Fuzzy Logic based Recommendation System: Crafts to Clients Suggestion

1st Saber Modallal
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine, P.O.Box:7
saber.m.m@hotmail.com

2nd Mamoon Hassan
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine, P.O.Box:7
mamooun.11@gmail.com

3rd Amjad Hawash*
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine, P.O.Box:7
amjad@najah.edu

***Abstract*—With the population explosion and the increasing number of different types of buildings and the variety of their related facilities, the need for rapid proper searching of craftsmen to make new installations and/or make some repairs is also increased. Usually, people seek close, reasonable wages, and professional craftsmen to make some repairs or installations. Searching for such craftsmen is not an easy task with the increase of population in countries as well as the lack of related and accurate information. In this work, we are suggesting a fuzzy logic-based recommendation system embedded within a web-based database application. The system enables clients (customers) to manually search for craftsmen as well as the ability of the system to suggest craftsmen to clients according to the professionalism of craftsmen and their closeness, all ranked in descending order. Since manual searching for craftsmen is also not an easy task, the recommendation system is able to suggest the most suitable craftsmen to clients according to their needs. The experimental tests at the end of the work emphasize the importance of using a recommendation system instead of the manual search of craftsmen by comparing the manual and the fuzzy-based craftsmen searching in terms of time and effort.**

***Index Terms*—Recommendation System, Fuzzy Logic, Ranking, Time, Cost.**

I. Introduction

Problems related to population growth are a lot. More governmental investments, minimize total income, more unemployment rates, food problems, more poverty, lack of resources, reducing the efficiency of labor forces, negative effect on the environment, and lower standards of living are all considered as negative effects of population explosion. Governments, NGOs, Human societies, and others all trying to put continuous solutions for such (and other) problems [1].

The negative impact of the population growth on people's income in some societies motivates them to search for the cheapest services and/or products in order to be able to survive from an economic point of view [2]. It becomes very important for people to search for the most appropriate products and/or services that fit their income and needs. Searching for these is not an easy job with the tremendous rate of population growth. The dependency of people relations, relatives, and/or friends to suggest the best services and/products is not always an optimal solution.

Searching for the craftsmen [1] that fit people's needs and income is a very important goal similar to searching for the best services/products [3]. The issue is considered an optimization problem in the way that some of the tasks need immediate processing from the most available professional artisans and at a reasonable cost. It is not an easy task in the existence of thousands of artisans.

Personalizing and recommending resources to clients saves their needs and efforts [4]. Recommendation System utilized Significantly in applications because it became a crucial research area, which helps users to deal with big data and facilitate searching about what they need. Successful real-world applications including recommendation systems of Google, YouTube, Amazon, and Netflix.

In this work, we are developing a web-based tool to mastery clients to search for the artisans according to clients' needs and requirements. The tool enables clients to manually search for artisans as well as giving them the ability to use the embedded recommendation system to help clients find the most suitable artisans to their needs. The tool has a set of functionalities to make it easy for customers as well as artisans to communicate and collaborate to accomplish some tasks. The conducted experimental results emerged a minimal effort and time to accomplish tasks compared with the manual arrangement between clients and artisans.

The rest of this paper is organized as follows: Previous works are discussed in Section II. Section III discusses the system architecture of the implemented work while Section IV discusses the Fuzzy-Based Recommendation System in detail. Section V is related to the experimental tests. Finally, Section VI concludes this paper.

II. Related Work

Several works and tools exist to enhance the communication and collaboration between people. Part of these tools are dedicated for clients-artisans collaboration. Flowdock [2], GoToMeeting [3], Slack [4] and WebEx [5] are all tools used to manage online communication between people. Each of which has its own capabilities and facilities to make a fruitful communications.

With respect to tools related to managing tasks and projects, there exist a lot of available tools with several services. Asana [6]

Corresponding Author:amjad@najah.edu

[1]From now and on, the terms *Craftsmen* and *Artisans* will be used exchangeably.

[2]https://www.flowdock.com/
[3]https://www.gotomeeting.com/
[4]https://resources.workable.com/tutorial/source-on-slack
[5]https://www.webex.com/
[6]https://asana.com/

is a project management tool allows users to assign tasks to other members, add followers to projects and monitor deadlines. Dapulse [7] a collaboration tool that helps users communicate, set objectives and assign tasks. Its big advantage goes to its great visual design so it is easy to be understood and coped with. Redbooth [8] is an easy-to-use project management tool with a platform that allows users to plan and collaborate through many functions from video conferencing to creating Gantt charts.

For service supplier tools, there are a lot of available tools as well. SugarCRM [9] is a tool that allows all employees to create extraordinary customer relationships. It builds smaller journeys within the overall customer lifecycle and tracks each customers progress through the evaluation, selection, purchasing, and on-boarding steps of their journey. CustomersFirstNow [10] offers best-in-class analytics, experts, and technology for improving B2B and B2C customer experience. Their CFN Insight tool is a journey mapping platform that enables organizations to better understand customer experience from the outside-in to create more business value.

Recommendation systems are of great use and importance these days. Customers prefer to be supplied with the best services that fit their needs without paying a lot of time/effort for that. Service providers rely on recommendation systems in directing their services to their customers according to their needs and abilities [5]. There are several types of recommendation systems according to [6] .Collaborative, Content Based, Demographic Based, Utility Based, Knowledge Based and Hybrid recommendation systems are all examples.

The work presented in [7] discussing the recommendation of movies to users in a ranked format according to their interests. Firstly, the recommendation system defines the target movie and calculates the average rating of users who rated this movie using the distance measurement that is based on either Euclidean Distance, Manhattan Distance, and/or Pearson Coefficient Correlation and Cosine similarity. Secondly it looks for movies that have the same type of the target movie and the highest number of matching genres. Then calculates the average of similar movies using Euclidean Distance, Manhattan Distance, Pearson Coefficient Correlation and Cosine Similarity. Finally it gives the two output of rating to find the decision using Madman inference systems. Finally, it compares the result of these systems and reports the performance and errors.

Fuzzy logic-based recommendation systems are famous and a lot of published works adopt it in the design and implementation of their recommendation systems. Authors of the work presented in [8] propose a personalized recommendation system driven by fuzzy logic technique for products that target to recommending optimal products to prospective buyers, promoting the rate at which customers visit online stores and eventually increase sales for online businesses.

Target tracking is discussed in the work of [9]. It is a significant issue in numerous territories, including military, space, and modern applications. In view of estimation clamor, target position measurements are regularly dubious. Fuzzy filter is designed with only one linguistic variable needed for the fuzzy tracking estimator in order to reduce the complexity.

Our participation in this work is related mainly to recommend artisans to clients by generating a ranked list of the most related artisans according to clients' needs. Our recommendation systems is based on quantifying the ratings of clients to artisans and the distance between clients and artisans measured using GPS calculations. The tool also contains the sufficient users' interaction techniques to enhance the interactivity with clients.

III. System Architecture

In this section, we present an overview of the system and its architecture showing its major components and their functionalities. Users of the system are of two categories: *clients (customers)* and *craftsmen (artisans)*. The interaction between both categories is done using **CRAFTEE** responsive Web and Android applications. Both clients and artisans are able to execute a set of actions. With respect to artisans, they can add images about their works, advertise for their services, accept or refuse a work request that could be generated from one or more clients and they can fill a questionnaire regards their satisfaction after they finish providing services to some clients. Clients, on the other side, are also able to do some actions like follow artisans to be updated with all new, put likes for images posted by artisans, advertise a service request, manually search for artisans and use the embedded fuzzy logic recommendation system in the case they need a ranked list of related artisans.

A sample scenario of a client needs a carpenter to make some fixes for his/her furniture, for example, starts after the login of the client using his/her credentials. The client has two options for asking for a service from an artisan: (1) either directly after the manual search for his/her profile, or (2) using the embedded fuzzy logic-based recommendation system after filling some data expressing the needs of that client. In the second case, the system makes some calculations to be used in generating a ranked list of related artisans to be sent to the client's terminal in order to choose one of the artisans that appear in the list, then the client will be able to ask for the service s/he needs. Upon the artisans, acceptance to provide the service to requesting client, and when the agreed time of repair is due, the system contacts the GPS sub-system to draw a directive map to make it easy for the artisan to reach the location of the client as in Figure 1.

A. System Components

The system is composed of a set of integrated parts, each of which is responsible to provide some services.

In the following list, each component is described briefly taking into account the structure of each component as well as its role in the system:

1) **Clients**: As mentioned before, they are the seekers of services. They could be persons, companies, organizations, and even artisans requesting some services [11]. The

[7] https://monday.com/
[8] https://redbooth.com/
[9] https://www.sugarcrm.com/solutions/customer-journey-management/
[10] http://customersfirstnow.com/cfn-insight/

[11] From now and on, we will consider these categories as humans.

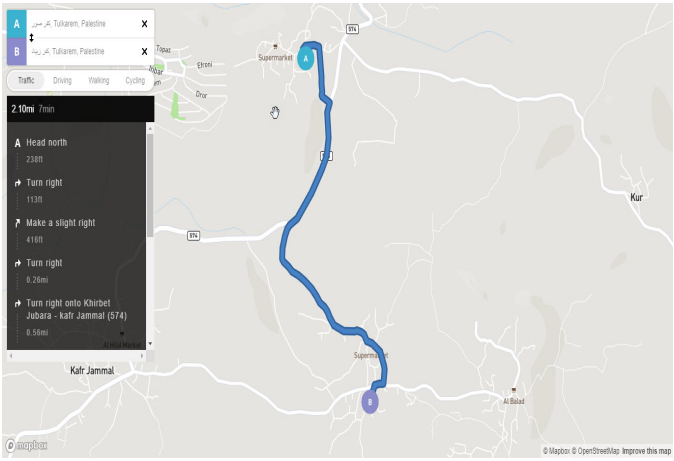

Fig. 1: The shortest path between a client and an artisan locations

system considers each artisan as a client. However, and during the creation of clients' accounts, the system asks them to provide their addresses to be converted to GPS locations.The code uses *Ajax* to save the GPS location of the client in the system's database. The system gives the ability for the client to choose the career as a beginning for artisan searching process that comes after requesting some service that includes the begin and end dates of the validity of requested service. The client can put likes to some images posted by artisans and fill evaluation reports for them.

2) **Artisans**: They represent the professional parts of the system. They are the parts that provide services. They can be persons, companies, or organizations. Upon some customer requests, they can accept or refuse the proposed requests. In the case of acceptance, the requesting client(s) is/are informed, all necessary data are saved in the database and the artisans were presented with a GPS map to reach the clients. Upon the completion of the service, the artisans are able to fill an evaluation form about customers to be saved also in the database.
3) **Database**: We used MYSQL as a DBMS in this work. Figure 2 below represents the Entity-Relationship diagram of the constructed database.
4) **The GPS Service**: As mentioned before and upon the acceptance of some artisan to accomplish some service, the system uses *MapBox* [12] API to calculate and draw the shortest distance between the client and the artisan.
5) **The Recommendation System**: It is the main part of the work and will be illustrated deeply in the following section.

IV. The Fuzzy-Based Recommendation System

As we mentioned before, the clients are able to manually search for artisans (if they know them) or they can fulfill the ability of the system to suggest the most appropriate artisans according to the clients' needs, using the fuzzy logic-based recommendation system. Figure 3 represents a sequence

[12]https://www.mapbox.com/

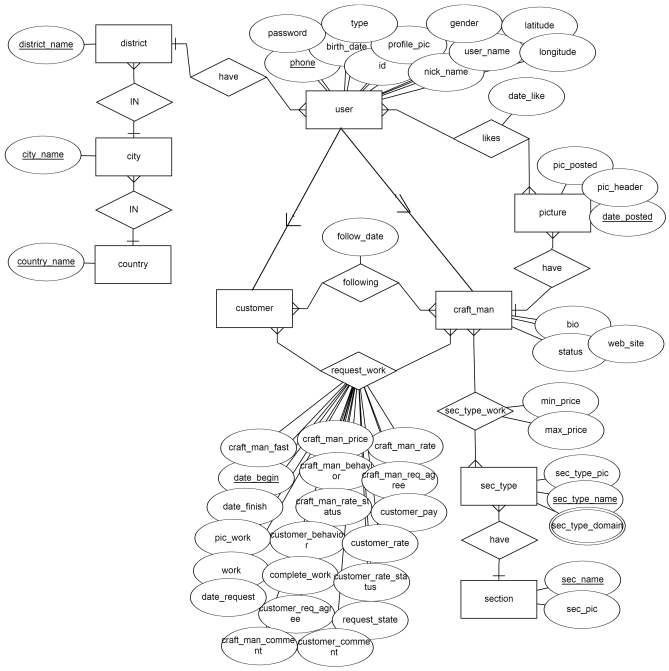

Fig. 2: The Entity-Relationship diagram.

diagram for requesting an artisan. The diagram includes the main objects involved in the process as well as the set of functions invoked.

Two fuzzy variables were used in the implemented fuzzy logic-based recommendation system: the clients' **Rating** and the **Closeness** between clients and artisans. If some client decides to use the recommendation system, his/her request is passed through an Ajax code to the database server to fetch all related artisans' information to be returned back from the database server to the recommendation system (implemented using PHP code). Some calculations take place in the recommendation system in order to quantify the first fuzzy variable (the ratings). Rating for each individual artisan is calculated using the formula:

$$\text{Rating} = p(\text{Likes})*0.2 + p(\text{Followers})*0.3 + p(\text{Works})*0.5 \quad (1)$$

where $p(Likes)$ represents the proportion of a number of all likes/artisan to a total number of likes for all artisans from the same career. $p(Followers)$ is calculated the same but with respect to the number of artisan followers, and $p(Works)$ is also calculated the same way but with respect to the number of artisan requests. The fractions that appear in the formula are given weights just to emphasize the importance of each factor in the formula [13].

For example, if some artisan (carpenter for example) has the following data: likes=100 (all carpenters likes=500), followers=50 (all carpenters followers=300) and accomplished works=80 (all accomplished carpenter works=400), then his/her rating will be calculated as follows: $rating = 100/500 * 0.2 + 50/300 * 0.3 + 80/400 * 0.5 = 0.19$

This calculated value will be used as his/her **Rating** fuzzy variable to be input in the fuzzy logic based recommendation

[13]Although these weights need to be tested in order to be sure of their exact values, they are chosen arbitrarily here just for trial of the system purposes.

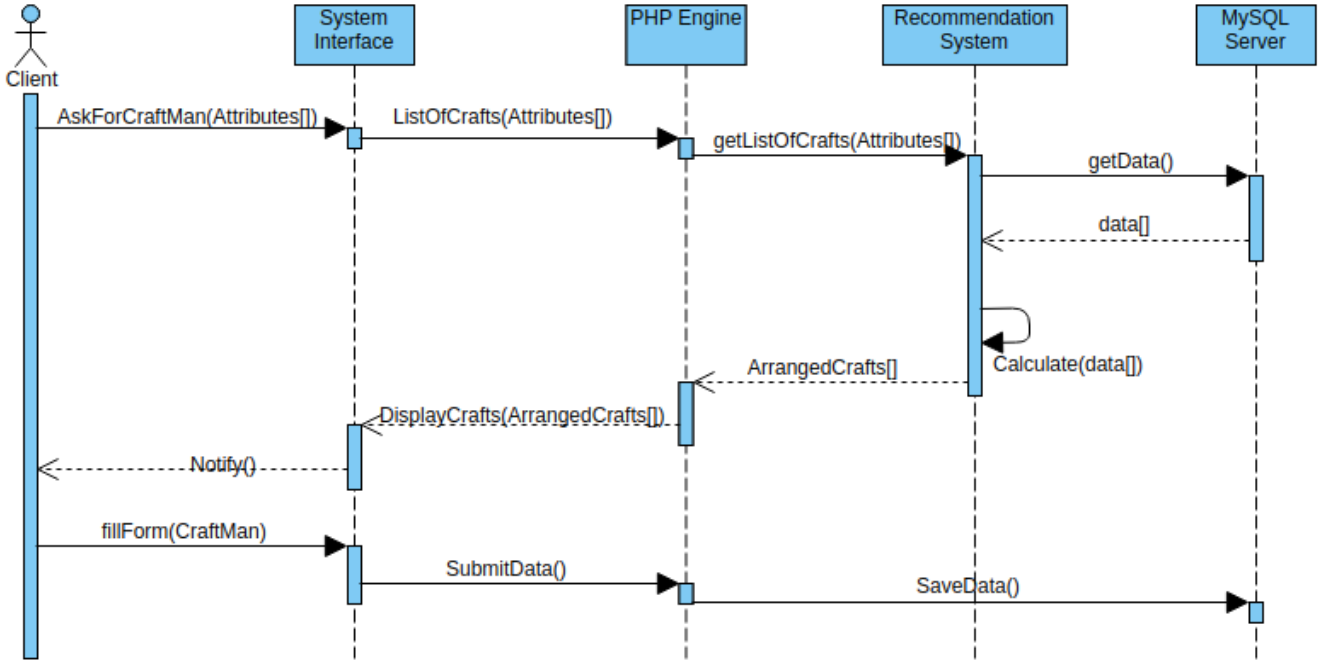

Fig. 3: Sequence diagram for artisans suggestion.

system. The second fuzzy variable will be the distance between the carpenter and the customer. All same data are calculated for all retrieved artisans in order to be inputs for the fuzzy system to generated a ranked list that will be returned to the client. However all artisans retrieved data are updated with respect to the rank of each one by updating the rank field in the artisan object.

A. Inference Engine

The fuzzy logic inference engine relies on two fuzzy variables: **Rating** and **Closeness**. Fuzzy intervals for **Rating** are: $0.0 - 0.5$ (poor), $0.25 - 0.75$ (fair), $0.5 - 1$ (good), and > 1 (excellent), while for **closeness** are: $0 - 10$ (close), $5 - 15$ (moderate), and > 15 (far) measured in Kilometers. In this work, we applied k-means algorithm to compute the sizes of these intervals followed by some tuning on the boundaries to have better and simpler sizes. Figure 4 illustrates the **Rating** intervals, while Figure 5 shows the **Closeness** intervals.

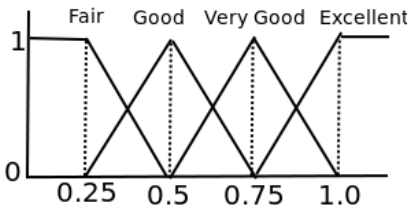

Fig. 4: **Rating** fuzzy intervals.

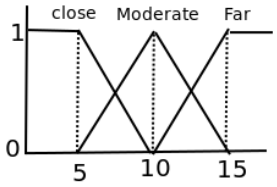

Fig. 5: **Closeness** fuzzy intervals.

Table I represents the possible values of combining the two fuzzy variables **Rating** and **Closeness** where the combination of any two values of these two variables provides a different linguistic value. For example: if **Rating** is *Good* and **Closeness** is *Close*, then the output will be *Very Good*. However, the fuzzification and defuzzification processes depending on this table that used to generate ranks of artisans to clients by applying the **And** operator on fuzzy sets.

TABLE I: Possible linguistic values.

	Fair	**Good**	**Very Good**	**Excellent**
Close	Fair	Good	Very Good	Super
Moderate	Acceptable	Fair	Good	Very Good
Far	Fail	Acceptable	Fair	Good

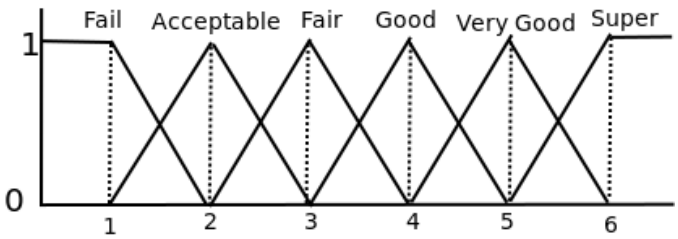

Fig. 6: The fuzzy intervals of the output linguistic values.

B. Defuzzification Process

To obtain final crisp values for both **Rating** and **Closeness**, a defuzzification process is executed depending on the **Centre of Gravity for Singletons** formula:

$$CrispValue = \frac{\sum_{i=1}^{n} \mu_i \times o_i}{\sum_{i=1}^{n} \mu_i} \quad (2)$$

where o_i is the corresponding value for the output variable of index i, μ_i is the membership function after accumulation for that variable, and n is the number of variables.

C. Example

To illustrate how the fuzzy based system works, we will go on with an example illustrating the different calculations to generate a ranked list of artisans. A client C needs to use the system to search for carpenters. C needs to use the recommendation system for carpenters suggestions. Two carpenters $R1$ and $R2$ with ratings 0.68 and 0.8, and distances 11 and 6 Kilometers respectively. In this case, the ranking of the two carpenters are calculated as follows, for bot $R1$ and $R2$:

For $R1$:

1) For **Rating**:
   a) $\mu(VeryGood) = \frac{0.68-0.5}{0.75-0.5} = 0.72$
   b) $\mu(Good) = \frac{0.75-0.68}{0.75-0.5} = 0.28$
2) For **Closeness**:
   a) $\mu(Far) = \frac{11-10}{15-10} = 0.2$
   b) $\mu(Moderate) = \frac{15-11}{15-10} = 0.8$

Now, from Table I, we can compute the following:

1) if *Very Good* and *Far* then *Fair*, so $0.72 \cap 0.2 = min(0.72, 0.2) = 0.2$
2) if *Very Good* and *Moderate* then *Good*, so $0.72 \cap 0.8 = min(0.2728, 0.8) = 0.72$
3) if *Good* and *Far* then *Acceptable*, so $0.28 \cap 0.2 = min(0.28, 0.2) = 0.2$
4) if *Good* and *Moderate* then *Fair*, so $0.28 \cap 0.8 = min(0.28, 0.8) = 0.28$

As a result we have:

1) $\mu(Fair) = max(0.2, 0.28) = 0.28$
2) $\mu(Good) = 0.72$
3) $\mu(Acceptable) = 0.2$

Then Ranking of $R1$ = $\frac{0.28*3+0.72*4+0.2*2}{0.28+0.72+0.2} \approx 3.43$ and represented by the shaded area in Figure 7 below.

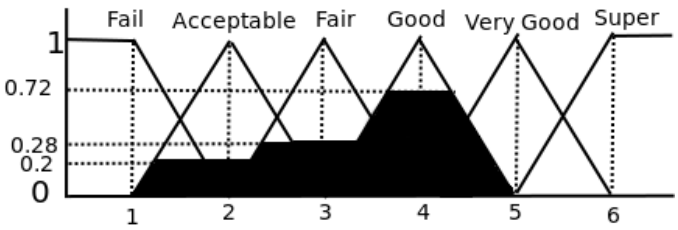

Fig. 7: The shaded are represents the calculated probability for both **Rating** and **Closeness** variables for $R1$.

For $R2$:

1) For **Rating**:
   a) $\mu(Excellent) = \frac{0.8-0.75}{1.0-0.75} = 0.2$
   b) $\mu(VeryGood) = \frac{1.0-0.8}{1.0-0.75} = 0.8$
2) For **Closeness**:
   a) $\mu(Moderate) = \frac{6-5}{10-5} = 0.2$
   b) $\mu(Close) = \frac{10-6}{10-5} = 0.8$

Now, from Table I, we can compute the following:

1) if *Excellent* and *Moderate* then *Very Good*, so $0.2 \cap 0.2 = min(0.2, 0.2) = 0.2$
2) if *Excellent* and *Close* then *Super*, so $0.2 \cap 0.8 = min(0.2, 0.8) = 0.2$
3) if *Very Good* and *Moderate* then *Good*, so $0.8 \cap 0.2 = min(0.8, 0.2) = 0.2$
4) if *Very Good* and *Close* then *Very Good*, so $0.8 \cap 0.8 = min(0.8, 0.8) = 0.8$

As a result we have:

1) $\mu(Good) = 0.2$
2) $\mu(VeryGood) = max(0.2, 0.8) = 0.8$
3) $\mu(Super) = 0.2$

Then Ranking of $R2$ = $\frac{0.28*4+0.8*5+0.2*6}{0.2+0.8+0.2} \approx 5.26$ and represented by the shaded area in Figure 8 below.

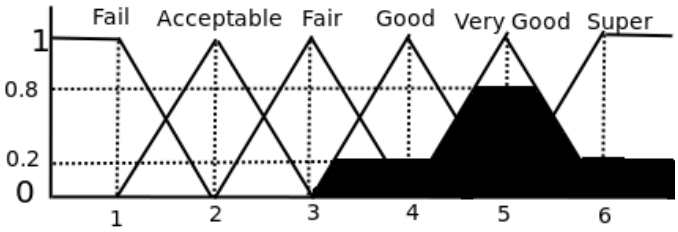

Fig. 8: The shaded are represents the calculated probability for both **Rating** and **Closeness** variables for $R2$.

By comparing the two ranking values 3.43 and 5.26 then $R2$ will have a higher rank and will appear in the ranked list before $R1$.

V. Experimental Tests

In order to examine the recommendation system for this work, we managed an experimental test to compare the searching for artisans by the manual search against the fuzzy-based recommendation system with respect to **time** and **cost**.

we asked a set of volunteers to participate in this test. The sample of volunteers was chosen randomly and they don't know each other. The sample is divided into two subsets: the **Clients** with 15 participant and the **Artisans** with 20 participants. Before beginning the experimental test, we provide **Clients** set with the names of **Artisans** members (just the names without mobile numbers nor their careers) and asked the members of **Clients** set to being the process of searching for artisans in a simulated way as if they are really requesting some services and searching for artisans to accomplish these services to them. During this process, we calculated both amounts of time needed to reach some artisan and the cost spent for that process in terms of the cost of mobile calls done to reach the artisan. We noticed during this test, that most of the members in the **Clients** set used to contact some of their relatives asking for the names of members in **Artisans**

set and asking their relatives if they know anything about the careers of these names. After we finished the first part of the experiment, we began the second part by asking the members of **Artisans** set to create profiles for themselves in the system and upload a set of images for their works. This process took several days to be sure that members of the **Artisans** set to have some how enough amounts of likes and followers from both sets. After that, we asked the members of the **Clients** set to use the recommendation system embedded in our work to search for some artisans to accomplish some supposed tasks. Also here we computed both the time and the cost for this part of the test.

Table II below, summarizes the collected data during the two parts of the experimental test. The column **Test#** is related to the number of participants from the **Clients** set. The columns **Search Time** and **Accumulated Time** under the column title **Manual Method** are related to the first part of the experimental test. The first column represents differences between the begin and end dates of the search time while the second represents the accumulated minutes spent for the search itself. The same columns are also appear for the recommendation system. Figures 9 and 10 depicts the data presented in Table II in which it is clear that there is a noticeable degradation in both search time and cost.

TABLE II: The collected data for the experiment.

	Manual Method			Recommendation System	
Test #	**Search Time**	**Accumulated Time**	**Cost**	**Search Time**	**Cost**
1	2 days	40 min	$ 7.00	1 min	$ 0.15
2	1 day	10 min	$ 5.00	2 min	$ 0.15
3	1 week	60 min	$ 10.00	2 min	$ 0.15
4	10 hours	8 min	$ 6.00	1 min	$ 0.15
5	4 hours	5 min	$ 3.00	3 min	$ 0.15
6	1 day	25 min	$ 5.00	1 min	$ 0.15
7	1 day	23 min	$ 7.00	2 min	$ 0.15
8	3 day	55 min	$ 2.00	2 min	$ 0.15
9	2 day	27 min	$ 3.00	3 min	$ 0.15
10	3 hours	19 min	$ 6.00	4 min	$ 0.15
11	5 hours	25 min	$ 7.00	3 min	$ 0.15
12	2 hours	19 min	$ 3.00	1 min	$ 0.15
13	5 days	36 min	$ 15.00	5 min	$ 0.15
14	2 days	26 min	$ 5.00	2 min	$ 0.15
15	3 hours	30 min	$ 2.00	2 min	$ 0.15

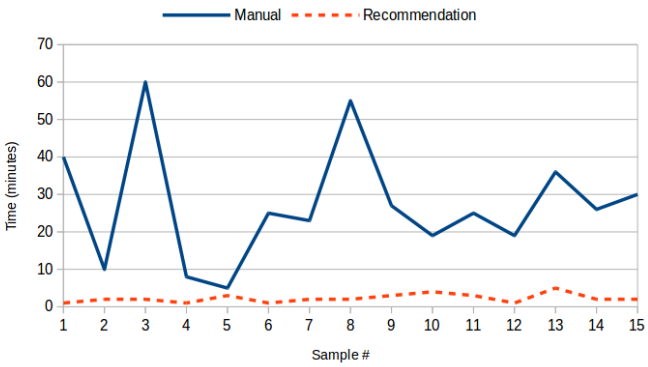

Fig. 9: The comparison between the manual and the recommendation-based searching in terms of **Time**.

VI. Conclusion

In this work, we developed a web-based system to enhance the mapping between Clients and Artisans. The system

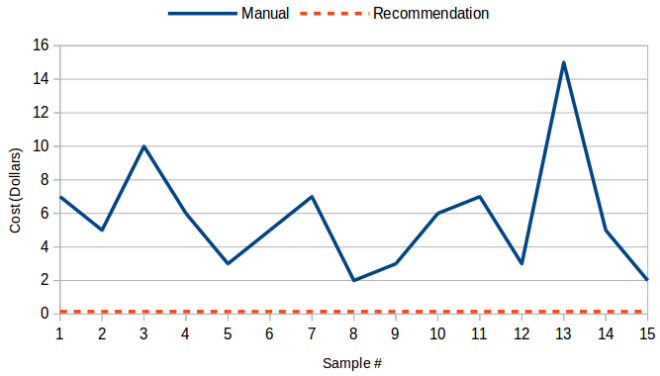

Fig. 10: The comparison between the manual and the recommendation-based searching in terms of **Cost**.

equipped with enough services to make it easy for both to create and manage their profiles. The problem of finding the most appropriate artisan in a tolerable time and cost is implemented and tested here by the developed fuzzy logic-based recommendation system. The experimental results at the end of the work emerged as a step forward in the process of searching for the most suitable artisans in terms of both ratings and closeness variables. For future works, we are planning to introduce more features to the system like policies to block fake accounts, enable advertisements and study the weights that appear in Formula 1.

References

[1] P. R. Ehrlich and J. P. Holdren, "Impact of population growth," *Science*, vol. 171, no. 3977, pp. 1212–1217, 1971. [Online]. Available: http://www.jstor.org/stable/1731166

[2] F. Cingano, "Trends in income inequality and its impact on economic growth," no. 163, 2014. [Online]. Available: https://www.oecd-ilibrary.org/content/paper/5jxrjncwxv6j-en

[3] A. Anderson and S. Jack, "Role typologies for enterprising education: The professional artisan?" *Journal of Small Business and Enterprise Development*, vol. 15, 05 2008.

[4] N. Polatidis and C. Georgiadis, "Recommender systems: The importance of personalization in e-business environments," *International Journal of E-Entrepreneurship and Innovation*, vol. 4, pp. 32–46, 10 2013.

[5] M. pilka, A. Posoldova, G. Rozinaj, and P. Podhradsk, "Importance of recommendation system in modern forms of learning," in *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, May 2016, pp. 1–4.

[6] A. Ansari, S. Essegaier, and R. Kohli, "Internet recommendation systems," *Journal of Marketing Research*, vol. 37, no. 3, pp. 363–375, 2000. [Online]. Available: https://doi.org/10.1509/jmkr.37.3.363.18779

[7] M. M. Rahman Siddiquee, N. Haider, and M. Rahman, "A fuzzy based recommendation system with collaborative filtering," *SKIMA 2014 - 8th International Conference on Software, Knowledge, Information Management and Applications*, 04 2015.

[8] B. Ojokoh, M. Omisore, O. Samuel, and O. O., "A fuzzy logic based personalized recommender system," *International Journal of Computer Science, Information Technology, & Security*, vol. 2, pp. 2249–9555, 10 2012.

[9] C.-W. Tao and W. E. Thompson, "A fuzzy logic approach to target tracking," 2013.