International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 10 Issue 4, April-2021, Impact Factor: 7.957

Online Medical Service Recommendation scheme in eHealthcare system

Dr. S. P. Predeep Kumar[1], Dr. E. Baburaj[2], S. P. Ajith Kumar[3], Dr. Hardeo Kumar Thakur[4]

[1]Professor, Mohandas College of Engineering & Technology, Nedumangad, Trivandrum. Kerala, India
[2]Professor, Marian College of Engineering & Technology, Kazhakuttam Thiruvananthapuram, Kerala, India
[3]Lecturer, (Selection Grade), Bhai Parmanand Institute of Business Studies, Govt. of NCT of Delhi
[4]Department of Computer Science and Technology of Manav Rachna University (MRU), Faridabad

ABSTRACT

With the continuous development of eHealthcare systems, medical service recommendation has received great attention. However, although it can recommend doctors to user's there are still challenges in ensuring the accuracy and privacy of recommendation. In this paper, to ensure the accuracy of the recommendation, also consider doctor's reputation scores and similarities between user's demands and doctor's information as the basis of the medical service recommendation. The doctor's reputation scores are measured by multiple feedbacks from users. Also propose two concrete algorithms to compute the similarity and the reputation scores in a privacy-preserving way based on the Modified Paillier cryptosystem, truth discovery technology, and the Dirichlet distribution. Detailed security analysis is given to show its security prosperities. In addition, extensive experiments demonstrate the efficiency in terms of computational time for truth discovery and recommendation process.

***Index Terms*—eHealthcare systems, medical service recommendation, privacy-preserving**.

INTRODUCTION

ONLINE medical service recommendation has become an indispensable part of daily life, due to the rapid development of eHealthcare industry. In a medical service recommender system, users submit their demands to the medical server, and then the medical server will recommend the suitable doctors according to the demands of the users. A series of existing studies have made efforts to design the recommendation systems. Some of these adopt trust and reputation as the basis of recommendation. while others give more importance to demands and interests of users. In the first type, trust and reputation are a reaction of the service provider's quality of service and a good service provider will have a high reputation scores. The server will recommend the service provider with high reputation scores to users. In the second type of works, the server matches the suitable service provider according to the user's demands (e.g., personal requirements or interests) However, considering only the single factor (i.e., reputation or user's demands) as the basis of recommendation may affect the accuracy of the recommendation results. I proposed a service recommendation scheme based on the reputation. The server recommends the service provider with high reputation to user's, however it is important to note that the service provider with high reputation may not be able to meet the user's demands well. Moreover, reputation is a factor derived from feedback of patients, which may or may not truly react the services needed by the users.

False feedback maliciously entered can also affect the reputation, hence it becomes extremely important. In the server recommends the doctors based on the similarities between user's demands and doctor's information. However, the recommendation scheme based on similarity only, may recommend doctors with bad quality of service. In real world, in order to get a better recommendation result, besides similarity of basic information, the feed backs of multiple user's on the service provider need to be considered. For example, if only similarities are considered, there is a possibility that the server may recommend a doctor who meets the basic demands (e.g., doctor's department, title) of the user but has a poor reputation to the user. To solve these problems, it is critical to propose an accurate medical service recommendation scheme based on both similarity and reputation scores, in which it can not only meet the basic demands of users, but also recommend doctors with high reputation scores to the user, so that the user can obtain a good quality of medical services. At the same time, the system should also identify and later malicious user's feedback, either it is positive or negative. Considering the sensitivity of information (e.g., feed backs, personal information, etc.) during the recommendation process, the information of users and doctors should be privacy preserving. Privacy-preserving means sensitive data is protected and not compromised. In key management is used to ensure the security, but key management is just the basic cryptographic

primitive and cannot be used to achieve privacy preserving recommendation. In SMPC-based methods are used to achieve privacy-preservation. However, these methods are computation intensive, which makes them less practical for large number of users. In random parameters are used to perturb the sensitive data before forwarding it to the server, causing the server to receive inaccurate user's feed backs and resulting users will receive low-quality recommendations. In this project i attempt to solve the challenges of recommender system by a privacy-preserving medical service recommendation scheme for eHealthcare system, called PPMR.

LITERATURE REVIEW

The author's Jerry C.C. Tseng, Bo-Hau Lin, Yu-Feng Lin [1] developed an interactive system it include the personal health management system module and a real-time interaction module also a mobile app for users easily make use the system. It cannot integrate the system with healthcare wearable devices.

These methods include, among others, the better modeling of users and items information stored in the database, taking into consideration of the extra information such as feed- back, multi criteria ratings into the recommendation process. This system is able to successfully integrate recommendation and semantics. This model proves that recommendation can be improved if semantic factor while recommending product or services is integrated in the system.

An OHC as a HHIN to keep rich context information of users and threads, and used the structural information to predict a user's preference in threads. Also used different sets of features to capture basic network metrics and thread-thread relationship. Then the features to train a binary classification model for thread recommendation. The results showed that they can effectively boost the recommendation performance by integrating thread-thread relationship with basic network metrics.

They are suggested a personalized healthcare service recommendation framework that considers consumer's health status to find adequate services for them. This framework gathers information about service consumer's health status and calculates medical similarities between consumer and healthcare services automatically. Based on these similarities of each consumer, the framework arranges and recommends proper healthcare services. Also implemented HSRF and evaluated its functionality and feasibility. Although the evaluation was not fully certain to prove all approaches of this paper [4].

SYSTEM DESIGN

This section formally presents the system and security models designed for PPMR, followed by its design goals.

A. system model

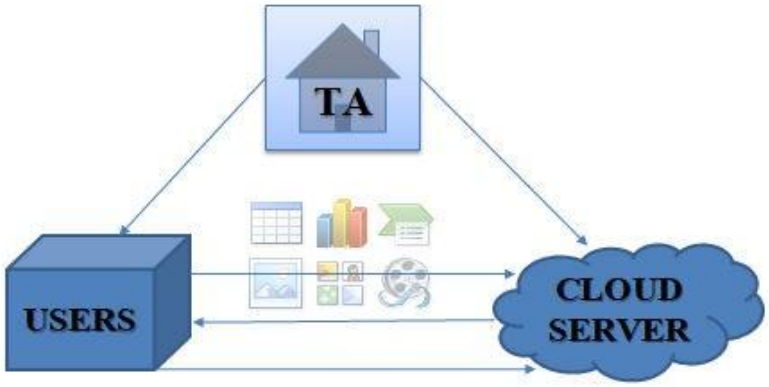

Fig. 1: The System model.

Fig. 1 presents the formal system model, which has three entities: Trust Authority (TA), Cloud Server, and Users (U). TA is responsible for managing and distributing the key materials to users and cloud server. Cloud Server entertains the demands from users for recommendation and receives feedbacks from users for calculating the doctor's reputation scores.

Users are a group of patients with medical requirements. Each user ui

$$u_i \in U = \{u_1, u_2, ..., u_N\} \quad (1)$$

can use a smart device to send demands and the acceptable threshold of similarity to the cloud server for recommending a suitable doctor. After finishing a medical service, the user gives feedback to cloud server as evaluating of the quality of the doctor's service.

Table I: Details of Notations Used

Symbol	*Definition*
N	No. of users
M	No. of doctors
l	No. of vector's dimensions
$\vec{A}_i$	Demand vector of user i
$\vec{B}_j$	Attribute vector of doctor j
$\vec{A}'$	Perturbed Demand vector of user i
$\tilde{}_i$	Perturbed attribute vector of doctor j
B'	Similarity threshold provided by user
i	Weight of user i (for feedback)
T_S	Feedback score of user i
w_i	Truth value of feedback scores
x_i	The standard deviation for users' feedback scores
x^*	
std_m	

PROPOSED SCHEME

The PPMR scheme is applied in the privacy-preserving medical service recommendation scenario, where the user sends a request to the server, and then the server recommends doctors based on user's demands. After the medical service is finished, the user will give a feedback according to the performance of the doctor to evaluate the doctor's service quality. In the System Initialization phase, TA will generate security key materials and distribute them to users and cloud server. In the Doctor Recommendation phase, after receiving the user's demands and the acceptable threshold of similarity, the cloud server will calculate the similarities between user's demand vectors and doctor's attribute vectors. Then the server will recommend a suitable doctor to the user according to the similarities and the acceptable threshold of similarity. In the Reputation Calculation phase, the server will aggregate the user's feedbacks and calculate the truth value. Finally, these truth values will be used to calculate the doctor's reputation.

A. System Initialization

TA is completely trusted by all entities, and hence responsible for bootstrapping the system based on the security parameter, the trust authority initially selects two large safe prime number p,q, such that,

$$|p| = |q| = K. \quad (2)$$

B. Doctor Recommendation

In this section, to introduce how the server recommends a suitable doctor to a user. The whole process can be divided into three parts: demands sending, similarity calculation, and doctor recommendation.

1) ***Demands sending:*** Before sending the request to med- ical server for recommendation, a user ui needs to verify its identity and provide credentials to server. However, in my work, do not discuss the details of the identity authentication process because it is not the main points of recommendation. Once the server verifies that ui is authorized by TA

2) Similarity calculation: After receiving ui's demand $\vec{A}_i$ the server calculates the similarities between $\vec{A}_i$ and doctors' attribute vector. The following steps are followed:

1) The server receives $\vec{A}_i$ from ui
2) TA computes $p' = s' * t \quad (3)$

and sends p' to the server. [Since s' is the secret key between TA and ui and t is chosen randomly, the server will not obtain and t is chosen randomly.the server will not obtain s' even if it knows p'].]

3) The server computes the distance dis between each doctor's attribute vector $\tilde{B}'_i$ and $\tilde{A}'$ as follows:

$$dis = \mathrm{Xi}^{\mathrm{X}}_{i=1} (a' - b')^2$$

4) The server utilizes the parameter p received from TA to compute the similarity as follows:

$$sim = \mathrm{Xi}^{\mathrm{Xi}}_{i=1} (ai - bi)^2$$

3)Doctor recommendation: When ui applies for a medical service to the server, ui sends his own demand Ai and an acceptable similarity threshold Ts to the server. After computing the similarity.

$$simj \in \{sim1, sim2, ..., simM\} \quad (4)$$

between ui and each doctor

$$Dj \in D = \{D1, D2, D3.....DM\} \quad (5)$$

the server selects doctor Dj Where

$$simj \supset Ts. \quad (6)$$

Then, among the selected doctors, the server selects the doctor Dj with the highest reputation score and recommends Dj to ui

C. Reputation score calculation

After completing a medical service, ui will give a feedback score to evaluate the doctor's service. These feedback scores are used to calculate the reputation score of the doctor. The whole process can be divided into two parts: privacy-preserving truth value calculation and reputation calculation.

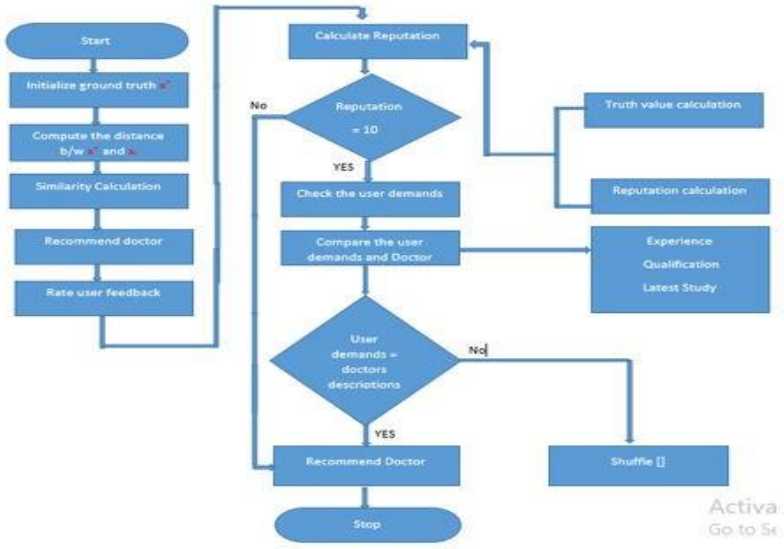

Fig. 2: Doctor Recommendation

1) Privacy-preserving truth value calculation: We dynamically assign different weights to the feedback scores of each user, and constantly update the truth value of multiple feedback scores so that a truth value can be computed to represents multiple feedback scores. During this process of truth value calculation, the user has the secret key s1 and the server has the secret key s2, hence the user and the server work together to achieve the privacy-preserving weight update and the truth value update. Specially, the truth value calculation is divided into the following two phases.

1) PHASE ONE: Secure weight update:

1. The server distributes the estimated ground truth x* to all the N users who provide the feedback scores for a specific doctor. For each user ui, the feedback score is xi. The estimated ground truth is a randomly initialized for the first iteration.

2. After receiving x from the server, each user ui computes the distance between xi and x as

$$Dis_i = d(xi, x) \tag{7}$$

according to Previous equation. Then ui uses a random number rij to compute

$$C_{ij} = (1 + n * Dis_i)h^{r_j} \mod n^2 \tag{8}$$

$$c_{ij} = g_{r_{ij}} \mod n^2$$

where j is the iteration times. Afterwards ui submits Cij and cij to the server.

3. When the server gets

$$(C_1j, C_2j, ..., C_Nj) and (c_1j, c_2j, ..., c_Nj)$$

from all users, it computes the aggregated results as:

$$C_j = \prod_{i=1}^{N} (1+nD\ iS_i)h \sum_{i=1}^{N} r_{ij} \mod n^2 \tag{9}$$

$$C_{ij} = g \sum_{i=1}^{N} r_{ij} \mod n^2$$

4. The server uses its secret key s2 to partially decrypt the aggregated results as:

$$C_j = \frac{C_j}{C^{s}_{j}2} \tag{10}$$

5. After receiving Cj and cj from server, each user ui can decrypt the aggregated results by using their secret key s1 as:

$$C_i = \frac{C_j}{C\ S\ J^1}$$

Following this the user ui computes the

N

$$\sum_{I=1}^{X} Dis_i \quad (11)$$

and the weight wi as follows:

$$Sum_d = \sum_{i=1}^{X} Dis_i W_i = \log \frac{sum_d}{Dis_i}$$

2) PHASE TWO: Secure truth update:

After updating all the weights, ground truth is updated. This is done using the following steps:

a) ui computes the weighted data as xi ·wi and then encrypts both the weight and weighted data as follows

$$W_ij, 1 = (1 + n(\dot{x}_i \dot{w}_i))h^r j mod n^2 \quad (12)$$

b) Similar to the Step 3 and Step 4 in secure weight update phase, the server aggregates all the user's data and collaborates with the user to compute

$$\sum_{i=1}^{N} X_i W_i$$

and

$$\sum_{i=1}^{N} X$$

Then the series update the ground truth as

$$V^* = \frac{\sum_{N} (X_i W_i)}{\sum^{N} (W_i)} \quad (13)$$

*2) **Reputation calculation:*** A two phase process is followed for calculating the doctor's reputation score, by using the truth values from different time periods as observation values. The first phase is truth aggregation this scheme use Dirichlet distribution to estimate and predict a doctor's reputation scores. The second one is Reputation score calculation.

PERFORMANCE EVALUATION

The performance of my scheme aims to evaluate the computational costs of the project during doctor recommendation and reputation score calculation process.

A. Doctor recommendation

To show the computation costs of PPMR, to compare my schemes with FSSR [8], which recommends doctor based on similarity. In my scheme, when recommending a doctor, the server considers the similarity as well as the doctor's reputation score. In Fig.5.1 first graph, to compare the running time of doctor recommendation against varying number of doctors. It can be observed that as the number of doctors increases, the running time of both PPMR and FSSR increases, this is due to the fact that the server calculates more similarities between user's demands and all the doctor's attributes. In Fig.3, see that the time required for selecting a doctor based on reputation score is very short. The reason is that the number of doctors selected based on similarity is small, so the server selects a suitable doctor based on reputation score among a small group of doctors.

Although, to consider similarity and reputation as the basis of recommendation, FSSR only considers similarity, but the efficiency of my PPMR scheme is similar to that of FSSR. From Fig.4 suppose when the number of doctors is increases,

we assume the time of similarity calculation is 8.301s, the time of selecting a doctor based on reputation score is 0.001s,

and the total time of recommendation is 8.302s. Hence, the running time of doctor recommendation in my PPMR scheme is 8.302s, while the running time of doctor recommendation according to the similarity in FSSR is 8.301s, which is almost equal.

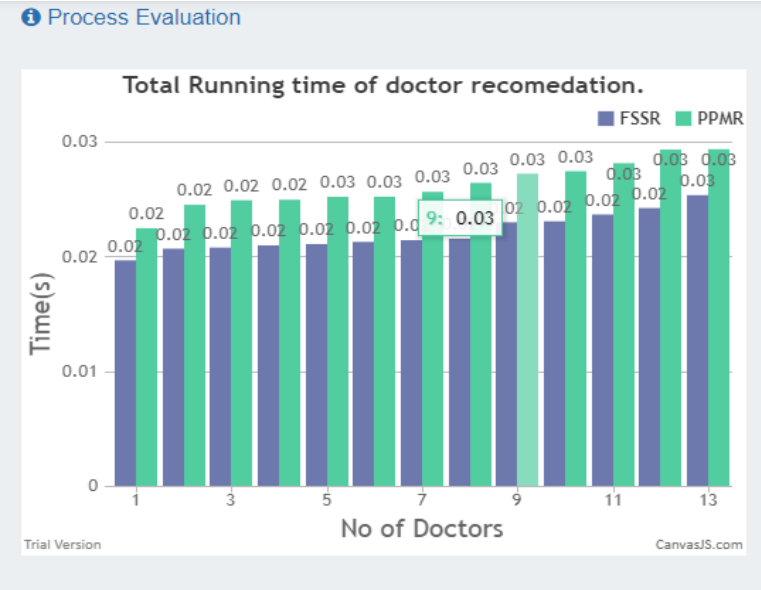

Fig. 3: Total running time of doctor recommendation

1) ***Reputation score calculation:*** Before calculating a doctor's reputation score, also need to obtain the truth values of multiple user's feedback scores from different time periods. In Fig. 5, 6, 7 , show the comparison of running time of privacy- preserving truth discovery scheme of my scheme[PPMR] and PPTD with varying number of users. From Fig.5, it can be observed that the running time of my scheme is far less than that of PPTD as the number of users increases. For example, when the number of users is 500, Assume that my PPMR

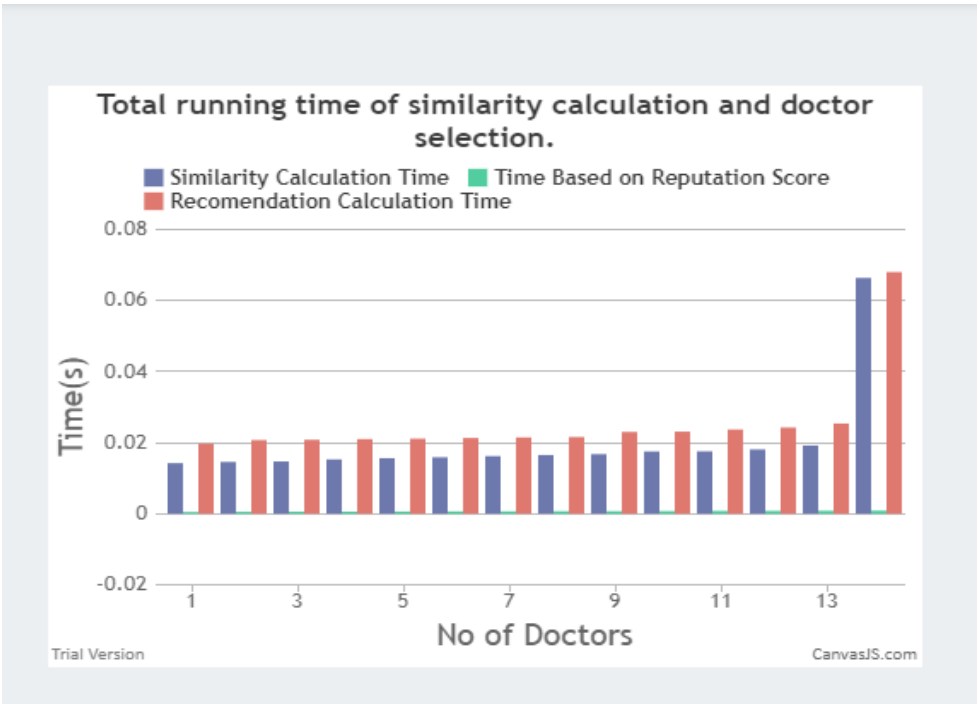

Fig. 4: Total running time of similarity calculation and doctor selection

Scheme costs 14.589s to complete the truth discovery process, while PPTD costs 652.23s. Fig. 6 and 7 show the time required in seonds for weight update, truth update respectively. We can observe that the running time of PPMR is less than that of PPTD as the users increase from 100 to 500. For example, when the number of users is 500, PPMR costs 0.116s to update the weights and costs 14.473s to update the truth value, while PPTD costs 617.721s to update the weights and costs 34.509s to update the truth value. Fig.8,9,10 shows the running time of truth calculation with varying number of users in PPMR scheme. From Fig. 8, we can see that as the number of users increases, the computation time of truth values grows linearly. In Fig.9, the running time of weight update and truth update also grow linearly with varying number of users. During the weight update, more user's feedback scores need to be used to compute the distance and need to be encrypted, hence the running time is related to the number of users. The same results in truth update process can be observed, as the number of users increase, the server receives more encrypted data from users and performs more aggregation operations to calculate wi and xi ·wi, thus the running time forms a linear relation with the number of users. 10 shows the total time of reputation score calculation with varying number of truths that are obtained from different time periods. Also, find that when the number of users increase the change in the time of reputation score calculation is relatively small. Compared with the time of calculating the truth values, the time of calculating the reputation score is very small, which confirms the efficiency for my scheme.

From the performance of the reputation score calculation phase, to see that my scheme is more efficient than PPTD in calculating the weight and truth values of users. The reason is that in PPTD scheme, during the truth discovery process (i.e., weight update and truth update phases), the users' values are encrypted by the threshold paillier cryptosystem, and PPTD needs to perform time-consuming module exponent operations. In PPMR, the user's values are encrypted by the modified paillier cryptosystem, and also use the properties under modulo n in modified paillier cryptosystem to convert the factorial operations into summation operations to improve the efficiency. Thus, PPTD needs to perform time-consuming

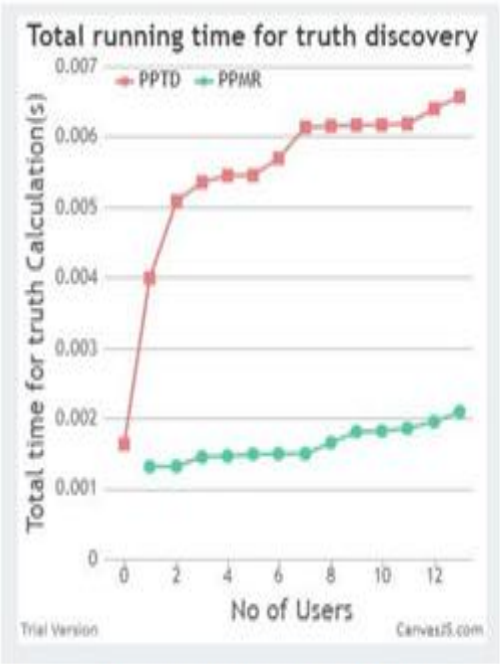

Fig. 5. Total running time for truth discovery

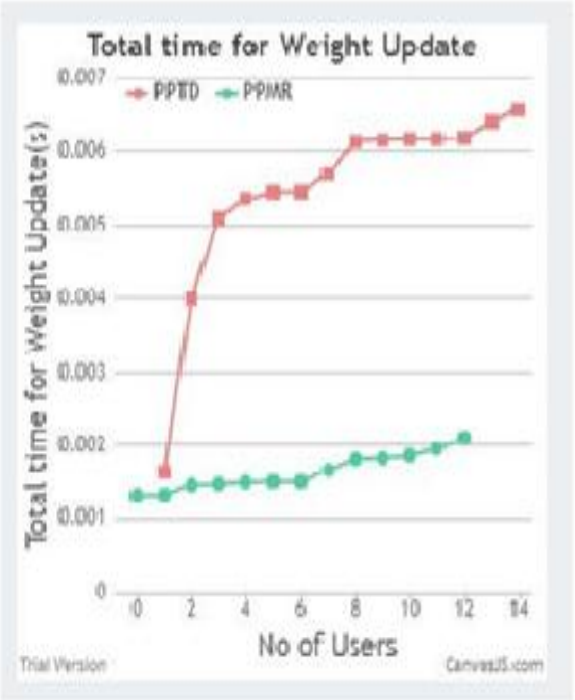

Fig. 6. Time for weight update

module exponent operations, while only multiplication operations are required in PPMR. Therefore, with the increase of the number of users, PPMR is more efficient.

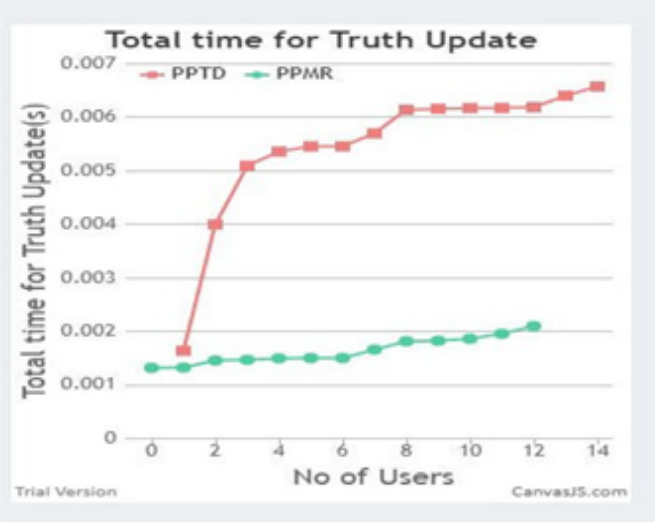

Fig. 7. Time for truth update

Fig. 8. Total running time of truth discovery

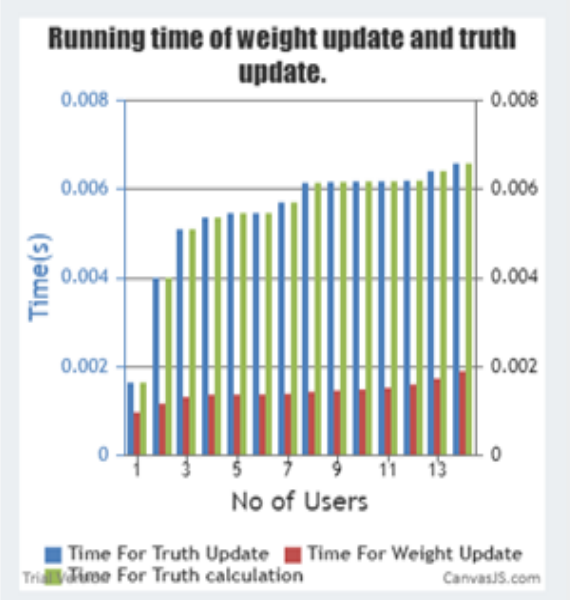

Fig. 9. Running time of weight update and truth update

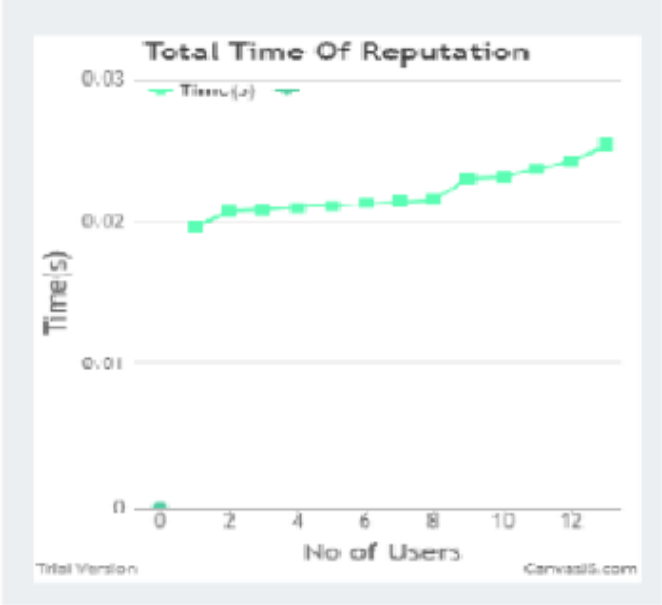

Fig. 10. Total time of reputation score calculation

CONCLUSION

The traditional process of manually taking and maintaining employee attendance is highly inefficient and time consuming. The attendance monitoring system based on biometric authentication has a potential to streamline the whole process. An Internet of Things (IoT) based portable biometric attendance system can prove to be of great value to institutions in this regard as it proves to be highly efficient and secure. The cost involved in making this system is quite less, when compared to conventional biometric attendance system. The use of cloud computing to store the attendance records makes all the data easy to access and retrieve as end when required by the manager's. The use of fingerprint scanner ensures the reliability of the attendance record. The system, due to its lack of complexity, proves to be easy to use and user friendly.

REFERENCES

[1]. Tseng, Jerry CC, et al. "An interactive healthcare system with personal- ized diet and exercise guideline recommendation." 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE, 2015.
[2]. Shaikh, Shakila, Sheetal Rathi, and Prachi Janrao. "Recommendation system in E-commerce websites: A Graph Based Approached." 2017 IEEE 7th International Advance Computing Conference (IACC). IEEE, 2017.
[3]. Jiang, Ling, and Christopher C. Yang. "Personalized Recommendation in online health communities with heterogeneous network mining." 2016 IEEE International Conference on Healthcare Informatics (ICHI). IEEE, 2016.
[4]. Lee, Choon-oh, et al. "A framework for personalized healthcare service recommendation." HealthCom 2008-10th International Conference on e-health Networking, Applications and Services. IEEE, 2008.
[5]. X. Ma, J. Ma, H. Li, Q. Jiang, and S. Gao, "ARMOR: A trust-based privacy-preserving framework for decentralized friend recommendation in online social networks," Future Gen. Comp. Syst., vol. 79, pp. 82–94, 2018.
[6]. H. Hu, R. Lu, Z. Zhang, and J. Shao, "Replace: A reliable trust-based platoon service recommendation scheme in vanet," IEEE Trans. on Veh. Tech., vol. 66, no. 2, pp. 1786–1797, 2017.
[7]. K. Wang, Y. Shao, L. Shu, C. Zhu, and Y. Zhang, "Mobile big data faulttolerant processing for ehealth networks," IEEE Network, vol. 30, no. 1, pp. 36–42, 2016.
[8]. F. G. Marmol and G. M. Perez, "Trip, a trust and reputation infrastructure based proposal for vehicular ad hoc networks," Journal of Net. and Comp. App., vol. 35, no. 3, pp. 934–941, 2012
[9]. J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," IEEE Trans. Intelligent Transportation Systems, vol. 19, no. 8, pp. 2627–2637, 2018.
[10]. C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: an efficient and privacypre- serving disease prediction scheme in cloud-based e-healthcare system," Future Generation Comp. Syst., vol. 79, pp. 16–25, 2018.
[11]. C. Zhang, L. Zhu, C. Xu, K. Sharif, and X. Liu, "Pptds: A privacypre- serving truth discovery scheme in crowd sensing systems," Information Sciences, 2019.
[12]. C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Du, and M. Guizani, "LPTD: achieving lightweight and privacy-preserving truth discovery in ciot," Future Generation Comp. Syst., vol. 90, pp. 175–184, 2019.
[13]. R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," IEEE Access, vol. 5, pp. 3302–3312, 2017.
[14]. W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 5, pp. 1484–1496, 2016. [Online]. Available: https://doi.org/10.1109/TPDS.2015.2448095.