not explicitly connected effectively and efficiently. For this reason, an end-to-end architecture is introduced to stack multiple trust convolutional layers to find hidden and predictive latent characteristics of trust in online social networks.

Chapter 3: Research Methodology

3.1 Research Proposal

The goal of the research is to describe, investigate and explain a phenomenon, theories, ideas or even a project. The purpose of this research paper is to explore how to compute a quality assessment of one person for another considering both individuals are not connected. The methodological approach, a case scenario, sample size, data collecting, data analysis, proposed models, and algorithm used in this study are all outlined in this section.

3.2 Research Approach

The problem is defined as "how do we compute quality assessment about person A for person B considering that person A and B are unconnected." It was argued that this data would be used to generate more acute social graph-based suggestions than traditional mathematics tools approaches and AI and information theory-based approaches. This is because the traditional mathematical approach aims to establish only a reliable mathematical model for assessing trustworthiness whereas AI and information theory-based approaches use machine learning to solve the problem of generalizability in Bayesian trust models. Our research goals can be broken down further into the following subcategories:

- Defining the directed network's edge weights: to determine edge weights, we use the structural properties of two nodes that constitute the edge. This is used to quantify the propagation of influence along with the nodes in the graphically displayed social network.
- Determining whether a node has been influenced by another node: we use vertex-dependent threshold values to assess whether a node has been influenced by another node.

- To generate neighbourhoods for producing recommendations for consumers, we use the influence propagation approach. As a result, a tailored social network-based subset of users emerges, resulting in more precise recommendations.

3.3 Research Case Study

The goal of trust is to locate the right person to work with or associate oneself with in order to complete a specific task or setting. Many elements influence a trustor's decision to cooperate or not cooperate. To demonstrate this phenomenon, a real-life example of trust is offered in the following part, as well as several circumstances that can influence the cooperation decision.

Assume Lisa needs to get her wig done for an upcoming event. Lisa is familiar with Serwaa and Laura. Serwaa has Lisa's trust, but she has little experience when it comes to the making of wigs. Despite her trust in Serwaa, Lisa will not ask her to style her wig because she believes she lacks the ability. Laura is a wig stylist, but the last time she made her wig, she fixed a twelve-inch synthetic hair and billed her for a twenty-two-inch human hair. Because Lisa believes Laura is dishonest and unwilling to complete the assignment properly. Nonetheless, Lisa may agree to work with Laura under dire circumstances, such as if no other wig stylist is available.

Imagine Serwaa now knows another wig stylist by the name of Marie. Marie's work on Serwaa had always left her satisfied. Lisa receives a recommendation from Serwaa. Lisa agrees to work with Marie even though she has never met her and was recommended by a trustee friend. Although trust is not transferable, it can be influenced by intermediaries. If there has been no previous direct interaction between the trustor and the trustee, a trustee intermediary can help to establish a link between them for the first time; the link will then be updated by direct interactions, so if Lisa was satisfied with Marie's work, her trust in her will grow; otherwise, it will be declined. This is how word-of-mouth marketing works. Friends whom you can trust can help you complete the work by recommending others. As a result, trust

relationships can be of various kinds and purposes. The authors of [19] define four characteristics of trust relationships:

1. Direct trust: Trust is formed solely through exchanges between the trustor and the trustee, as in the case of Lisa-Serwaa, Lisa-Laura, and Serwaa-Marie relationships.
2. Indirect trust: This occurs when two people are unfamiliar with each other. Trust is established by the use of trustee intermediaries, such as Lisa Marie's relation.
3. Functional trust: The trustor expects the trustee to do the work herself, as in the Lisa Laura and Lisa Marie relationship.
4. Referential trust: The trustor expects the trustee to recommend someone to complete the duty, such as a relation named Lisa Serwaa. It is worth noting that Serwaa's endorsement could potentially be based on her referential faith in someone who knows Marie. In other words, the trustee in a referential trust has no obligation to base her recommendation on a functional trust relationship. A succession of referential trust connections must usually be followed by one functional trust relation [20]. As seen in figure 2, Zanita plays a key role when it comes to referential trust.

Other people (as represented in the diagram as Ewuradjoa) may give Lisa advice as seen in figure 2, where *solid lines signify trust and dashed lines imply distrust*. When it comes to Marie, Lisa has conflicting knowledge. It is a difficult task to deal with this data. Should Lisa, for example, distrust Ewuradjoa's opinion or consider the absolute opposite?

If Lisa is a newcomer to the city and has yet to make friends. For her, the word-of-mouth strategy would be ineffective. Nevertheless, she can rely on Marie's reputation by consulting a specialized magazine dedicated to classifying wig stylists, or by calling the maker of the brand and requesting a qualified wig stylist in her area. We have been thinking of the context as "hair makeover", but we might be more specific by giving details like the length of

the wig, the colour of the wig, and the texture of the wig. If Lisa's wig is substantially different from Serwaa's wig, Marie who was a skilled wig stylist for Serwaa's wig may be less competent for Lisa's. A more exact search can be achieved by refining the context.

Using these previous interactions and the word-of-mouth strategy, the trustor can establish her belief in the trustee in this example. It also demonstrates the method's limitations for new users and how a reputation-based system can provide them with more information if they have no or few friends. Furthermore, it emphasizes the significance of context and the impact of its interpretation.

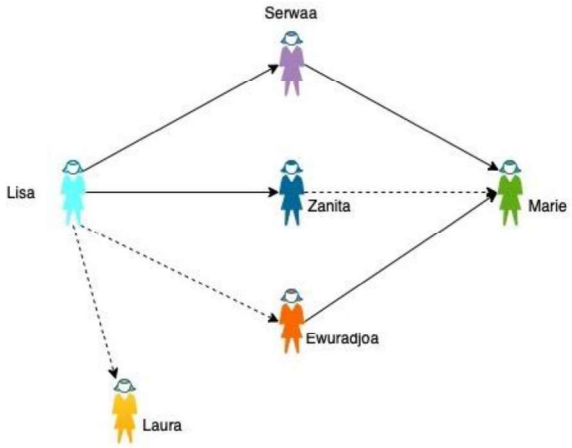


Figure 2. *Network of Trust*

3.4 Data Collection

For this research, a quantitative survey method was used. A google form was utilized to create an English Language questionnaire that was distributed across social media channels like WhatsApp, Telegram and Snapchat. This method assured that everyone in the group was familiar with online social networks. Unlike Wang, who sampled primarily Wei Bo users, this approach is aimed to generate responses from a wide range of social networking users. The questionnaire was meant to assure privacy and confidentiality, although participation was entirely voluntary. The questionnaire came with a brief message explaining the study's purpose. The demographics of relevant respondents as well as their assessments of critical elements that increase trust were gathered. Thus, respondents' perceptions of *social interaction ties, identification, the norm of reciprocity, and trust were scored* using a ten-point 'Likert scale' ranging from strongly distrust (1) to strongly trust (10). All components and question elements were taken from previous studies and adjusted. The convenience sample technique was used to find respondents.

Table 1. *Definition of Concept*

Concept	**Definition**
Social Interaction Ties	It denotes the regularity with which network members engage as well as the intensity of their bonds.
Identification	It refers to a person's sense of a society's inclusion.
Norm of Reciprocity	It refers to people having the disposition to feel obligated to reciprocate when they assume others will do the same for them.

Trust	It refers to the belief that network members will not intentionally damage others and will follow the appropriate standards.

3.5 Data Analysis

300 respondents were identified via convenience sampling from Ashesi University's class of 2022, and the questionnaire was sent to them via various social media sites. Out of 300 respondents identified, 200 replies were received, resulting in a 45 percent response rate. All of the respondents were regular social media users who spent at least 30 minutes per day on the platforms. There was no missing data in the 200 responses because all fields in the questionnaire were required. The male trustors made up 61 percent of the sample, while the females made up the remaining 39 whilst the male trustees made up 54 percent of the sample, while the females made up the remaining 46. Table 2 summarizes the demographics of the respondents.

Demographics of Respondents (N=200)

Table 2. *Demographics of Respondents*

Demographics	Value	Frequency	Percentage
Trustors	Male	122	61%
	Female	78	39%
Trustee	Male	112	56%
	Female	88	44%
Length of Relationship	0-6 months	3	3.3%
	1-2 years	6	6.7%

	3-4 years 5 years and above	26 55	6.1% 28.9%

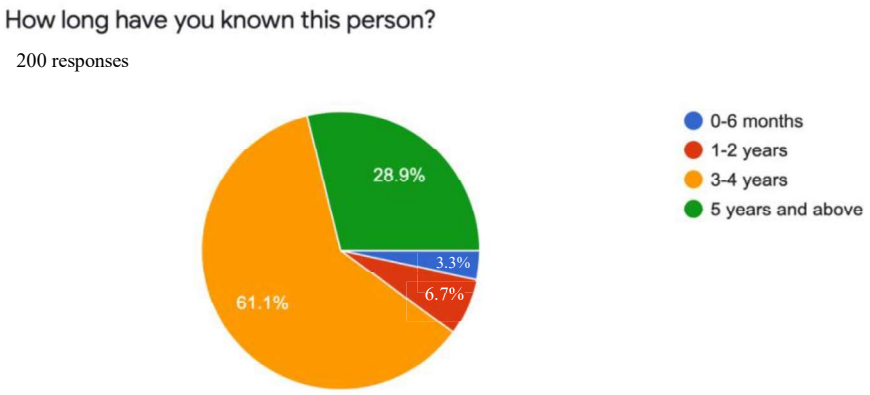


Figure 3. *A pie chart showing how long respondents have known their trustees*

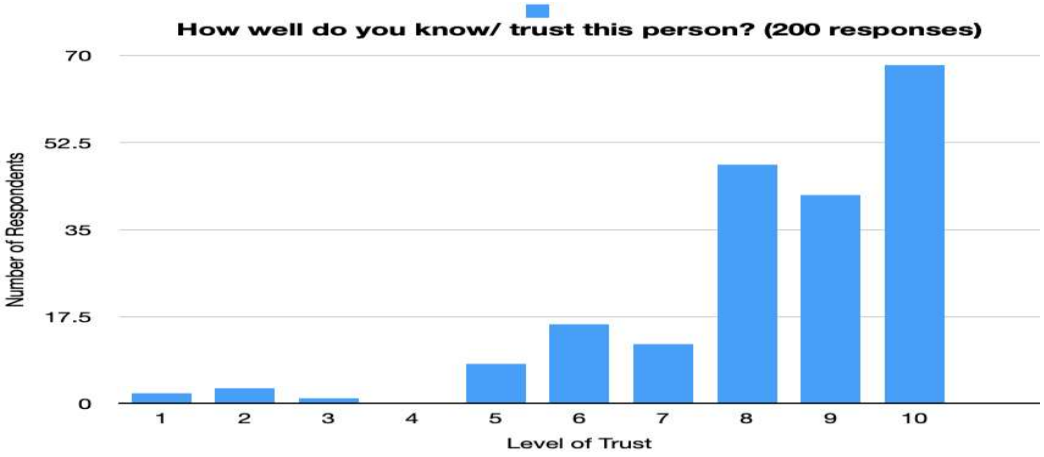


Figure 4. *A bar graph showing the number of respondents against their level of trust on their trustees*

3.6 Research Methodology

Our issue is stated as the following: “Determine the quality assessment about person A for person B considering that each of them is not connected”. In the sense that, finding a trust inference mechanism between these two persons that connects them in the social network, as well as the trust values along the connections to create a recommendation about how much person A and person B should trust each other even though they are not connected directly. Despite this, we determine the number of initial influential that will result in the number of influence nodes in the social network.” This section explains our approach to the problem in depth.

For example, on a site where users contribute product reviews, users might be asked to give a positive trust statement of a user “whose reviews and ratings they have consistently found to be valuable” and a negative trust statement to “reviewers whose reviews they find consistently offensive, inaccurate or in general not valuable”. In this project, trust is described as a real number in the range [0,1], with T (A, B) = 0 which indicates that A has said that his or her level of trust in B is the lowest, that is, this person completely distrusts B. T (C, B) = 1, on the other hand, denotes that C has complete trust in B. Users can obtain varying trust levels from various users, hence trust claims are subjective.

In addition to the above, they are also asymmetric in the sense that just because A trusts B at 0.8 does not mean B has to trust A at 0.8 as well; in contrast, B possibly does not even know A. In most cases, a user has a direct opinion on only a small percentage of other users whilst unknown users make up the rest of the group [21].

It is important to remember that a number of formats for properly expressing and encoding trust claims are beginning to develop. The trust extension of the FOAF4 format proposed in [22] is worth noticing from the Semantic Web community.

Furthermore, as previously said, it is becoming increasingly common for online systems to allow users to express their relationships with other users of the system, even if most of them are now not easily transferrable and only have meaning within the community.

The social network can be created by aggregating all of the trust statements stated by each user, showing the society of users and their trust relationships in a snapshot. Figure 1 shows an example of a simple trust network. As a result of the previously discussed trust features, such a network is a directed, weighted graph with nodes representing users and edges representing trust declarations.

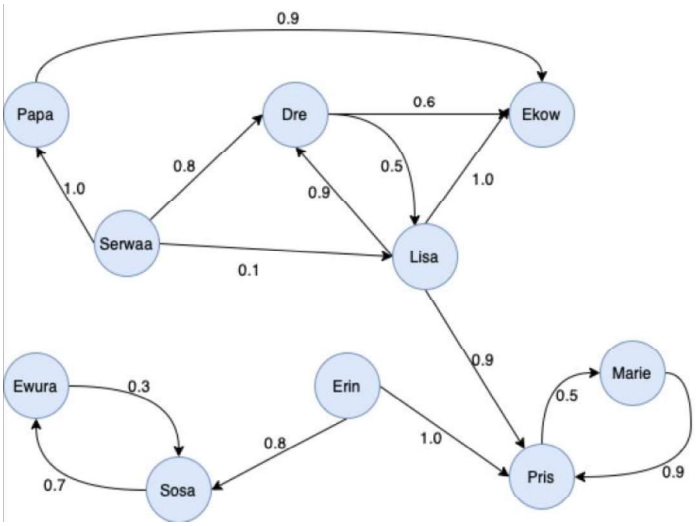


Figure 5. Directed network of trust

a. Defining the directed network's edge-weights.

A weighted-directed graph, G = (U, C, E), can be used to describe a social network, where U is the set of all users in the graph, C is the set of connections and E is the set of

edged weights. An edge from node A to node B in the directed graph denotes that user A "knows" user B in the social network.

Note: Edge weights are defined by the structural properties of the two nodes that form the edge in the network. For example, the edge weight measures the influence of node B on node A whereas the link surrounding the nodes is denoted as;

Recommendation influence = dominance / outdegree

where dominance denotes the degree to which Node B is essential. This however is only if the edge weight is greater than the threshold value, where the threshold is an algorithm parameter and Node B is considered influential whilst Node A is regarded as an influenced node. If this statement is true, then Node B is the parent node for producing recommendations, and Node A is regarded to be in Node B's neighbourhood.

Furthermore, if the edged weight is equal to the threshold value, where the threshold is an algorithm parameter, then person B is influential and person A is an influenced node. If this statement is true, then person B is the node for producing recommendations, and person A is regarded to be in person B's neighbourhood.

On the other hand, to show the cascade effect of this influence spreading from a single node like node B, another node C is added after Node A, with element E from the graph being the edge weight. The symbol for this is Node C -> Node A -> Node B. Two degrees separate Node C and Node B. As a result, if node A is influenced by Node B, we can conclude that Node C is also influenced by Node B. Figure 6 below denotes the relationship between these three nodes.

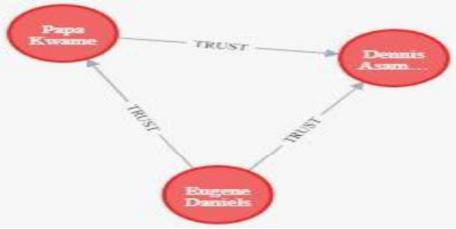


Figure 6. *Relationship between three nodes*

b. Determining whether a node has been influenced by another node.

The threshold which was introduced in the previous approach was to assess if Node A is influenced by Node B. Three threshold criteria are proposed here for determining whether or not a node is influenced:

Criterion I:

No threshold not needed: The first criterion is used for multiple in the graph where there is no influence propagation threshold.

Criterion II:

Using an average threshold: The second criterion uses a network's average edge weights as its threshold. This implies that the threshold is constant for all nodes, depending on the network's overall characteristics. As shown in figure 2 below.

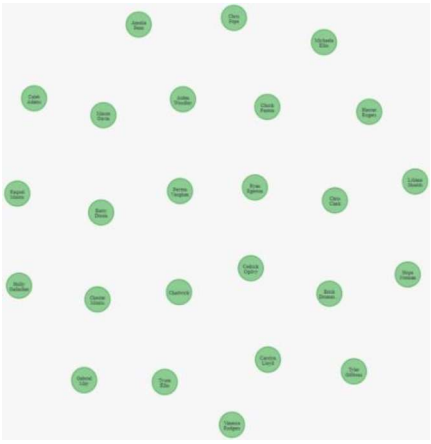


Figure 7. *Network of trust*

Criterion III:

Using edge-weight as a dependent threshold:

The third criterion calculates the threshold by averaging the edge-weights of all the outgoing edges from each node in the graphically displayed social network. As a result, this threshold condition is vertex-specific but universal for all nodes.

3.6 Proposed Models

Our model is designed to determine the most appropriate answer to a given query from a list of options. First of all, we analyze two types of trust-aware models, each of which proposes a different context interpretation. The first relies on local trust metrics, whereas the second one relies on global trust metrics. Nevertheless, both approaches rely on subjective reasoning. Local trust is termed as, trust between two users, and global trust is termed as a user's reputation among the community are the two basic axes that trust modelling follows.

Local trust metrics generate a customized trust score based on the assessing user's point of view. Whilst Global trust metrics assign each user a unique trust score, which is the same regardless of which user is assessing the other's trustworthiness.

In other words, when estimating the trust, a user places in unknown users, local trust metrics take into account the active user's subjective beliefs. As a result, when a user's trust score is projected from the perspective of other users, the result can be different. Instead, global trust metrics calculate a trust score that approximates how much the entire community trusts a single person and is thus independent of the specific active user who is asking "How much should I trust this unknown user?" Even if there is currently no consensus on definitions, this global value is referred to as "reputation," and "reputation systems" are what we referred to as "global trust metrics." For example, the well-known auctions website Ebay.com displays each user's "reputation" by letting users see how many other users have left favourable, neutral, or negative comments about that individual. We can argue that Ebay.com utilizes a worldwide trust metric because this "reputation" number is independent of the surfing user. T: U→[0, 1] is the formal definition of a global trust metric, whereas T: UxU → [0, 1] is the formal definition of a local trust metric. (Where T is defined as trust, with U as users).

Therefore, in general, while local trust metrics can be more precise and tailored to a single user's peculiar views and opinions, they are also computationally more expensive, since they must be computed for every single user whereas global trust metrics just run once for all the community. When it comes to *Controversial Users,* the variations between local and global trust metrics are very noticeable. This is because controversial users are those who are judged by other users in a variety of ways, for example, they are liked by many users but despised by others. It is vital to remember that trust statements are subjective, therefore, "right" by definition. However, it is not acceptable to assume that one is a good user (stating the "right" trust statement) and the other is a bad or harmful user if two users disagree about another user. They simply have opposing but equally valid viewpoints.

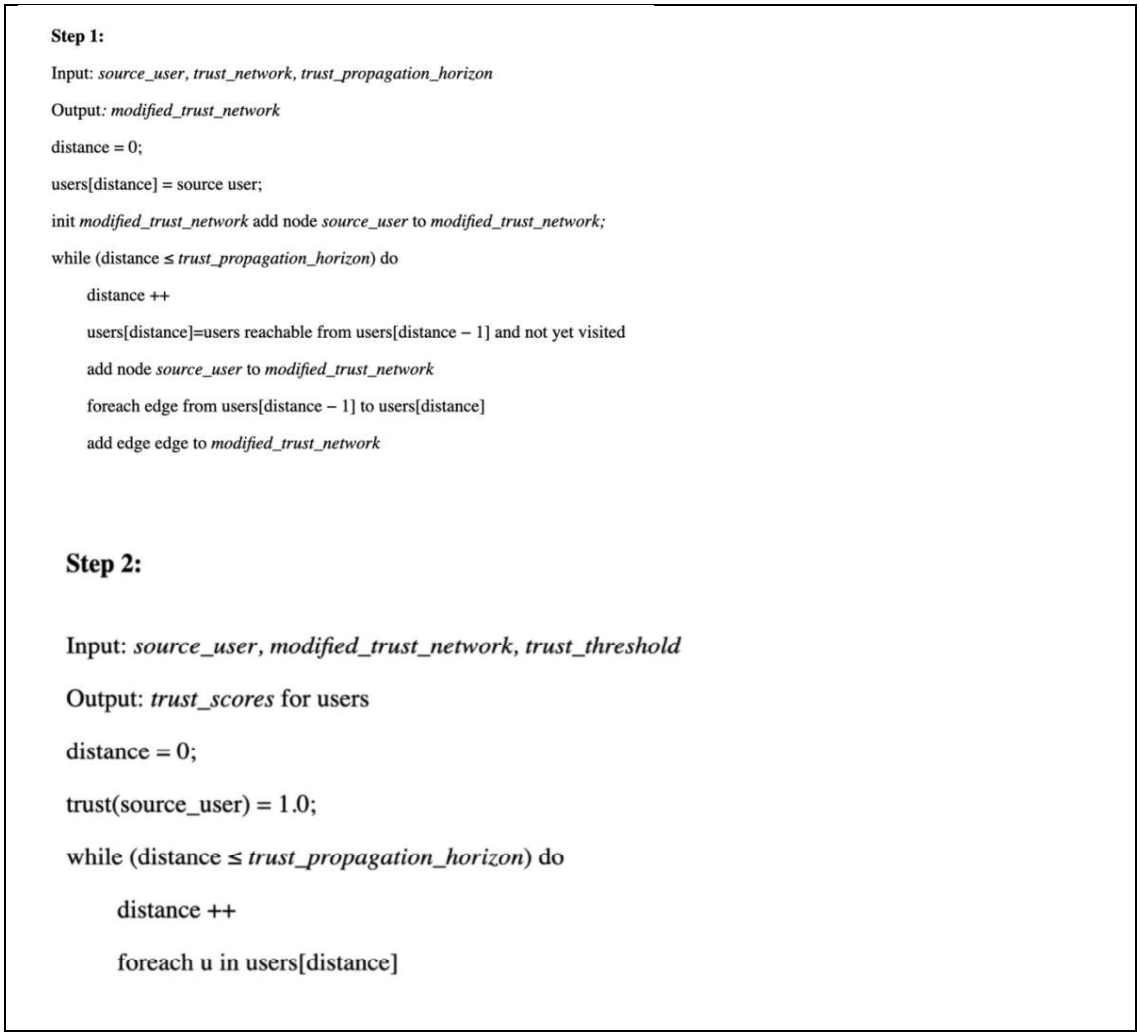
Despite this, we expatiated on these three types of models to have a better understanding and to know the how and why of which metric is suitable for our proposed algorithm. For local metrics, MoleTrust and Multiplicative Strategy for trust propagation are analyzed whilst with global metrics, Eigen Trust is analyzed.

3.6.1 A Local Trust Metric: MoleTrust

This section explains the local trust metric that we utilized. The MoleTrust local trust metric was chosen [21]. The necessity for a time-efficient local trust measure motivated our decision, given the huge number of trust scores to be predicted in the tests. Another reason we chose MoleTrust was that we intended to investigate different levels of locality in the propagation of trust. As a result, the existence of a configurable trust propagation horizon as a MoleTrust input parameter was critical in the selection of this type of local trust metric.

We started by navigating the social network with the source user and propagating trust along trust edges since MoleTrust predicts the source user's trust score on the destination user.

That is, a user's trust score is based on other users' trust statements about the person. Thus, what other users think of the person, is weighted by the trust ratings of the users who issue the trust statements. The concept is that the weight given to a user's viewpoint is proportional to the degree to which that user is regarded as trustworthy. This is because each trust propagation begins with a distinct source user, with the predicted trust score of a certain "user A" which may vary depending on the source user. In this way, the predicted trust score is tailored to the individual. The diagram below illustrates the pseudocode.



Step 1:

Input: *source_user*, *trust_network*, *trust_propagation_horizon*

Output*: modified_trust_network*

distance = 0;

users[distance] = source user;

init *modified_trust_network* add node *source_user* to *modified_trust_network;*

while (distance ≤ *trust_propagation_horizon*) do

distance ++

users[distance]=users reachable from users[distance − 1] and not yet visited

add node *source_user* to *modified_trust_network*

foreach edge from users[distance − 1] to users[distance]

add edge edge to *modified_trust_network*

Step 2:

Input: *source_user*, *modified_trust_network*, *trust_threshold*

Output: *trust_scores* for users

distance = 0;

trust(source_user) = 1.0;

while (distance ≤ *trust_propagation_horizon*) do

distance ++

foreach u in users[distance]

Figure 8*. MoleTrust Pseudocode*

Further, the MoleTrust measure is represented in two steps. The first step is to eliminate cycles from the trust network and convert it into a directed trust graph. Step two walks us through a graph that starts at the source node and ends with the trust score of all visited nodes. Accepting this data, when it comes to step one, the cycles provide an issue since they require several visits to a node during the graph walk, with each time altering the temporary trust value until it converges. It is more efficient to visit each node just once and compute its final trust value. In doing so, the time complexity is proportional to the number of nodes. For this reason, it is important to use a huge number of trust propagations in our test.

Step one is the shortest path distance from the source user is used to sort people. This is because the trust propagation horizon is an important MoleTrust input option. This number determines the maximum distance at which trust may be transmitted from the originating user. The notion behind this is that with each subsequent trust propagation hop, the dependability of the propagated trust falls. Also, this option allows for a reduction in the number of visited users, which results in a reduction in processing time. As a result of this, we can deduce that step one alters the social network by arranging individuals depending on their distance from the source user, retaining only people who are inside the trust propagation horizon. Nonetheless, we are aware that this step eliminates trust statements that can be helpful, but we feel that this is a fair trade-off given the need for time efficiency. The redesigned social network becomes a smaller directed graph after step one, with trust flowing away from the original user and never returning.

On the other hand, step two is a straightforward graph that walks over the changed social network, beginning with the source user. The source user's initial trust score is set at 1. Further, MoleTrust initially calculates the trust score of all users at distance 1, which are users for whom the source user, has made a direct trust declaration. After which the MoleTrust goes

on to users at distance 2, and so on. Note that according to step one, a user's trust score at distance one is solely dependent on the trust scores of users at distance two and above, which have already been computed and determined. This eliminates the need to walk over a user several times, such as that the person's predicted trust scores converge.

In addition to the above, MoleTrust evaluates all incoming trust edges and accepts only those from users with a predicted trust score that is greater or equal to a specific threshold for estimating a user's trust score. Therefore, the average of all approved incoming trust edge values that are weighted by the trust score of the user who made the trust statement, is the projected trust score of a user. As a result, the formula:

$$trust\ (u) = \frac{\Sigma_{\,i \in predecessors}\ (trust(i) * trust\ edge\ (i,u))}{\Sigma_{\,i\,\in\,predecessors}\ (trust\ (i))}$$

Where;

predecessors	users of (i) with a trust edge in users of (u) and a trust of users (i) is greater than the threshold
$\sum$	sum of all predecessors
$\in$	Elements of all predecessors

3.6.2. A Local Trust Metric: Multiplicative Strategy for Trust Propagation

This section explains another local trust metric that we analyzed. The Multiplicative Strategy for Trust Propagation local trust metric was chosen. Despite its simplicity, the multiplicative method has some fascinating characteristics. To begin, if all of the trust values along the trust chain are 1, the propagated trust between the source and destination nodes is also 1. Second, as the number of users along the trust path grows, the propagated trust value

will drop. Third, even if the direct trust values between the next nodes in the path are high, the propagated trust value of the path will decline if the source node does not trust the next node in the chain.

Assume that all of the path's trust values are 1. Therefore, in this situation, the function suggested in this trust value is 1, which reflects the fact that absolute trust exists across the chain. Considering the case where one or more of the trust values on a path is zero (0). That is, one has lost trust in the entity with which it has established a trust relationship. Hence, the function suggested in this trust value is 0. As a result, the suggested value accurately reflects that one does not trust an entity on this path. We consider another path with a length of 3 and a trust value of 0.9 for each of the trust values. The recommended trust value is 0.9 x 0.9 x 0.9 = 0.73. The proposed trust rating is 0.73, despite the fact that each person has a high trust of 0.9 in the recommendation user. The degree of separation between the source user and the target user is reflected in this value. Intuitively, as the distance between the source user and the target user widens, trust decreases.

For example, the path (x_1, x_2, x_3, u) with t (x_1, x_2) = 0.1, t (x_2, x_3) = 0.8, and t (x_3, u) = 0.9. Consequently, 0.1 x 0.8 x 0.9 = 0.07 would be the proposed trust value. Although x_2, and x_3 have very high trust in x_3 and u, respectively, the propagated trust value remains low because x_1 has low trust in x_2.

For this reason, the multiplicative strategy for trust propagation is the local trust measure we employed in our project. However, the global trust measure we used for the comparison is introduced in the following part, which also describes the dataset we used in our testing. This is because the global trust metric we picked was influenced by the properties of the data available.

3.6.3. A Global Trust Metric: Eigen Trust

This section explains the global trust metric that we utilized. The Eigen Trust global trust metric was chosen [23]. The necessity for choosing this type of trust metric was because when it comes to computing trust among nodes in a peer-to-peer network, the Eigen Trust method is a probabilistic graph-based trust algorithm. Hence, it motivated our decision in choosing this type of metric. Also, [24] used this approach in computing global trust levels for nodes in peer-to-peer networks. Besides, mutual transactions between peers are used to compute local trust levels between them. These exchanges might be satisfying or unsatisfying in nature. The local trust value between two people "i" and "j" is defined as Si, j = sat (i, j) – unsat (i, j), where sat (i, j) is the number of satisfactory transactions and unsat (i, j) is the number of unsatisfying transactions.

The domain of these local trust values is unrestricted, and they can even be negative. According to Eigen Trust, they must be standardized to values between 0 and 1. This equation gives the normalized values of trust between nodes "i" and "j":

$$C_{i,j} = \frac{\max(S_{i,j}, 0)}{\sum_{j \in adj\ (i)} \max(S_{i,j}, 0)}$$

Where $S_{i,j}$ is the local trust value between nodes "i" and "j", and $C_{i,j}$ is the normalized local trust value between nodes "i" and "j". All the trust values are from the interval [0, 1], as we can see from the equation above. Interaction-based direct trust values are used as local trust values between nodes in this project's adaption. These numbers have been normalized to have a domain of [0,1].

After adjusting the Eigen Trust algorithm's computation of local trust values, we now adapt its peer-to-peer calculations. That is, the user whose trust value algorithm is searching is

referred to as the *source*, while the person who is being searched for is referred to as the *sink*. The source seeks for the sink's trust value in his neighbors first in the Eigen Trust. Thus, the node's neighbors are his social network buddies.

This technique is repeated till the depth, d, is reached. This is because the algorithm involves the multiplication of trust, where trust values from the interval are required. For this reason, longer pathways to the sink require more multiplications. As a result, it becomes necessary that the value of trust decreases with each successive level, which is achieved by normalizing direct algorithm trust values to the interval [0,1].

Since Eigen Trust is a probabilistic method, this algorithm differs from the preceding two. It also investigates propagation and aggregation as trust qualities, but in a different method than the previous two algorithms. This method differs from others because it calculates local trust values from interactions rather than experiences. In addition, in a peer-to-peer network, the Eigen Trust algorithm produces a global trust value for each user, whereas this adaption calculates trust values between two users in a social network.

3.7 Algorithm

This section explains the algorithm that we utilized. The ***Dijkstra Algorithm*** was chosen as our algorithm. The necessity to discover the shortest path between any two network vertices motivated our decision, given the variety of algorithms available. In the sense that, if any user wants to know the reliability of another user in an online social network, our goal is to find the most trusted path in the shortest amount of time possible.

Nonetheless, because the shortest distance between two vertices may not include all of the graph's vertices, it differs from the least set of vertices. Also, we keep two sets: one which contains vertices that are already in the shortest path tree, and the other which contains vertices that are not yet in the shortest path tree. Vertex which is in the other set (a set that is not yet included) is the shortest distance from the source at each stage of the procedure is found. On the other hand, negative weights will lead this algorithm to provide inaccurate results.

The steps used in the Dijkstra algorithm for finding the shortest path from a single source vertex to all other vertices in a graph are listed below.

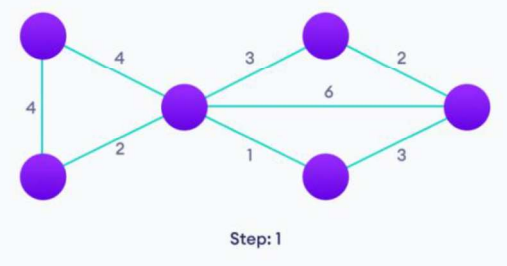
1. Make a set called the shortest path tree set (sptSet) to keep track of the vertices in the shortest path tree. That is, those whose minimal distance from the source is calculated and completed. This set is initially empty.
2. Assign a distance value to each of the input graph's vertices. All distance values should be infinite. Assign the source vertex a distance value of 0 to ensure that it is chosen first.
3. Because the shortest path tree set does not contain all vertices.
    a. Choose a vertex "u" that is not in the shortest path tree set and has a minimal distance value.

b. Add “u” to the shortest path tree set

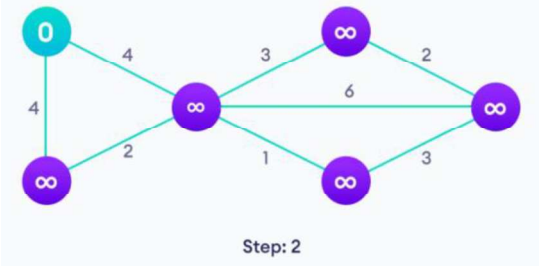
c. Update the distance between all of “u’s” adjacent vertices. Then iterate through all nearby vertices to update the distance values. If the total of the distance value of u (from source) and the weight of edge u-v is less than the distance value of v for each neighbouring vertex “v”, then update the distance of “v”.

An example is provided to help better understand this algorithm.

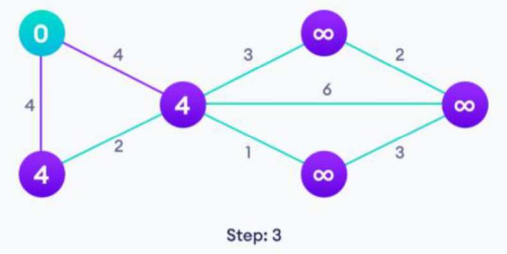
Step 1: create a weighted graph.



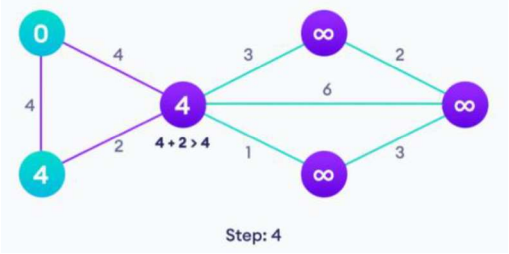
Step 2: pick a starting vertex and give all other devices infinity path values.



Step 3: Update the path length of each vertex.

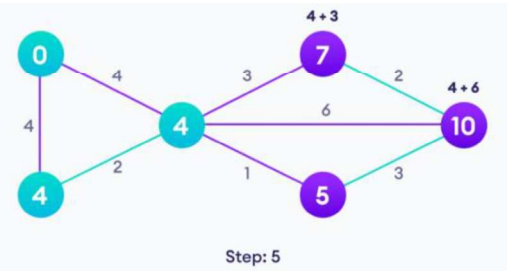


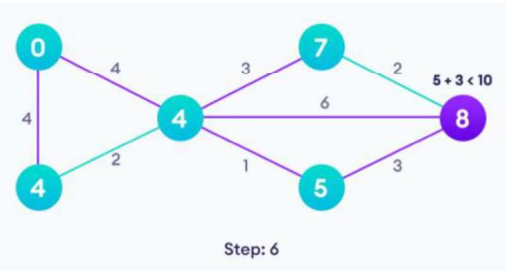
Step 4: if the path length of the neighbouring vertex is less than the new path length, do not update it.



Step 5: Do not update the path lengths of vertices you have previously visited.

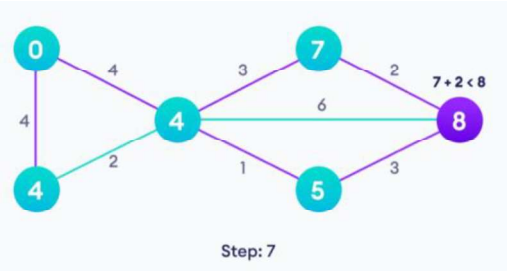
Step 6: Choose the unvisited vertex with the shortest path length after each iteration. Hence, 5 is chosen before 7.

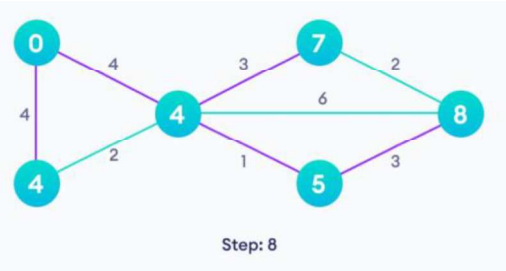




Step 7: Notice how the path length of the rightmost vertex is updated twice

Step 8: Repeat until all of the vertices have been visited.





We utilize the same terms and concepts as before and characterize the social network as a weighted-directed graph, where an edge from node A to node B denotes that user A "knows" user B in the social network, and the edged weight denotes user B's impact on user A. Our goal is to find the important nodes that influence the remaining nodes.

3.7.1 Procedure of Algorithm

To discover the trust path in our algorithm, we used an approach in which we chose the edge with the highest trust ratings are each level rather than the lowest, as said in Dijkstra's algorithm above. All of the trust rates in the trust dataset are replaced with their

reciprocals to achieve this. After this, we apply Dijkstra's method to discover the shortest path, which gives us the most reliable path.

The goal is to increase the smaller trust value while decreasing the larger value. Considering the following scenario with a set of three ratings: 3, 6, and 8. As a result of this, the highest rating 8 now becomes the smallest trust rating 0.1, and the lowest value 3 is now the largest rating 0.3. the generated matrix is readily applied to Dijkstra's shortest path algorithm. The most trusted path will then be chosen in this scenario, rather than the minimal weight path.

For this reason, the propagated trust value from the source node "s" to the target node "t" for the set of selected nodes in the shortest and most trusted path "P", is the average of the trust ratings from each node in "P" weighted by the propagative distance "d_t" from the source node to each node "i" in the path "P". As shown in the formula below:

$$propagated\ trust\ value = \sum(\frac{d_t\ x\ trust_{i,j}}{d_t})$$

Where "i" and "j" are neighbours that are directed connected.

3.8 System Architecture

The system will be hosted on a Google Cloud Platform web server as a web application.

3.8.1. Application Architecture

A 3-tier application architecture is used in this project, and it comprises a presentation tier, an application tier, and a data tier in a flexible client-server architecture. The data tier would store data, the application tier will handle the functionality and the presentation tier will be the graphical user interface that connects the two other tiers. These three tiers are conceptual rather than physical.

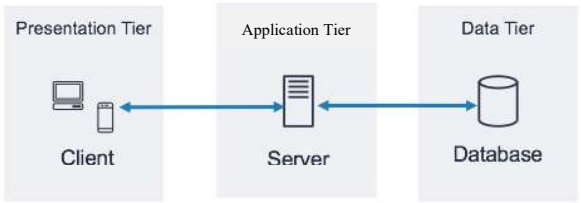


Figure 9. 3-Tier Architecture

Presentation Layer: This layer is the application's frontend layer and the interface in which end-users will interact directly. The presentation layer will connect with other tiers by transmitting results to the browser and is based on web development frameworks like HTML, CSS, Bootstrap, PHP, and JavaScript.

Application Layer: This layer will be responsible for the codes which will be written in programming languages like Python, and JavaScript, and will govern the application's basic functionality by conducting extensive processing.

Data Layer: In this layer, the database server is located here, and this is where information will be stored and retrieved. The data in this layer will be handled and accessible using Neo4J.

Improved horizontal scalability, speed, and availability are the main reasons why we will be employing a 3-tier design. The 3-tier model makes it easier for us to continuously expand on the application as new insights arise since the programming for one layer may be altered without affecting the other layers.

3.8.2. Functional Requirement

This section explains the functional requirements used in building the application. The following examples below show the product features that we used in building the application in order for the users to achieve their purpose. It specifies how the underlying system behaves in certain situations.

- [FR01] - Registration: user needs to sign up before he or she can access the application
- [FR02] - User profile: to make a profile, edit the profile, to make a friend list.
- [FR03] - A user can search for a specific person within a wide range of nodes.
- [FR04] - A trustor is spoilt with the option of 1-10 when inferring trust onto a trustee. (Where 1= strongly distrust and 10= strongly trust)
- [FR05] - The application shall show the time and date of a recommendation is/ was made.
- [FR06] - Preview of recommendations of someone in the social network.

3.8.3. Non-Functional Requirement

This section explains the non-functional requirements used in building the application. These requirements show the quality attributes that specify how our application should behave.

The following are examples of the basic non-functional requirements that were used to carry out this application:

- [NFR01] - Scalability: This describes the evaluation of the system's ability to handle the highest workloads while still meeting performance requirements.
- [NFR02] - Security: This describes how the system and its data are being secured against cyber-attacks.
- [NFR03] - Performance: This describes the way people engage with our application in various contexts. Therefore, if the system provides poor performance, it might result in an unpleasant user experience and well also put the system's security at risk.
- [NFR04] - Capacity: This quality feature allows us to know how to scale up our system to meet rising volume demands.
- [NFR05] - Reliability: This quality attribute shows the likelihood the system will operate without failure for a certain amount of time under specified conditions. In the sense that, it allows us to know the amount of time provided to users in the event of a downtime.
- [NFR06] - Usability: This requirement describes the degree of ease at which a user interacts with our application to attain the desired goals effectively and efficiently.

3.8.4. Flowchart

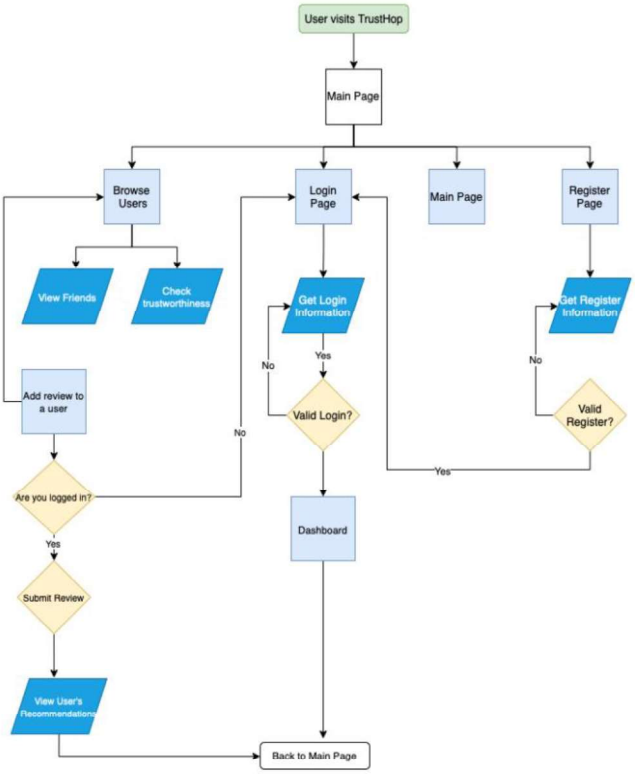


***Figure 10**. Flowchart*

Chapter 4: Experiment, Implementation, and Results

In our tests, to infer propagating trust in social networks, our system uses explicit trust ratings. We compared the performance of alternative trust evaluation models using three extensively used, a real-life survey as an actual dataset, as well as Advogato datasets. Advogato is the dataset that is an online social network for open-source engineers. This network offers four degrees of trustworthiness to allow members to certify one other. Observer, Apprentice, Journeyer, and Master are the four types of trustworthiness.

To infer propagating trust in social networks, our system uses explicit trust ratings. For our studies, we used a real-life survey as an actual dataset, as well as one publicly available real-world dataset.

4.1 Datasets used and their descriptions:

4.1.1. Data from a real-world survey

We ran a poll intending to generate a real-world trust dataset to evaluate the accuracy of our algorithm. A questionnaire was originally devised and evaluated in an undergraduate Senior class at Ashesi University, with 200 students participating.

On a scale of 1 to 10, students were asked to rank their close friends (least trusted to most trusted). The information gathered from this survey was utilized to analyze the algorithm. The findings imply that the most popular individual in class may not be the most trustworthy person in the class, and vice versa. Indeed, a student “i” might recommend “j” as a trustworthy buddy without being recommended by “j”. As a result, calculating trust based on the number of trust ratings is pointless because trust is personal and based on an individual's opinion.

4.1.2. Real-World Dataset

Advogato: We utilized Advogato, an online community for open-source software engineers, as the core data set for our investigation. The site's users assign a level of trust to one another. Master, journeyer, apprentice, and observer are the trust value preferences, with the master being the highest degree in each category. Because it is possible to trust oneself on Advogato, there are self-loops in the dataset; however, we eliminate them because they do not fit our model. We replace the following values for its four trust values: master = 10, journeyer = 6.6, apprentice = 3.3, and observer = 1. The outcome of these ratings among members is a rich network of trust, with 6,551 users and 47,337 trust ratings after the self-loops have been removed.

4.2 Results and Findings

4.2.1. Methods of Validation of Results and Exceptions

1. Assuring that the validation delivers enough results on time to review and evaluate if the initial set of queries needs to be modified. Research must be speedy in early case analysis, but results must be sorted by relevance so that search results may be promptly analyzed and iterated.
2. Allowing for a comparison of different algorithm methods. Our method of validation helps us to take note of certain points such that when evaluating different algorithm technologies, it is crucial to keep in mind that each one is likely to return different results. Therefore, to analyze the success of an algorithm, one must consider the context of the query with the research.

Also, there is a high threshold of accuracy when searching meta-data using fielded queries and it is necessary to understand variances in field attributes, naming patterns, and syntax.

4.2 Model Experimentation

Using the same dataset, we compared our technique against a multiplicative approach and trust propagation. In comparing the methods, we used the "leave-one-out" strategy, which means that for each direct trust link in the Trust Graph, we first remove the link, then compute the propagated trust value using our methodology between the corresponding nodes, and then restore the connecting link.

The following is the testing procedure:

1. For each edge in the Trust matrix, G between the source and destination users, S and D.
2. Direct trust = matrix G's weight (S, D)
3. Using a modified Dijkstra method, find the shortest and most reliable path between S and D and store it in an array list called PATH.
4. Using our Trust Propagation Algorithm, calculate the Propagated Trust between S and D
5. Trust propagation Algorithm=Propagated Trust (S, D) (PATH, S, D)
6. Compare and contrast S and T's direct and propagated trust.
7. Finally, the results of the trust network execution with estimated local trust were saved in Microsoft Excel files for further investigation.

4.3 Results

To give a sufficient and fair analysis, we compared our method to the iterative multiplicative approach and trust propagation in our study. In our tests, we found that our approach outperforms both of these algorithms. For trust calculation, we use the most trusted and shortest path in our approach, but in [24], the least trusted component, although being the shortest path, is used. This is the primary advantage of our method the iterative multiplicative strategy, as we know that a chain of highly trusted individuals is more reliable than a network of low-trust individuals. Furthermore, our results show that direct and propagated trust obtained using our algorithm has a considerably strong positive linear correlation of 0.32 (rounded up to two decimal places), whereas the iterative multiplicative strategy has a 0.38 for the Advogato dataset [19]. It should be highlighted that in the experiment, the values of direct trust and spread trust are earned independently of one another. Because the data set employed represents a real and vast web of trust, this result is significant. The figure shows a scatter plot of the direct trust values and the related propagated trust values.

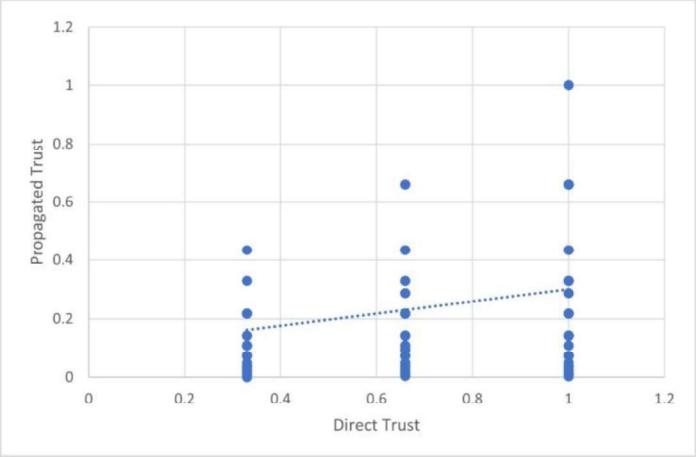


***Figure 11**. A scatter plot showing direct trust values and propagated trust values*

4.4 Analysis

It can be observed that in the experiment, the values of directed trust and propagated trust are earned independently of one another. This is because the data set employed represents a genuine and vase network of trust. For the dataset utilized, we used the Mean Absolute Percentage Error as our assessment metric to quantify the degree of departure of inferred trust values from direct trust values. In statistics, the mean absolute percentage error is a measure of prediction accuracy. It is commonly expressed as a percentage, and it is calculated using the formula:

$$\frac{1}{n} \sum_{i,j=1 \mid i!=j} \left| \frac{DT_{i,j} - PT_{i,j}}{DT_{i,j}} \right.$$

Where 'n' represents the total number of users in the trust graph, $DT_{i,j}$ represents direct trust, and $PT_{i,j}$ represents predicted trust between any two users 'i' and 'j'.

***Table 3**. A table showing the mean absolute percentage error*

Dataset	Number of users	Trust propagation	Multiplicative Strategy
Advogato	300	0.208141556	0.483397908

The description of the dataset is specified in section 4.1 and the outcome is shown in the table above. According to the findings, our approach beats the multiplicative strategy. As can be seen, the mean absolute percentage error is calculated to be substantially lower for our technique than for the dataset utilized. In other words, our algorithm is used to choose the path; however, the only difference between the two approaches is the mechanism for calculating trust.

The relationship between direct trust and propagated trust in this method is investigated. The popular Pearson correlation coefficient is used in this project, which provides a measure of the linear connection between two variables A and B in the range [-1, +1]. The correlation is produced using the same dataset to perform a comparison between these approaches. The outcome is shown in the table below. For large graphs, this approach produces a significant positive correlation between direct and propagated trust ratings; however, this algorithm fails to produce good results for extremely short graphs.

***Table 4**. A table showing the correlation between direct trust and propagated trust*

Dataset	Number of users	Trust propagation	Multiplicative Strategy
Advogato	300	0.361424776	0.263317734

4.5 Implementation of Application Used for the Experiment

For our project, our validation developed a system where users can hop onto the site and input a rating system or input a recommendation for a business or company. This information is then stored in the database, meaning all data inputted by users cannot be tampered with unless by administrators. Therefore, once the user visits the site, only credible information is displayed or shown to users. The client-server model approach was used to create the application, which involves a division between the clients and the servers. The following briefly describes the development of the website.

Frontend Development

This part of the web development focuses on what the users view on their end. It entails converting the backend codes into a graphical interface as well as ensuring that data is

displayed in an easy-to-read and understand style. Throughout the frontend development, it was ensured that the web application is accessible on various devious devices, through the consideration of the assortment of screen sizes and operating systems.

Another factor that was considered was the web browser preferences of users. Hence, this website is made compatible with every browser, such as Safari, Chrome, Firefox, and others.

On the other hand, during the frontend development, Hypertext Markup language (HTML), CSS and JavaScript were used to turn the coding data into a user-friendly interface. The three technologies are described briefly below.

HTML: was used for describing and marking up so that the browser can display it correctly.

CSS: Aids in managing the formatting, presentation, and layout of the website

JavaScript: Used to help with altering the content of the webpage in response to the user's action

Finally, wireframes, that is, the basic drawings of the user flow, prototypes (functioning examples of the site), and user testing were all part of the frontend development process.

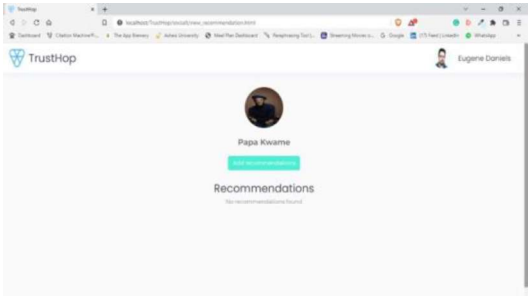


Figure 12. *Result after searching for a trustee*

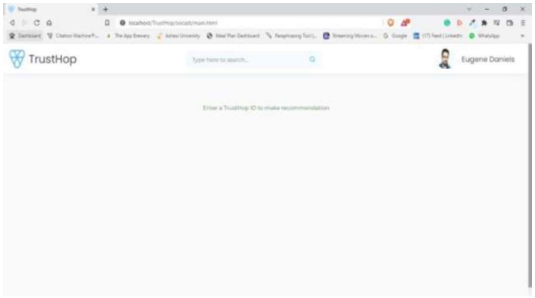
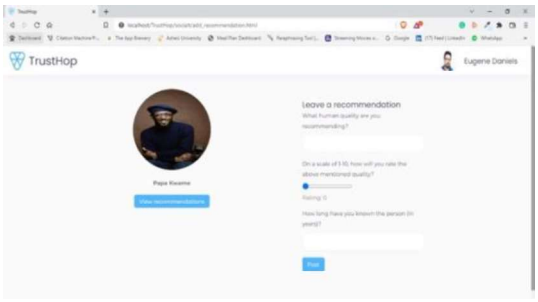


Figure 13. *Searching for a trustee*



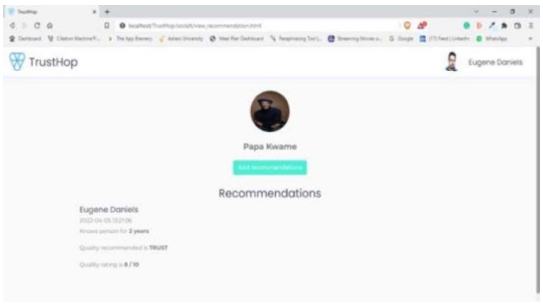


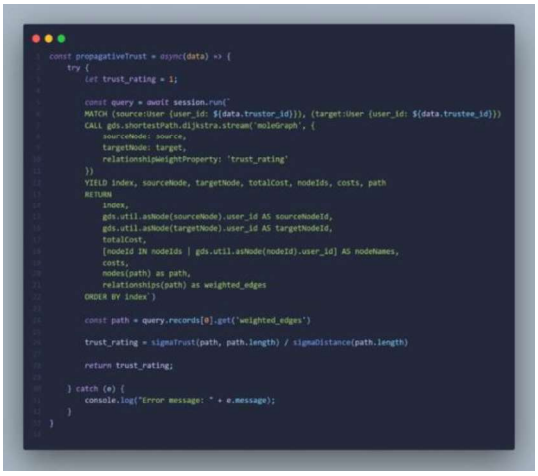
Figure 14. *Adding a recommendation*

Figure 15. *Viewing a recommendation*

Backend Development

This part of the web development focuses on how the website functions, and the behind-the-scenes of the frontend. Throughout this development, it was ensured that our end users receive the data or services they requested in a timely and error-free manner. As a result of this, the backend development necessitates a diverse set of programming abilities and knowledge.

Servers, a database, and an application programming interface are the three devices in which the backend development was done. PHP, Node JS, Neo4J graph database, and Java are the programming languages that this system runs on.



```
const propagativeTrust = async(data) => {
    try {
        let trust_rating = 1;

        const query = await session.run(`
        MATCH (source:User {user_id: ${data.trustor_id}}), (target:User {user_id: ${data.trustee_id}})
        CALL gds.shortestPath.dijkstra.stream('moleGraph', {
            sourceNode: source,
            targetNode: target,
            relationshipWeightProperty: 'trust_rating'
        })
        YIELD index, sourceNode, targetNode, totalCost, nodeIds, costs, path
        RETURN
            index,
            gds.util.asNode(sourceNode).user_id AS sourceNodeId,
            gds.util.asNode(targetNode).user_id AS targetNodeId,
            totalCost,
            [nodeId IN nodeIds | gds.util.asNode(nodeId).user_id] AS nodeNames,
            costs,
            nodes(path) as path,
            relationships(path) as weighted_edges
        ORDER BY index`)

        const path = query.records[0].get('weighted_edges')

        trust_rating = sigmaTrust(path, path.length) / sigmaDistance(path.length)

        return trust_rating;

    } catch (e) {
        console.log("Error message: " + e.message);
    }
}
```

```
const multiplicativeStrategy = async(data) => {
    try {
        let trust_value = 1;
        const nameGraph = await session.run(`
        MATCH (source:User {user_id: ${data.trustor_id}}), (target:User {user_id: ${data.trustee_id}})
        CALL gds.shortestPath.dijkstra.stream('multiplicativeGraph', {
            sourceNode: source,
            targetNode: target,
            relationshipWeightProperty: 'trust_rating'
        })
        YIELD index, sourceNode, targetNode, totalCost, nodeIds, costs, path
        RETURN
            index,
            gds.util.asNode(sourceNode).user_id AS sourceNodeId,
            gds.util.asNode(targetNode).user_id AS targetNodeId,
            totalCost,
            [nodeId IN nodeIds | gds.util.asNode(nodeId).user_id] AS nodeNames,
            costs,
            nodes(path) as path
        ORDER BY index`)

        const costs = nameGraph.records[0].get('costs')

        trust_value = costs[1]

        for (let i = 1; i < costs.length; i++) {

            if (costs[i + 1] != undefined) {
                let node_rating = costs[i]
                let neighbor_rating = costs[i + 1]
                let trust_rating = neighbor_rating - node_rating
                trust_value = trust_value * trust_rating
            }
        }
        return trust_value;

    } catch (e) {
        console.log("Error message: " + e.message);
    }
}
```

Figure 16. *Code that implements the trust propagation*

Figure 17. *Code that implements the multiplicative strategy*

4.6 Layers of Implementation

Suppose a request from a user to obtain information about a specific person to demonstrate how the level architecture works. The request flows down to the database to obtain the user's data and the response flows back up to the screen to display the recommendation data, as shown by the black arrows. A trustee's data and recommendation reviews data are combined in this scenario.

Accepting the request and displaying the user's information is the responsibility of the user screen. This has no idea where the data is stored, how it is accessed, or how many database tables must be queried to obtain the information. When the user screen receives a quest for a piece of information for a specific person, the request is forwarded to the user delegate module. This module is in charge of determining which business layer modules can handle the request, as well as how to get to that module and what data it requires.

In the business layer, the user object is in charge of gathering all of the data required by the business request (in this case to get the trustee's information). This module uses the persistence layer's user data access object module to collect the person's data, as well as the request data access object to get recommendation reviews. These modules then execute the database operations to retrieve the necessary data and return it to the business layer's user object. When the user object receives the data, it aggregates it and sends it back to the user delegate, who then sends it to the user screen, where it is displayed to the user.

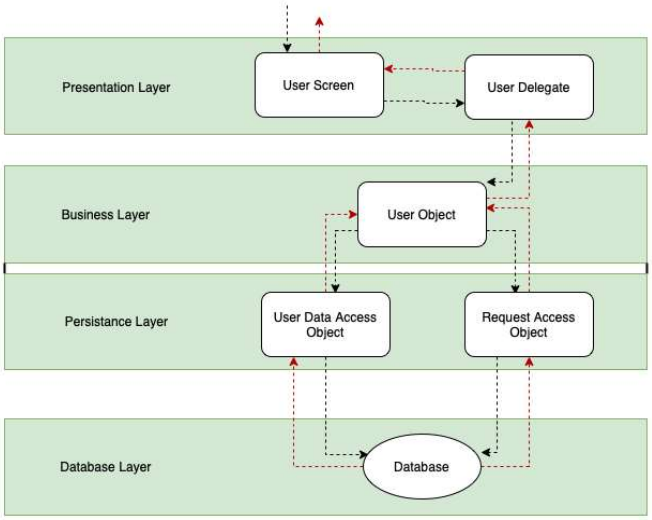


***Figure 18**. Layer of Implementation*

4.7 Risk Management

In this project, failure in our development refers to a negative influence on the project, which might take the form of worse end-product quality, delayed completion, or outright project failure. The process in which we will be managing our risk is identifying, assessing, and prioritizing risks. There are two main types of risk management we employed in this project; risk assessment and risk reporting, which includes risk identification, analysis, and prioritization. The risk management we employed required six steps: requirement phase, development phase, and test phase.

- Requirement Phase: Users' requirements are acquired during the requirement phase. As a result, risks are recognized and evaluated here. During this phase, two processes take place:
  - Asset identification: Assessment of the possibility of certain disruptions occurring and the procedures in place to reduce the organization's exposure to such risk.
  - Threat identification: During the requirement phase, this method is utilized to detect threats. To identify risks and guide subsequent design, coding, and testing decisions, a

threat analysis approach is utilized. This is because many systems have unique characteristics, identifying security threats is a planned process that needs some imagination.

- Development phase: During this phase, our major purpose is to create the solution components' code as well as documentation. Throughout this phase, we continue to identify all risks and address new ones as they arise. As a team, we utilized two steps to this procedure:
  - Code reviews: This procedure is to determine whether the code matches local standards, and it may even lead to the discovery of some issues before compiling, which could prompt future concerns.
  - Unit testing and static testing: The team will check the security functionality of components as well as verify that the countermeasures being built minimize any security risks previously discovered through threat modelling and source code analysis by using unit tests and dynamic analysis.
- Deployment phase: During this phase, our project is only partially finished. Throughout this phase, all risks in the entire life cycle are recognized, and a proper test plan is created.
  - Periodic testing: This refers to third-party testing that will be performed on our project that is still being developed.
  - Risk management strategy: Creating a risk management plan will include creating a risk categorization table, ranking the risk, preparing, and sorting the risk table, and ensuring that risk management is a continuous process throughout the project.

Chapter 5: Conclusion and Recommendations

In today's online social networks, finding trustworthy people with whom to form relationships is a top priority. To overcome these challenges, the concept of "trust" is crucial in social networking platforms. Despite this, the purpose of this thesis is to know how to compute a quality assessment of one person for another considering both individuals are not connected. Therefore, we presented a trust propagation technique in this paper. When it comes to online social networks, finding the best and most dependable trust path is always a challenge. The length of trust pathways and varied measuring methodologies that select how to unite diverse information sources affect the accuracy of trust propagation predictions. For trust inference, this technique uses the most trusted as well as the shortest path. Then, to determine trust for the chosen path, the average of trust values is weighted by the propagation distance. This technique is compared to the Multiplicative Strategy proposed as well as the Eigen Metric. We show that this method outperforms the other alternative through experimental evaluation. The Advogato dataset, which had over 47,337 trust linkages, was principally utilized. Finally, the experimental research findings were presented and interpreted.

Moreover, we evaluated local and global trust metrics in the task of predicting trust scores of unknown users by assessing the variations in accuracy and coverage of two representative examples of these trust metrics. We focused our attention on individuals, who are defined as users (trustees) who are judged in a variety of ways by other users (trustors) and we found that these individuals make us a sizable fraction of the dataset. We denoted that global trust metrics have an inherent constraint on these users and that local trust metrics are more suited in circumstances where opinions are subjective. The empirical findings show that the chosen local trust metric can greatly reduce prediction error while maintaining a high level of trust. The evidence offered in this study is pertinent to the consideration of the different types

of societies caused by different trust metrics, such as the impact of global trust metrics as well as the local trust metrics. As a result, we concluded by examining the dangers that both of these metrics pose.

5.1 Limitations and Recommendations

As previously stated, this project has intriguing implications, nonetheless, it is not without flaws. In terms of data collection, we used convenience sampling to gather responses. The following are the issues:

First, data was collected in a specific school, Ashesi. Secondly, the sample distribution was heavily weighted towards the Class of 2022. Because of these two constraints, we are unable to generalize our findings. As a result, in future studies, we should broaden the sample's coverage. To incorporate respondents with more different demographic traits, therefore, the conclusions must take into account users from various education levels and multiple services.

Also, we would like to improve the implementation in the future to increase throughput. The Dijkstra algorithm is used to determine the shortest path, which will locate just one shortest path at a time, however, there may be more than one. In the future, we plan to use stochastic optimization techniques to overcome this challenge (thus, when unpredictability is involved, stochastic optimization is defined as a group of strategies for minimizing or maximizing an objective function. Another option is to integrate content-related elements in order to determine the criteria for trustworthy and untrustworthy behaviour. Although it would be more in the realm of artificial intelligence, it would still be a fascinating extension.

References

[1] Avni Gulati and Magdalini Eirinaki. 2018. Influence Propagation for Social Graph-based Recommendations. In *2018 IEEE International Conference on Big Data (Big Data)*,2180–2189. DOI:https://doi.org/10.1109/BigData.2018.8622213

[2] Cristiano Castelfranchi, Rino Falcone, and Francesca Marzo. 2006. Being Trusted in a Social Network: Trust as Relational Capital. *Trust Management*, Springer, Berlin, Heidelberg, 19–32. DOI:https://doi.org/10.1007/11755593_3

[3] Gunnar Lind Haase Svendsen, Gert Tinggaard Svendsen, and Peter Graeff. 2012. Explaining the Emergence of Social Trust: Denmark and Germany. *Historical Social Research / Historische Sozialforschung* 37, 3 (141) (2012), 351–367.

[4] Julius Mansa. 2021. Understanding Social Networking. *Investopedia*. Retrieved October 18, 2021 from https://www.investopedia.com/terms/s/social-networking.asp

[5] Mihnea C. Moldoveanu and Joel A. C. Baum. 2011. "I Think You Think I Think You're Lying": The Interactive Epistemology of Trust in Social Networks. *Management Science* 57, 2 (2011), 393–412.

[6] Raquel Ureña, Francisco Chiclana, and Enrique Herrera-Viedma. 2020. DeciTrustNET: A graph based trust and reputation framework for social networks. *Information Fusion* 61, (2020), 101–112. DOI:https://doi.org/10.1016/j.inffus.2020.03.006

[7] Sonja Grabner-Kräuter and Sofie Bitter. 2015. Trust in online social networks: A multifaceted perspective. *Forum for Social Economics* 44, 1 (January 2015), 48–68. DOI:https://doi.org/10.1080/07360932.2013.781517

[8] Su Rong Yan, Xiao Lin Zheng, Yan Wang, William Wei Song, and Wen Yu Zhang. 2015. A graph-based comprehensive reputation model: Exploiting the social context of opinions to enhance trust in social commerce. *Information Sciences* 318, (October 2015), 51–72. DOI:https://doi.org/10.1016/j.ins.2014.09.036

[9] Surya Nepal, Cécile Paris, Sanat Bista, and Wanita Sherchan. 2013. A Trust Model Based Analysis of Social Networks. *Int. J. of Trust Management in Computing and Communications* 1, (November 2013). DOI:https://doi.org/10.1504/IJTMCC.2013.052522

[10] Wanita Sherchan, Surya Nepal, and Cecile Paris. 2013. A survey of trust in social networks. *ACM Comput. Surv.* 45, 4 (August 2013), 1–33. DOI:https://doi.org/10.1145/2501654.2501661

[11] Wanyu Lin, Zhaolin Gao, and Baochun Li. 2020. Guardian: Evaluating Trust in Online Social Networks with Graph Convolutional Networks. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 914–923. DOI:https://doi.org/10.1109/INFOCOM41043.2020.9155370

[12] Wenjun Jiang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Jie Wu. 2016. Understanding Graph-Based Trust Evaluation in Online Social Networks: Methodologies and Challenges: ACM Computing Surveys: Vol 49, No 1. *ACM Computing Surveys*. Retrieved from https://dl.acm.org/doi/10.1145/2906151

[13] Yu Wang. 2017. Antecedents of Social Network Trust in SNS Usage: The Moderating Role of Offline Familiarity. *Social Networking*. Retrieved from https://www.scirp.org/journal/paperinformation.aspx?paperid=75391

[14] Hsu, M., Chang, C. and Yen, C. Exploring the antecedents of trust in virtual communities. *Behaviour & Information Technology 30*, 5 (2011), 587-601.

[15] Deutsch, M. Trust and suspicion. *Journal of Conflict Resolution 2*, 4 (1958), 265-279.

[16] Corritore, C.L., Kracher, B. and Wiedenbeck, S. (2003) On-Line Trust: Concepts, Evolving Themes, a Model. International Journal of Human-Computer Studies, 58, 737-758.

[17] Josang, R. Ismail, and C. Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43, 2 (2007), 618–644.

[18] McKnight, D.H. and Chervany, N.L. (2001) Conceptualizing Trust: A Typology and E-Commerce Customer Relationships Model. Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Maui, HI, 6 January 2001, 10.

[19] Josang, A., R. Hayward, and S. Pope.Trust Network Analy- sis with Subjective Logic. In Proceedings of the 29th Aus- tralasian Computer Science Conference-Volume 48, 8594, 2006. http://dl.acm.org/citation.cfm?id=1151710.

[20] Josang, A., and S. Pope. Semantic Constraints for Trust Transitivity.In Proceedings of the 2nd Asia-Pacific Conference on Conceptual[32] modelling-Volume 43, 5968, 2005.

[21] Massa, P. and Avesani, P., 2007. Trust Metrics on Controversial Users. *International Journal on Semantic Web and Information Systems,* 3(1), pp. 39-64.

[22] Jennifer Golbeck. Computing and Applying Trust in Web-based Social Net- works. PhD thesis, University of Maryland, 2005.

[23] Šitum, M., 2014. *Analysis of Algorithm for Determining Trust among Friends on Social Networks*. Retrieved from https://www.fer.unizg.hr/_download/repository/Master_Thesis_-_Mirjam_Situm.pdf

[24] Bhattacharya, M. and Nesa, Na. An Algorithm for Predicting Local Trust based on Trust Propagation in Online Social Networks. *International Journal of Computer Applications 156*, 7 (2016), 8-15.