Journal of Biomedical Informatics 110 (2020) 103549

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Original Research

Building a trust-based doctor recommendation system on top of multilayer graph database

Safikureshi Mondal [a,*], Anwesha Basu [b], Nandini Mukherjee [c]

[a] Department of Computer Science and Engineering, Narula Institute of Technology, Kolkata 700109 West Bengal, India
[b] Software Analyst, Capgemini Technology Services, Bangalore, India
[c] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032 West Bengal, India

ARTICLE INFO

Keywords:
E-healthcare
Patient-doctor relationship
Multilayer graph data model
Trust Model
NoSQL
Neo4j

ABSTRACT

In healthcare applications, developing a data model for storing patient–doctor relationships is important. Though relational models are popular for many commercial and business applications, they may not be appropriate for modeling patient–doctor relationships due to their inherent irregular nature and complexities. In this paper, as a case study, we propose to build a doctor recommendation system for the patients. The recommendation system is built on top of a multilayer graph data model. Contemporary research papers have already shown that *multilayer* graph data models can be efficiently used in many applications where large, heterogeneous data are to be modeled. As part of the recommendation system, the paper also introduces a concept of *trust* which is one important ingredient of any kind of recommendation. The trust factor introduced in the paper exploits certain characteristics of the multilayer graph model. The paper also presents some analysis to demonstrate the efficiency of the graph data model in comparison with relational data model.

1. Introduction

The proliferation of e-health applications, and increased scope of delivering quality health-care remotely encourage the researchers to focus on building an appropriate model for patient–doctor relationships. It is generally recommended that a data model should closely reflect the data access patterns. That means, if the applications execute queries for accessing large number of records of similar type, then a relational database is suitable. On the other hand, when queries are executed to traverse through the complex and irregular relationships among entities, then some data model reflecting such relationships will suit best for the applications.

In practice, a patient visits doctors having different specializations depending on the patient's current symptoms. Whenever a patient visits a doctor, a treatment episode starts. Doctors diagnose the diseases and prescribe medicines to each patient or advise to go through few clinical tests. A patient may visit a doctor multiple times during one particular treatment episode. On the other hand, when a treatment episode ends between a patient and a doctor, a new treatment episode may again be started between the same pair of patient and doctor for a new set of symptoms. Ontological descriptions of healthcare domain [1] characterize the relationships among the entities (e.g. patient and doctor) and highlight their properties effectively. However, while translating the ontological description into relational schema, it is found that relational databases are unable to capture such complex relationships. The tables in relational databases store the elements in a particular set as rows in the table and thus, effectively express the relationships among the entities of the same type (such as a set of patients in a patient table). However, relational databases cannot effectively capture the relationships that exist among different types of entities, such as a patient and a doctor. In other words, relational databases cannot efficiently express the relationships among individual data elements. On the contrary, graph databases can nicely express the relationships among individual data elements by creating a node for each data element and connecting the nodes through edges in order to establish relationships among them. In a graph database, data can be modeled using a high number of data relationships. Such models are flexible enough to incorporate new data or data relationships. Moreover, querying data relationships in real-time is also possible [2–5].

In this paper, as a case study, we propose to build a doctor recommendation system for the patients. The recommendation system is intended to be used in a remote healthcare scenario for choosing a doctor from pool of available doctors. At the back end of the recommendation system, the patient–doctor relationship is modeled as a graph. As there can be multiple treatment episodes between patients and doctors and doctors have different specializations, the graph model proposed in this paper is *multilayer*. It has already been shown that *multilayer* graph

* Corresponding author.
E-mail addresses: msafi.cse@gmail.com (S. Mondal), anweshabasu4@gmail.com (A. Basu), nmukherjee@cse.jdvu.ac.in (N. Mukherjee).

https://doi.org/10.1016/j.jbi.2020.103549
Received 10 April 2020; Received in revised form 25 August 2020; Accepted 26 August 2020
Available online 29 August 2020

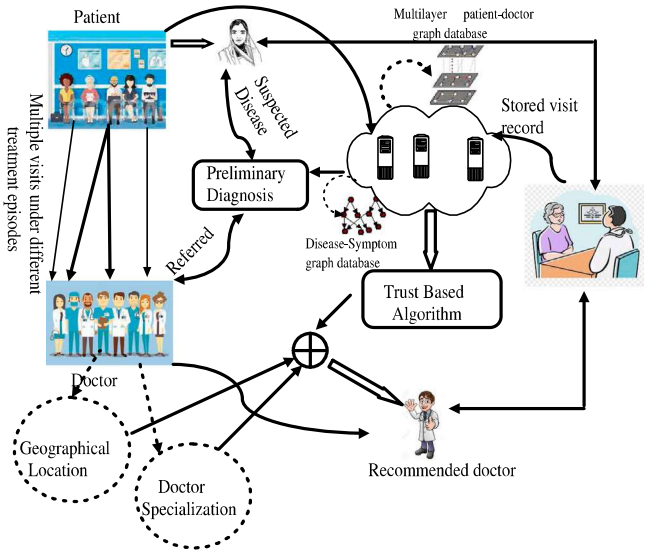

Fig. 1. A doctor recommendation system in healthcare.

data models can be efficiently used in many applications where large, heterogeneous data are generated [6–9].

Alongside, *trust* has a key role in any recommendation system. In day-to-day life, the concept of trust is used to express patient's level of satisfaction with the doctors. It has been observed that patient's trust to a doctor improves quality of care and satisfaction of the patient, and is also effective for better health outcomes [10–13]. Nowadays, 'trust in recommendation system' is an important research topic in social network analysis paradigm. Similarly, in doctor recommendation system also *trust* factor must be given emphasis while dealing with remote health-care as well as general health-care system. In our recommendation system, a measure related to the cumulative trust is associated with each doctor. This measure increases with patients' satisfaction. We propose an algorithm to build the trust factor on the basis of the patients' choice. That means the trust factor is modified every time when a patient decides to revisit the same doctor for a different treatment episode or a patient decides to visit another doctor for the same set of symptoms. Thus, the system takes into account the patients' preference or disfavor for a doctor and records this as trust factor. When a patient enters his or her symptoms to the system and searches for a doctor, the search takes place on the basis of the preliminary diagnosis of the patient based on symptoms, specializations of doctors and the trust factors.

Thus, the contribution of this paper is two-fold. We first propose a multilayer graph model for patient–doctor relationships. We also demonstrate the efficiency of our graph data model by comparing it with the popular relational data model to map the patient–doctor relationships. The second contribution is developing the concept of a trust factor for using it in a recommendation system. The recommendation system considers three factors — *the specialization of the doctor* based on patient's symptoms, *location of the doctor* (in case of remote healthcare this factor is not important), and the *trust factor* as presented in this paper. Based on these three factors, the system recommends a doctor for a patient. The trust factor introduced in this paper is heavily dependent on the underlying multilayer graph database and dynamically updated for any change in the database. Fig. 1 provides an overview of the system.

The remainder of the paper is organized as follows. A review of the application of *multilayer* data models and the concept of using *trust factor*, which are important contributions in this paper, is carried out in Section 2. In Section 3, the *patient–doctor multilayer graph* data model is proposed and validated by implementing it on a *NoSQL Neo4j* [14] graph database with a sample dataset. Our proposed *multilayer patient–doctor* graph data model is compared with relational data model in Section 4 and its efficiency is demonstrated. In Section 5, a *trust model* is proposed for a recommendation system based on *multilayer patient–doctor* graph database. Analysis of the proposed model is presented in Section 6. Section 7 discusses experiments with a real dataset and the results and analysis through some case studies. Limitations of this research work is discussed in Section 8 and finally the paper is concluded in Section 9.

2. Related work

Efficient data model for large, heterogeneous *patient–doctor* relationship data and suitable algorithm for computing *trust factor* are two major issues handled in this paper. Therefore, in this section we focus on the contemporary research works in these two areas.

Patient–Doctor Data model: A lot of research works [15–18] focused on modeling *patient–doctor* or *doctor–patient* relationships. The idea of *patient–doctor* relationship is not a new one. But technical implementation of such type of data has not been done in many healthcare or remote healthcare domains. In [15], the authors depicted three models, these are activity–passivity, guidance–cooperation and mutual participation. The activity–passivity and guidance co-operation models are entirely paternalistic and thus predominantly doctor-centric. In [17], the author reviewed the patient–doctor relationship. In this paper, the author also reviewed the bioethics model and clinical model. Bioethics and clinical models are normative and prescriptive in nature being derived from various ethical and social theories. In [19], the authors presented a cloud based system for the relationship of patient, doctor, disease and symptom. In this paper, the authors proposed a system to predict appropriate diseases based on the symptoms and then display the list of specialist doctors in nearby areas using k-mean clustering. In [18], the authors proposed a graphical model-based approach to predict target physicians by jointly modeling physician and patient features from different data spaces and utilizing the extra relational information through factor graph. In all the above-mentioned research papers, authors do not mention how the patient, doctor data along with their clinical history are stored in the cloud, i.e. the technical implementations of the data models are not detailed out.

Some research works consider graph models for other applications. In [20], the authors have introduced *Hetionet* biological graph database which integrates all biological objects like *disease*, *symptom*, *gene* etc. It has almost *50* thousand nodes of *24* types and approximately *2.25* million relationships. However, it focuses on a completely different domain.

In other research works, multilayered graph data models are used for storing or manipulation of large heterogeneous networks, like social networks, information networks, technological networks and biological networks [21,22] and as well as different applications [23,24], e.g., *Online-social network*, *Twitter* and *Transport network* etc. Many research works have been published [19,23–38] in recent times where *multilayer* data model concepts are used. Variations of *multilayer network* have been studied in [30]. The variations are shown in Fig. 2.

Trust in Doctor Recommendation System: *Trust* and *trustworthiness* are used in different applications related to *Online Communities* [39–41] such as *Social network* [42] and *Security* [43], *Health Trust* [44–49], *E-commerce* [50–53] and *Semantic Web* [52–56] (such as *web-Resources* and *e-mail*).

A state of the art of *trust* modeling is shown in Fig. 3. The figure shows various applications where *trust* is used. In healthcare domain, there are several recommender systems which are efficiently used by different types of users. For example, assisting decision making process in personalized care [57], identifying important opinions of medical practitioners [58], finding preventative healthcare for planning personalized therapy [59], providing personalized healthcare guidance [60]

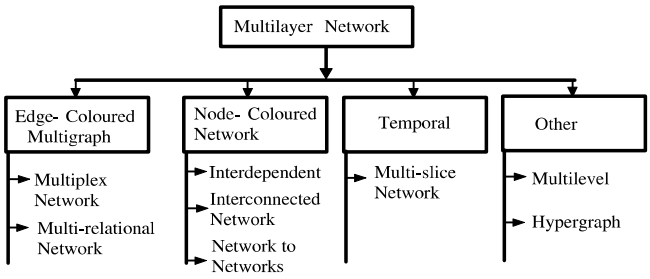

Fig. 2. Variation of multilayer network [30].

etc. In recent research, patients are recommended to doctors based on their previous consultation history as implemented in [61,62].

From the implementation viewpoint, recommendation systems are developed as collaborative filtering (CF) based, content-based or hybrid based using machine learning approaches. Recommendation systems are used in clinical settings to assist health professionals, family members, patients and caregivers [63,64]. It has been observed that different recommendation systems face important challenges; such as, for large and sparse data, there are some performance related challenges of recommendation, cold start problems arise due to sparse data, high complexity of algorithm, scalability problem as database grows etc. Hence, filtering methods are not appropriate and useful in applications, where huge heterogeneous real data is generated.

In our case, patients' visits to the same doctor multiple times in a treatment episode imply patients' positive opinions. However visiting another doctor for the same treatment episode implies the patients' negative opinions. Thus, for computation of the trust factor, no external data is needed and it entirely depends on the underlying multilayer graph. The trust factor is also updated with any change in the data. Moreover, the above-mentioned widely used recommendation systems always predict specified number of *top-n* results. In that sense, our requirement is also different. The algorithm proposed in this paper finds the best trusted doctor for a patient. In contrast with the above research works, our paper primarily focuses on *patient–doctor* relationship data model and use this model to build a trust model between a set of patients and doctors.

3. Overview of patient–doctor data model

Table 1 depicts the commonly used symbols in this paper. In this paper, the main focus is on building a patient–doctor data model. Whenever a patient becomes sick, the patient visits a doctor. A treatment episode starts from this instance and one or multiple visits are required for treatment of the particular disease. A treatment episode is constituted of all these visits. An ICT-based healthcare system requires to store large volume of heterogeneous information related to patients, doctors and treatment episodes for different types of users including healthcare personnel, patients, management and Government. This information includes patient history along with every past visit. In order to store this information, a suitable data model of *patient–doctor* relationship involving treatment episodes is needed.

Table 1
Symbols and descriptions.

Symbols	Descriptions
P_i	ith number of *patient* nodes
D_j	jth number of *doctor* nodes
T_k	kth number of *time* nodes
G^M	M-*multilayer*s Graph
V^M	M-*multilayer*s Vertices
E^M	M-*multilayer*s Edges
M	Total number of treatment episode layer
m	m_{th} treatment episode layer
u	Total number of *patient* nodes
v	Total number of *doctor* nodes
w	Total number of *time* nodes
T_f	*trust factor*
m	*Treatment episode layers where particular doctor is visited by patient*
x	x_{th} Doctor node
$X(ELT)$	Earlier visit
$Y(SFT)$	Number of shift visit
R	Max possible patient visit for all treatment episodes
$r_1, r_2 ... r_m$	Max patient visits in individual treatment episode

3.1. Characterization of patient–doctor relationship

Patient to doctor visit information and the *patient–doctor* relationships can be characterized as follows:

1. A patient can visit a particular doctor multiple times under any treatment episode.
2. If any patient visits a doctor at a particular time under a particular treatment episode, other patients cannot visit the same doctor at the same time for any other treatment episode.
3. Also patients can change their decisions to visit specific doctors under particular treatment episodes due to dissatisfaction and can go to other doctors of their choices with previous constraints for the same treatment episodes.

Based on the above observations, we propose a multilayer graph to model *patient–doctor* relationships as discussed in the following section.

3.2. Multilayer graph-based approach

Graph oriented database management systems are designed to facilitate relationships between the nodes. Instead of using foreign keys to represent a relationship, graph databases use arcs that directly connect two related nodes. Operations on this model can be performed through a graph query language. In this research work, we define multiple layers of treatment episodes, each containing three types of nodes. The nodes are either patient nodes or doctor nodes related through a third type of nodes, i.e. time nodes, which represent the visit time of a doctor to a patient. The requirements of a data model for *patient–doctor* relationships is to store necessary information about patients, doctors, and their visiting times along with patients' medical history, as well as severity of symptoms and many other details. As a solution to this problem, a *multilayer patient–doctor graph* data model is proposed. In this data model, a *patient* node and a *doctor* node are connected

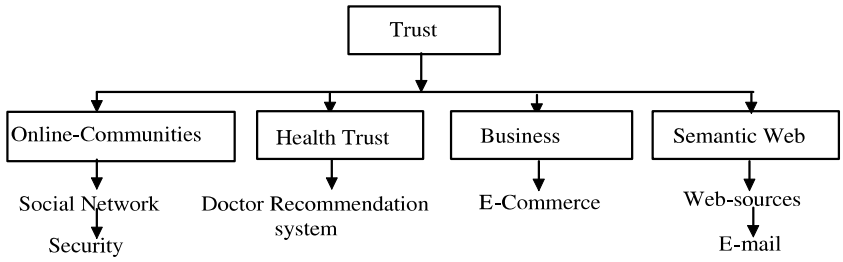

Fig. 3. Different applications in where *trust* is used.

Table 2
Doctor node properties (Names have been changed).

D_id	Name	Gender	Age	Address	PH	Specialization	Visiting Hours
D1	Dr,. abhijit Bhattacharaya	M	42	Kolkata	7766693812	General Medicine	Mon: 5 PM, Wed 9 AM
D2	Anirban Das	M	50	Howrah	9438390671	General Medicine	Tue: 3 PM, Wed: 9 AM
D3	Dr. Arijit Das	M	45	N 24 PGS	765093880	General Medicine	Wed: 9 AM, Thurs: 5 PM
D4	Dr, P.K. Moitra	M	35	Kolkata	8011614865	General Medicine	Tue: 8 PM, Wed: 9 AM

Table 3
Patient node properties (Names have been changed).

P_id	Name	Gender	Age	Address	PH	Emergency Contact	Danger Sign	History
P1	Amit Pal	M	56	Newtown, Kolkata	8825577841	9474009591	High Fever	Allergic
P2	Nishu Raj	M	76	Jadavpur, Kolkata	8007971252	9474009391	High Fever with Cough	Non-Allergic
P3	Debjit Dutta	M	77	Kolkata	7980369747	9474009395	Loose motion with blood	Allergic
P4	Arshi Naaz	F	55	Kolkata	8622602912	9474009495	High Fever	Non-Allergic

through a set of *time* nodes. The edges connecting these nodes form a set of unique directed edges under each *treatment episode*. The multilayer model is built with a number of *treatment episode* layers, where each layer corresponds to a specific disease. Doctors having appropriate specialization can handle those treatment episodes and therefore, can reside only on those layers which are relevant for their specializations. When a patient with specific symptoms wishes to visit a doctor with relevant specialization, the patient node is placed in the proper treatment episode layer, a time slot is allotted for visit and two directed edges are created to connect the nodes. The directed edges connect *patient* nodes to *time* nodes, and *time* nodes to *doctor* nodes. However, *patient* nodes are not directly connected to the *doctor* nodes. Moreover, any *patient* node cannot connect to *doctor* node at two different treatment episode layers through a common *time* node. Also, any doctor node cannot be connected to two different patient nodes at two different treatment episode layers through the same time node. The above mentioned constraints are imposed based on the characterization of patient–doctor relationship as discussed in the previous section. A formal definition of the model is given in the next section.

3.3. Definition of proposed multilayer patient–doctor graph data model

The *patient–doctor* data model is defined as a *multilayer graph* which is shown in Fig. 4. The *multilayer patient–doctor* graph data model is defined as a graph $G^M = (V^M, E^M, M)$ in where $V^M = (P_u^M, T_v^M, D_w^M)$ and $E^M = E_1^M \cup E_2^M$. Here:

1. M represents the treatment episode layers; TE_1, TE_2, TE_3, $\ldots..TE_M$ treatment episode layers, where each treatment episode corresponds to a disease.
2. V^M: V is the set of P, T and D nodes; P_1, P_2, P_3, $\ldots.P_u$ are the *Patient* nodes; T_1, T_2, T_3, $\ldots.T_v$ are the *Time* nodes; and D_1, D_2, D_3, $\ldots.D_w$ are the *Doctor* nodes.
3. E^M: the set of directed edges connecting nodes in P, T and D, such that $P \rightarrow T \rightarrow D$.
4. E_1^M: the set of directed edges from nodes in P to nodes in T. Thus, E_1^M is the set of edges e_1^l connecting vertices P_i^l and T_k^l, where $P_i^l \in P_u^l$ and $T_k^l \in T_v^l$ and $l \in M$, i.e. the lth layer.
5. E_2^M: the set of directed edges from nodes in T to nodes in D. Thus, E_2^M is the set of edges e_2^l connecting vertices T_k^l and D_j^l, where $T_k^l \in T_v^l$ and $D_j^l \in D_w^l$.

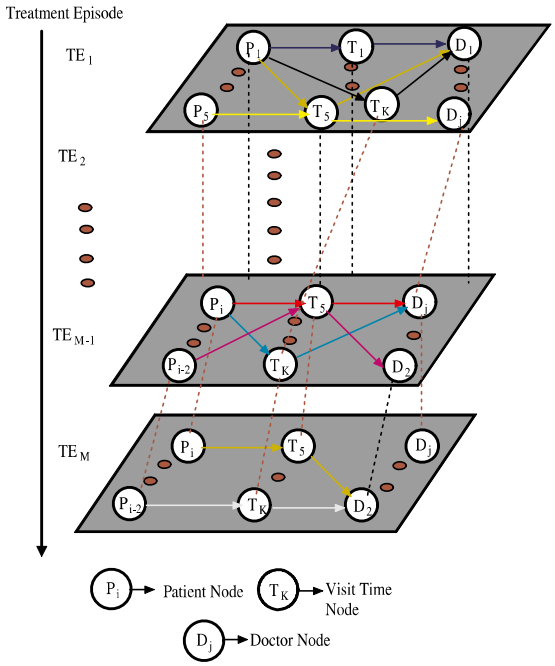

Fig. 4. Proposed *multilayer patient–doctor data model* (to show the uniqueness of each visit, the directed edges are represented using different colors or line style). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Thus, at each layer, two bipartite graphs, $G_1^l = (P_u^l, T_v^l, E_1^l)$ and $G_2^l = (T_v^l, D_w^l, E_2^l)$ are stored.

According to the *multilayer patient–doctor* graph data model, every patient P_i visits a doctor D_j at time T_k at the TE_l treatment episode layer. P_i, D_j and T_k nodes create a directed edge $P_i^l \rightarrow T_k^l \rightarrow D_j^l$ in a treatment episode layer or plane (TE_l). Also, every directed edge $P_i^l \rightarrow T_k^l \rightarrow D_j^l$ must be unique at every treatment episode layer. In other words, no two directed edges should connect the same set of nodes. In order to distinguish between the visits by different patients to different doctors or to show the uniqueness of each visit, the directed edges are represented with different colors and line style. In *multilayer patient–doctor* data model, a number of treatment episode layers for a patient

Table 4
Time node properties.

Time_id	Date	Day	Time
T1	03/01/2018	Wednesday	17:10
T2	10/01/2018	Wednesday	17 :20
T3	17/01/2018	Wednesday	17 :50
T4	24/01/2018	Wednesday	18 :20
T5	31/01/2018	Wednesday	17 :30

Table 5
Records of database.

D_id	P_id	Date	Time	Date and Time
D1	P1	03/01/2018	17:10	T1
D1	P1	10/01/2018	17 :20	T2
D1	P1	17/01/2018	17 :50	T3
D2	P2	10/01/2018	17 :20	T2
D3	P2	24/01/2018	18 :20	T4
D3	P3	03/01/2018	17:10	T1
D2	P3	17/01/2018	17 :50	T3
D2	P1	24/01/2018	18 :20	T4
D2	P2	03/01/2018	17:10	T1
D4	P2	10/01/2018	17 :50	T3
D3	P3	31/01/2018	17 :30	T5
D4	P3	24/01/2018	18 :20	T4
D2	P1	31/01/2018	17 :30	T5
D3	P4	31/01/2018	17 :20	T2
D1	P4	24/01/2018	18 :20	T4
D3	P2	17/01/2018	17 :50	T3

is possible and it depends on the history of the patient, because the database is dynamically updated.

3.4. An example of multilayer patient–doctor database

An example of a *multilayer patient–doctor* graph database is shown in Fig. 5. The example depicts a small portion of a large dataset. Corresponding patient information, doctor information, time information and their visiting times are shown in Tables 2–5 respectively. Table 5 provides the key information for building the graph database. It may be noted that the records shown in the tables are taken from the supplementary materials, however, the names shown in these tables are edited for the sake of privacy.

In this example, only three treatment episodes are considered and total 4 patient nodes, 5 time nodes and 4 doctor nodes are present in the dataset. The set of directed edges are maintained following the requirements of *patient–doctor* relationship model.

3.5. Validation of data model

The *multilayer patient–doctor* graph database is validated using *Neo4j NoSQL* graph database and by executing a set of cypher queries after assigning property values to patient, doctor and time nodes. The patient node properties, doctor node properties and Time node properties are shown in Tables 2, 3, and 4 respectively. Before going into the implementation details, it should be necessary to create an equivalent *multidimension* or *multilayer* graph, which is shown in Fig. 6.

Some edge properties are defined from *patient* nodes to *time* nodes and *time* nodes to *doctor* nodes to satisfy uniqueness of the edges. All the visit edges ($P_i \rightarrow T_k \rightarrow D_j$), which belong to a particular *treatment episode layer*, add relevant *time* nodes to the layer related to a *suspected disease*. These edges are unique because each directed edge refers to an individual visit in a *treatment episode* plane.

In order to validate the data model, a set of seven cypher queries is tested on the sample graph database. Each query has been executed as a test and the resulting graphs are displayed. The queries in natural language along with their cypher version are shown from Query 1 to Query 7 and the results are shown from Figs. 7 to 13.

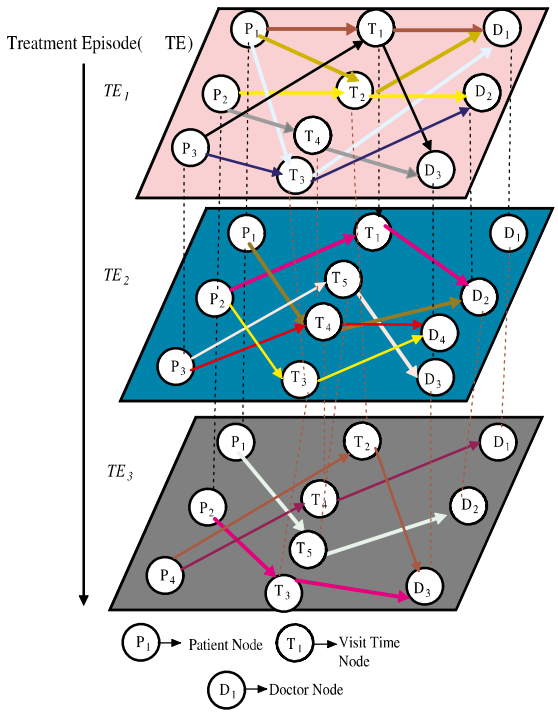

Fig. 5. An example of *multilayer patient–doctor* data model (to show the uniqueness of each visit, the directed edges are represented using different colors or line style). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

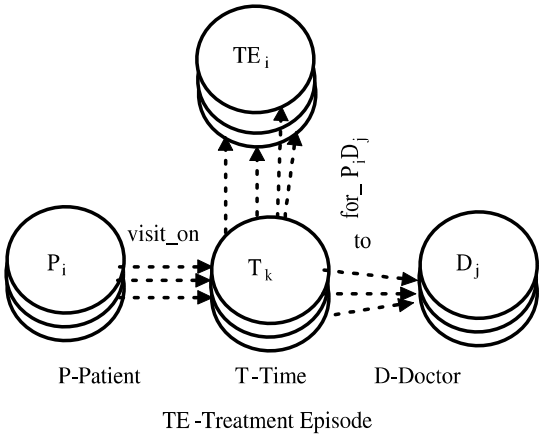

Fig. 6. Equivalent *multilayered patient–doctor* relationship.

Query 1: Show all patients who visited doctors multiple times under all the treatment episodes.
Q1: MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$, $(b : Time) - [] \rightarrow (e : Disease)$ return a, b, c

Query 2: Show all patients, doctors along with their visiting times in a particular treatment episode
Q2: MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$, $(b : Time) - [] \rightarrow (e : Disease)$ where e.Name="Treatment Episode1" return a, b, c, e

Query 3: Show the doctors and visit times for a particular patient.
Q3: MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [] \rightarrow (c : Doctor)$ where a.Name="Debjit Dutta" return a, b, c

Table 6
Queries on relational database and Neo4j.

Queries	Relational	Neo4j
Q1: Show all patients who visited doctors multiple times under all the treatment episodes.	SELECT DISTINCT p_id, d_name, suspected_disease, cnt FROM (SELECT count(*) cnt, p_id, d_name, suspected_disease FROM RECORD GROUP BY p_id, d_name, suspected_disease)tab WHERE tab.cnt>1	Q1 query as test 1
Q2: Show all patients, doctors along with their visiting times in a particular treatment episode	SELECT p_name, d_name, suspected_disease, visit_date, visit_time FROM RECORD re JOIN PATIENT_PAPER pa ON (pa.p_id = re.p_id) WHERE suspected_disease='Cataract'	Q2 query as test 2
Q3: Show the doctors and visit times for a particular patient.	SELECT p_name, d_name, suspected_disease, visit_date, visit_time FROM RECORD re JOIN PATIENT_PAPER pa ON (pa.p_id = re.p_id) WHERE pa.p_name = 'AAGHAZ AHMED'	Q3 query as test3
Q4: List all patients who visited a doctor on a particular date	SELECT p_name, re.visit_date, re.visit_time, re.d_name, re.suspected_disease FROM RECORD re INNER JOIN (SELECT COUNT(p_id) cnt,p_id, visit_date, suspected_disease FROM RECORD GROUP BY visit_date, p_id, suspected_disease)tab ON (tab.visit_date = re.visit_date AND tab.p_id = re.p_id) JOIN PATIENT_PAPER pa ON (pa.p_id = tab.p_id) WHERE cnt>1 ORDER BY visit_date,p_name	Q4 query as test 4
Q5: Show all patients along with the doctors visited by each of them and the total number of visits.	SELECT tab.p_id, tab.d_name, pa.p_name FROM (SELECT DISTINCT re.p_id, re.d_name FROM (SELECT p_id, d_name FROM record)re LEFT JOIN (SELECT p_id, p_name, d_id, d_name FROM patient_paper, DOCTOR_paper)tab ON(tab.p_id = re.p_id AND tab.d_name = re.d_name))tab JOIN patient_paper pa ON(pa.p_id=tab.p_id) ORDER BY tab.p_id	Q5 query as test 5
Q6: List all patents who visited Dr. Arijit Das along with the frequency of visits.	SELECT re.p_id, pa.p_name, re.d_name, re.suspected_disease, re.visit_date, re.visit_time FROM RECORD RE JOIN (SELECT count(*) cnt, d_name, p_id FROM RECORD r group by d_name, p_id)tab ON (tab.d_name=re.d_name AND tab.p_id= re.p_id) JOIN PATIENT_PAPER pa on (re.p_id= pa.p_id) WHERE tab.cnt>1 ORDER BY tab.p_id, visit_date	Q6 query as test 6

Query 4: List all patients who visited a doctor on a particular date.
Q4: MATCH $(a : Patient) - [r : visit_on] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$ where b.Date='3/01/2018' return a, b, c

Query 5: Show all patients along with the doctors visited by each of them and the total number of visits.
Q5: MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$) return a, Name as Patient, collect(c.Name) as Doctor, count(c.Name) as frequency

Query 6: List all patents who visited Dr. Arijit Das along with the frequency of visits.
Q6. MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$ where c.Name="Dr. Arijit Das" return a.Name as Patient , count(c.Name) as frequency.

Query 7: Show the dates of visits of a particular patient to a particular doctor in a treatment Episode layer.
Q7: MATCH $(a : Patient) - [r] \rightarrow (b : Time) - [r1] \rightarrow (c : Doctor)$ where e.Name="Treatment Episode1" AND c.Name="Dr. Abhijit Bhattacharaya" AND a.Name="Amit Pal" return a, b, c, count(b.Date) as frequency

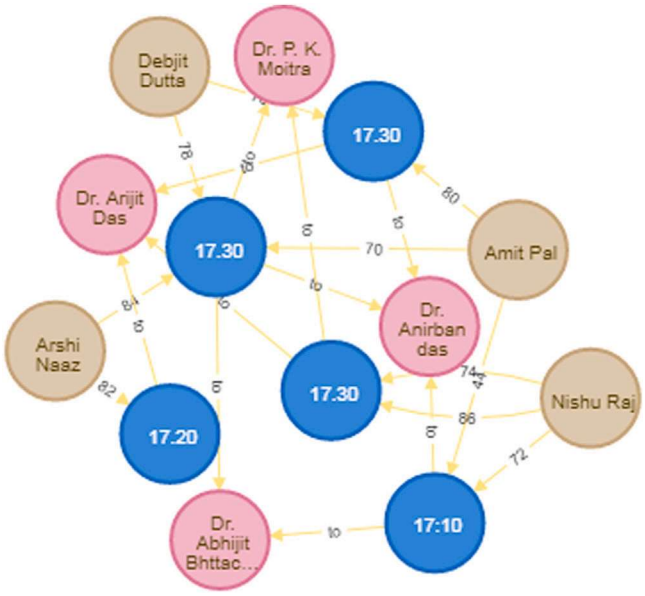

Fig. 7. *Q1 query result (Names have been changed).*

4. Comparison between relational and graph based patient–doctor data model

Patient–Doctor data model can be built using relational based and graph based approaches. In this section, a comparative analysis is carried out for both relational *patient–doctor* database and *multilayer patient–doctor* graph database in order to decide an efficient data model representation for *patient–doctor* relationships.

4.1. Relational patient–doctor data model

The relational model represents data as a collection of relations, and a relation is essentially a table of values where each row represents a set of related values, column headers represent the attributes and every row shares the same set of attributes, but not their values. In real life situation, patient–doctor data is combination of structured and unstructured data. However, in a relational approach unstructured data is represented in structured form, that is as a set of normalized relations. According to our dataset, it is represented with seven tables. The ER diagram is shown in Fig. 14.

4.2. Implementation and analysis

The efficiency of the graph data model is demonstrated by comparing it with the relational model. Complexity of the performing queries in each approach is discussed first followed by some experimental studies .

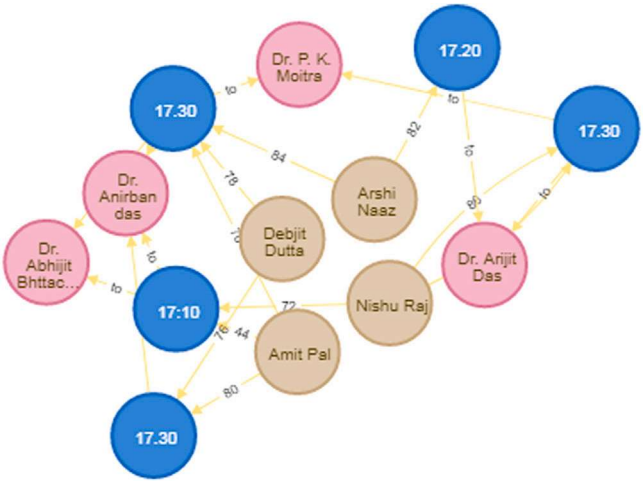

Fig. 8. *Q2 query result (Name have been changed).*

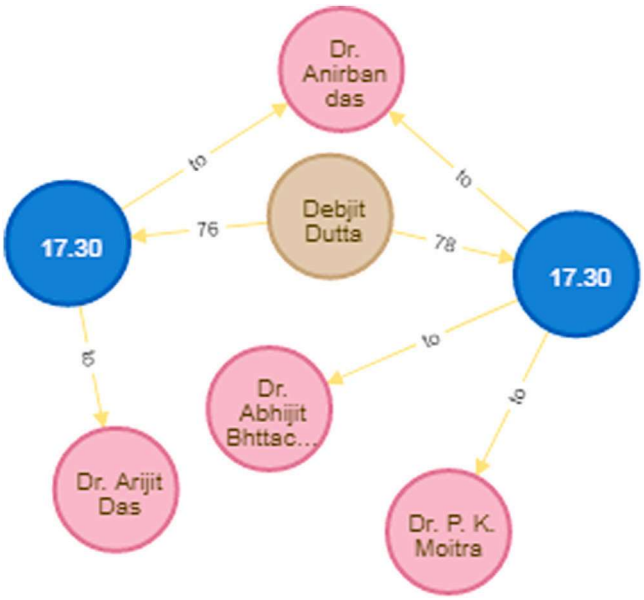

Fig. 9. *Q3 query result (Names have been changed).*

4.2.1. Complexity of database queries

In order to measure the complexity of database queries when executed on both, relational and graph based data models, the following operations are used in the same order [65]:

- *NodeIndexSeek*, *ExpandAll*, *Filter*, *Limit* and *Projection* functions are used in the graph model.
- *Nested Loops* or *Join* and *Select* are used for relational model.

Projection returns the rows after evaluation of expression, *Limit* returns the specified rows as result, *Filter* filters the result with condition, *ExpandAll* traverses the result and *NodeIndexSeek* finds the desired result by given index. In the other case, i.e. in relational model, *Nested Loops* is used for *Join* and it is used for both patient (with *KeyLookUp* operation) and doctor (with *Index Seek* operation) tables. *Select* is used to return the result satisfying specific condition. A common operation in such a database will be "to find all patient–doctor relationships for a given doctor id". Let n be the number of patients and m be the number of doctors available in both databases. Then the complexity can be measured as follows:

Fig. 10. *Q4 query result (Names have been changed).*

1. To run this particular query in relational approach, two operations are needed. First, a *Join* operation will take place. If it is a simple *natural join*, then the complexity will be $O(m * n)$. Considering that both tables are large, it is not possible to join them in-memory. Also we consider that the patient table and the doctor table are indexed on the patient-id and doctor-id columns respectively. The doctor-id is a foreign key in the patient table. So only one table has an index on the joined column. Therefore, the complexity will be $O(n + mlogm)$. In the worst case the number of records in the joined table will be $O(mn)$. Next, the given doctor-id will have to be searched from the resultant table and all relevant records are to be retrieved. In the relational approach, to search a record from a table, B-tree data structure is used. Hence, the complexity is $O(logmn)$. If k number of patients are retrieved, then the complexity of this operation is $O(k * logmn)$. Thus, complexity of retrieving the relationships is $O(n + mlogm)+O(k * logmn)$.
2. Similar to relational approach, graph database uses B-tree index to find a record with given doctor-id from the set of doctor nodes and the complexity is $O(logm)$. Next, graph database performs *expandall* and *filter* operations for the k number of nodes/relations and in this case, complexity is $O(2 * k)$. Thus, the complexity to search all patients who visited a particular doctor is $O(logm)+O(2 * k)$, which is much lower than the previous case.

4.2.2. Experimental study

In order to compare the relational and graph data models, a set of queries are executed and their performances are observed. MySQL and PostgreSQL are chosen as representatives of relational databases, because of their use in wide range of applications. Neo4j is used as a representative graph database. Six of the previously discussed seven queries (Section 3.5) are used for these experiments and are shown in Table 6. The patient–doctor relational data model is implemented as seven tables consisting of patient–doctor data of 5300 records which have been collected from different health organizations. For privacy and security, all the names of patients and doctors with contact numbers are replaced with other names. Same queries are tested and time and space are measured in both environments. The access time depends on sub-queries or co-related queries, join operations and number of tables used in a query in relational data model. The whole experiment is done using *MySQL*, *Posegresql* and *Neo4j4.0* in *64* bit windows machine with *Intel(R)core(TM)i3-5010U CPU@2.10 GHZ* processor and 4 GB

Fig. 11. Q5 query result (Names have been changed).

Fig. 12. *Q6 query result (Names have been changed).*

Fig. 13. *Q7 query result (Names have been changed).*

Table 7
Memory requirements for loading the entire database.

Dataset	*Neo4j*	*PostgreSQL*	*MySQL*
Record	…	424 kb	448 kb
Patient	..	200 kb	128 kb
Doctor	.	40 kb	16 kb
Nodes with relationship	265 KB	..	…

Table 8
Join and sub queries for given queries in relational databases.

Queries	*No. of Join operations*	*No. of sub Queries*	*Order by/ Group by*
Q_1	–	1	1
Q_2	1	1	–
Q_3	1	–	–
Q_4	2	2	2
Q_5	2	3	1
Q_6	2	1	1

RAM. For all the queries, number of sub-queries, join operations and Group by or Order By clauses used are shown in Table 8. Query execution times for the six queries are shown in Fig. 15. The access time for query Q_1 is almost same for all three databases. But, in case of other five queries, the access times for the graph database is much lower compared to the relational databases. This phenomenon is the result of the presence of multiple join operations in the relational database queries. The query Q_5 contains more join operations and subqueries compared to others, and as a result more access time is required for relational databases to produce the result.On other hand, Neo4j graph database requires larger memory to load the graph data compared to the relational databases when same data is stored. The result is shown in Table 7.

5. Modeling *trust factor*

In any healthcare system, a patient visits the doctor one or multiple times. Based on the patient's satisfaction about the treatment, the trust of the patient bestowed upon the doctor increases. In a trust based recommendation system, a measure related to the cumulative trust bestowed upon a doctor by the patients is used as a trust score. When a patient searches for a doctor, the system recommends a set of doctors according to the preliminary diagnosis of the patient, based on the current symptoms (this search is based on the specialization of the doctors) and the trust factor. In third section, we propose an algorithm for modeling trust factor. In practical situation, patients draw up their own feedback regarding the doctors based on several facts and ideas. However, they do not share this feedback in most cases and has no provision to know the general perception related to the doctor. In our recommender systems, we introduce a trust value to assist a patient to make an informed choice. The trust value needs to be updated dynamically after every visit of a patient to a doctor. In our *multilayer patient–doctor* data model, various constraints affect the doctor's *trust factor*. Thus, *Trust factor* of a doctor is dependent on the following four data:

1. Number of treatment episode layers in which patient visits the doctor.
2. Number of earlier patient visits to any particular doctor for whom trust factor is required to be computed.
3. Number of maximum possible visits in each of the individual treatment episode layers and maximum possible visits in all treatment episode layers as a whole in which a particular doctor is visited by the patients.
4. Number of "shifts" that have been made by patients after visiting a particular doctor. By "shift" we mean a patient, while visiting a doctor for treatment of a disease during a particular treatment episode decides to change the doctor and chooses a new doctor due to dissatisfaction.

5.1. Overview of the trust in healthcare

Trust is an important factor in *patient–doctor* relationship. The meaning of trust in any medical field is to believe someone (in this

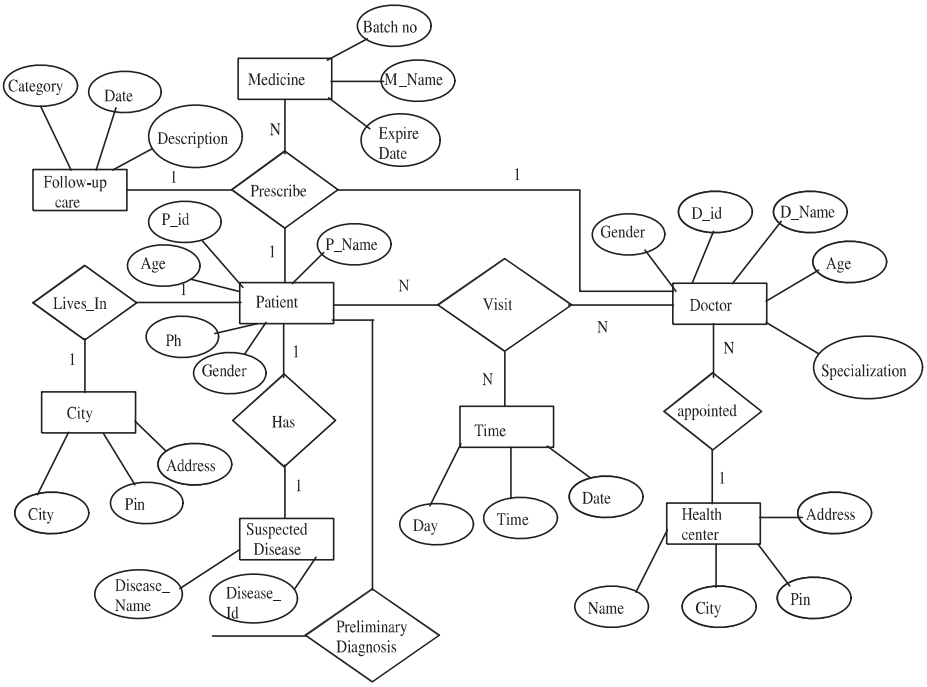

Fig. 14. ER Diagram for Patient–Doctor dataset.

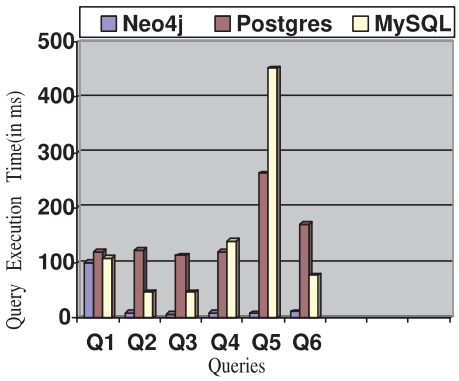

Fig. 15. Access time comparison of relational and graph data model on patient–doctor data.

case, doctor) and this belief must be good or honest [66]. In any healthcare system, patients expect that the doctor is competent, honest, empathetic in their good will and expect a good outcome from every visit [67]. Also it can be said that the *patient–doctor* relation is built by both personalities and social systems and trust occurs from their interactions [68]. There are some research works which already have focused on the measure of *trust* in healthcare system based on *patient–doctor* relation. But these works mostly calculated trust factor by assigning values to different characteristics of the patients and doctors. On the other hand, in our recommender system, there is no need to collect external data. Rather, in this system, trust is built and updated based on the data input to the system through its different actions.

In *patient–doctor* relationship, every patient forms opinion about a doctor based on the satisfaction gained during a treatment. According to our proposed *multilayer patient–doctor* data model, every patient visit is represented by the $P_i \rightarrow T_k \rightarrow D_j$ directed edge which is shown in Fig. 4. The total number of visits by all patients to a particular doctor in all treatment episode layers with some symptoms or diseases are termed as *earlier visits (ELT)* for this doctor and shifting to other doctors by these patients for the same symptoms or diseases in all treatment episode layers is termed as *shift visit (SFT)*. To calculate the *trust factor* of a doctor, the following steps are proposed:

1. Calculate the total number of earlier visits for a particular doctor for whom *trust factor* will be calculated.
2. Find the maximum possible visits in the treatment episode layers where the particular doctor is present. Also find the maximum possible visits at each individual layer.
3. Next, number of changes for a doctor is computed for the patients who previously visited the particular doctor (for whom *trust factor* is being calculated) and then shifted to another doctor.
4. Finally, *trust factor* is calculated as the differences between earlier visits relative to the maximum visits at all treatment episode layers and sum of all doctor changes at all individual layers relative to the maximum possible visits at individual treatment episode layers.

The algorithm is further elaborated in the next subsection.

5.2. Trust evaluation

In this research work, *trust factor* is measured from the visit history in the *patient–doctor* database and is dynamically updated. Before detailing the computation of trust factors, two theorems are discussed.

Theorem 1. *If P_1, P_2... P_u are the u number of patient nodes, D_1, D_2.... D_v are v number of doctor nodes, and T_1, T_2.. T_w are w number of allocated time slots present in a single treatment episode layer, then the total number of maximum possible visits in a single treatment episode layer is* $\min(P_{i=1,2,\ldots,u}, D_{j=1,2,..v}) \times (T_{k=1,2,..w})$ *or min(u, v)* $\times w$.

The theorem is proved as follows.

Every single visit is represented by three nodes, i.e. $P_i \rightarrow T_k \rightarrow D_j$, where i, j and k are positive integers. If in a single treatment episode layer, number of patients, doctors and time nodes are u, v and w respectively, then total number of possible visits is represented by $P_{i=1,2,\ldots,u} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1,2,..v}$ in that layer.

Now, a path $P_{i=1,2} \rightarrow T_{k=1} \rightarrow D_{j=1}$ is not possible as only one patient can use the time slot to visit the doctor, i.e. total visit= 1.

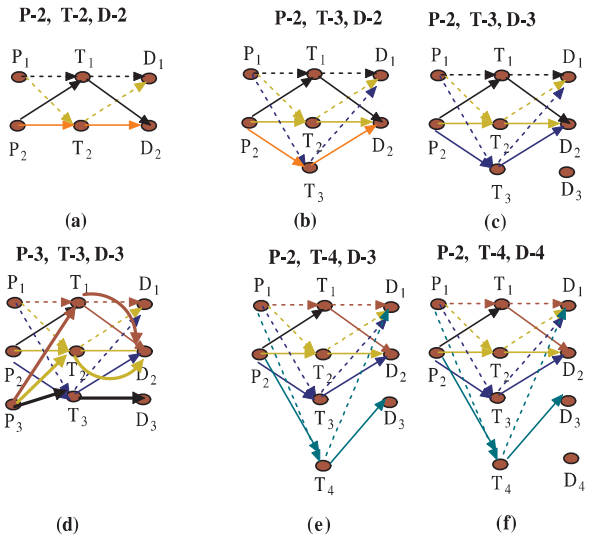

Fig. 16. Example of single layer *patient–doctor* data model: (a) $P = 2$, $T = 2$ and $D = 2$, maximum possible edges are 4. (b) $P = 2$, $T = 3$ and $D = 2$, maximum possible edges are 6. (c) $P = 2$, $T = 3$ and $D = 3$, maximum possible edges are 6. (d) $P = 3$, $T = 3$ and $D = 3$ maximum possible edges are 9, (e) $P = 2$, $T = 4$ and $D = 3$ maximum possible edges are 8, and (f) $P = 2$, $T = 4$ and $D = 4$, maximum possible edges are 8. (to show the uniqueness of each visit, the directed edges are represented using different colors or line style). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

If number of time nodes increases, i.e. visits are represented by $P_{i=1,2} \rightarrow T_{k=1,2} \rightarrow D_{j=1}$, then total visits= 2, because two patients can visit the doctor individually at two different time slots T_1 and T_2.

If we increase time slots for one doctor, i.e. $P_{i=1,2} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1}$, then maximum w visit slots are possible, even though there are only two patients

If we increase number of patients, i.e. $P_{i=1,2,\ldots,u} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1}$, then also number of possible visits is w. Thus, there is no impact of the increase in number of patients unless number of time slots is increased.

If we increase number of doctors, i.e. $P_{i=1,2,\ldots,u} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1,2,3,4}$, i.e. u number of patients can visit doctor D_1 in w number of time slots and again the same u number of patients can visit doctor D_2 in w number of time slots and so on, then, total number of possible visits = $w \times 4$.

If v number of doctors are available with w number time slots for only one patient, then the patient can visit any doctor during the w time slots. Hence, it clearly understandable that number of possible visits depends on minimum of patient and doctor nodes multiplied with number of time slots.

An example of *multilayer patient–doctor* database with a single treatment episode layer is shown in Fig. 16. Six cases are depicted from (a) to (f) in Fig. 16. Maximum number of directed edges ($P_i \rightarrow T_k \rightarrow D_j$) in a single layer, or total number of patient visits to the doctors in a single layer can be evident from these six cases, i.e. if $P = u$, $T = w$ and $D = v$ then maximum number of visits are $\min(u, v) \times w$.

Theorem 2. *P_1, P_2...P_u are the u number of distinct patients who visit v number of distinct doctor nodes D_1, D_2.... D_v through unique w number allocated T_1, T_2.. T_w time slots in total M number of treatment episode layers. Then the maximum number of multiple visits in M treatment episode layers is* $\min(u,v)\times(w)$ *or* $\min(\bigcup_{i=u1}^{ua} P_i^M, \bigcup_{j=v1}^{vb} D_j^M) \times \bigcup_{k=w1}^{wc} T_k^M$. *Here,* $u = P_{u1}^1 \cup P_{u2}^2 \cup P_{u3}^3.. \cup P_{ua}^M$, $v = D_{v1}^1 \cup D_{v2}^2 \cup D_{v3}^3.. \cup D_{vb}^M$ *and* $w = T_{w1}^1 \cup T_{w2}^2 \cup T_{w3}^3.. \cup T_{wc}^M$ *are represented as the number of distinct patients, doctors and time slots in M treatment episode layers. Here $u1$, $u2$, $u3$...ua, $v1$, $v2$, $v3$..vb, $w1$, $w2$..wc are all integers.*

The theorem can be proved as follows. From Theorem 1, it is clearly understandable that if the number of patients, doctors and time nodes are u, v and w respectively, then visits are represented by $P_{i=1,2,\ldots,u} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1,2,..v}$ and total number of possible visits is $\min(P_{i=1,2,\ldots,u}, D_{j=1,2,..v}) \times (T_{k=1,2,..w})$ or $\min(u, v) \times w$ in a single treatment episode layer.

Now, in multiple treatment episode layers, if the number of distinct patients, doctors and time nodes are u, v and w respectively, then total number of possible visits are represented by $P_{i=1,2,\ldots,u} \rightarrow T_{k=1,2,..w} \rightarrow D_{j=1,2,..v}$. If any patient use any time slot in any treatment episode layer, then the same patient cannot use the same time slot to visit any doctor in other treatment episode layers. Similarly, the same time slot cannot be allocated to the same doctor in other layers to treat any other patient. But other patients can use the same time slots to visit other doctors in other treatment episode layers. This is shown with suitable examples with two and three layer *patient–doctor* databases in Fig. 17(a) to (f) and to show the uniqueness of each visit by individual patients, directed edges are represented using different colors or line style.

Thus, if the number of distinct patients, doctors and their allocated time slots are u, v and w in multiple treatment episode layers, then the total number of possible visits = $\min(u, v) \times w$ or $\min(\bigcup_{i=u1}^{ua} P_i^M, \bigcup_{j=v1}^{vb} D_j^M) \times \bigcup_{k=w1}^{wc} T_k^M$.

5.3. Algorithm for computing trust factor

Trust factor is the measure of satisfaction for patients regarding a doctor. Hence, the algorithm for computing *trust factor* associates a *trust* value or *trust factor* with every doctor node. The proposed algorithm for computing *trust factor* is shown in Algorithm 1.

The algorithm is explained in this section with a small *multilayer patient–doctor* graph database (shown in Fig. 5), which is part of a large dataset. The large dataset is attached as *supplementary material*. In this example, only three treatment episode layers are shown. Here, number of distinct patients, time slots and doctor nodes are $P = 4$, $T = 5$ and $D = 4$, where $P = 3$, $T = 4$ and $D = 3$ in the 1st layer, $P = 3$, $T = 4$ and $D = 4$ in the 2nd layer, and $P = 3$, $T = 4$ and $D = 3$ in the 3rd layer.

A doctor D_x is input to the algorithm for whom the trust factor will be computed. Line 4 computes $Total_m$, i.e. the number of possible visits in all the treatment episode layers(m) in which doctor D_x is present.

When all four doctors are considered, it can be seen that D_1 node is present in all three layers; similarly, D_2 in $m = 1$, 2 and 3; D_3 in $m = 1$, 2 and 3; and D_4 is present in $m = 2$ layer only.

Next, line 4 computes the maximum possible visits for each doctor in all treatment episode layers where patients visit that particular doctor. Thus, considering all three treatment episode layers, the values of $Total_m$ are shown in the second column of Table 9.

Line 6 of the proposed algorithm finds number of actual visits to the particular doctor D_x previously occurred in all layers. In this example, earlier visits to doctor D_1 are 4, doctor D_2 are 4, doctor D_3 are 5, and doctor D_4 are 2 as shown in the third column of Table 9.

Maximum possible visits at individual treatment episode layers are calculated as follows. At the first layer, maximum number of possible visits is $Visit_{\max} = \min(3, 3) \times 4 = 12$; at $m = 2$ layer, the value is $Visit_{\max} = \min(3, 4) \times 4 = 12$; and at the third layer, it is $Visit_{\max} = \min(3, 3) \times 4 = 12$.

For each patient who previously visited a doctor D_x, but at a later time visits another doctor under the same treatment episode (change in the same layer), the value SFT_m is increased. This value is counted for each of the m layers in which the particular doctor D_x exists and the number of shifts is recorded. Thus, as in the example given in Fig. 5, the SFT_m values for the doctors are shown in the third column of Table 9.

Finally, the *trust factor* is computed according to the proposed algorithm as shown in line numbers 9–12. The *trust factor* values of all doctors are calculated for the given example (Fig. 5) and shown in the last column of Table 9.

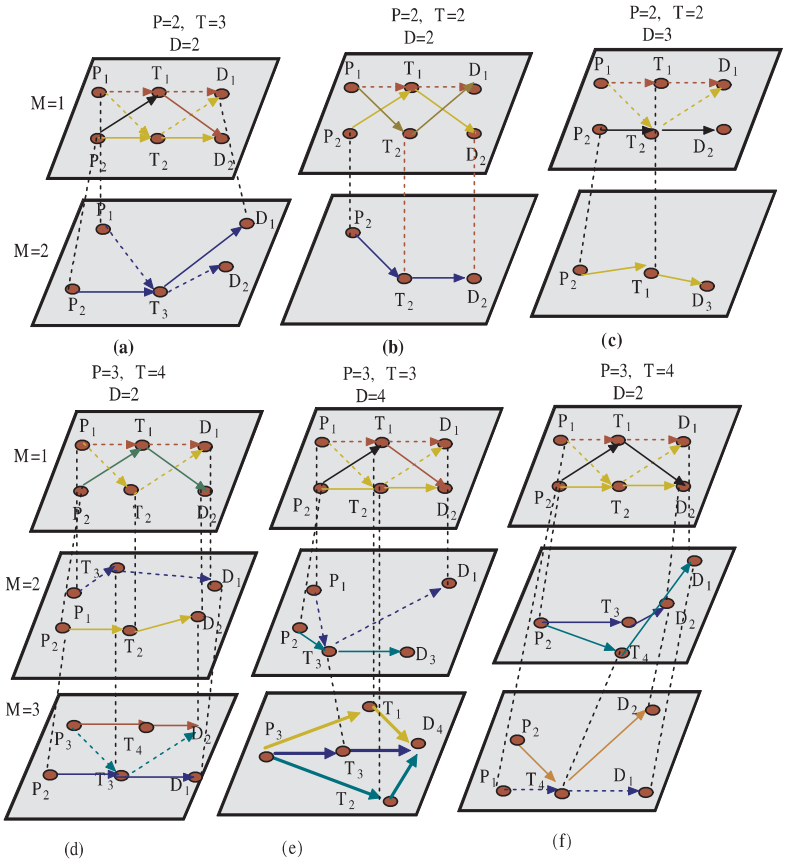

Fig. 17. Example of multilayer *patient–doctor* data model (to show the uniqueness of each visit, the directed edges are represented using different colors or line style). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1 Measure of *Trust*

procedure $Trust(T_f, D_x)$
  Input node D_x. ▷ $x \in v$
  For all m, where $m \in M$ if $D_x^m \in M$ do. ▷ m is the number of treatment episode layers where D_x node present.
  Find $Total_m = \min(\bigcup_{i=u1}^{ua} P_i^m, \bigcup_{j=v1}^{vb} D_j^m) \times \bigcup_{k=w1}^{wc} T_k^m$. ▷ $(\bigcup_{i=u1}^{ua} P_i^m, \bigcup_{j=v1}^{vb} D_j^m$ and $\bigcup_{k=w1}^{wc} T_k^m)$ are the distinct patient, doctor and time slots in m treatment episode layers.
  for ∀m∈ M **do**
    $Find\ ELT_m = \sum_1^M (P_i \to T_k \to D_x)^m$ ▷ *ELT* refer as earlier patient visit to the doctor D_x
    $Visit_{\max} = min(P^m_{i=1,2..u}, D^m_{j=1,2..v}) \times T^m_{k=1,2..w}$ ▷ $Visit_{max} = min(u, v) \times w$
    $SFT_m = (P_i \to T_k \to D_L)^m$ ▷ The *SFT* refer as a shift visit by the patients who visited earlier to D_x doctor.
  if $P_i \to T_t \to D_L \neq True$ **then** ▷ $D_L \cap D_j = 0$ and $P_i \to T_t \to D_x \neq P_i \to T_k \to D_L$
    trust factor $T_f = \frac{ELT_m}{Total_m}$
  else
    Trust factor $T_f = \frac{ELT_m}{Total_m} - \sum_1^M \frac{SFT_m}{Visit_{max}}$

Table 9
The *trust factor* of doctors for given example.

Doctor(D)	*Number of possible visits* ($Total_m$)	*Earlier visits*($ELTm$)	*Shift visits* (SFT_m)	*Trust Factor*
D_1	Present in all three layers. $Total_m = \min(4,4) \times 5 = 20$	$P_1 = 3$ at TE_1, $P_4 = 1$ at TE_3	No shift visit	0.2
D_2	present in all three layers. $Total_m = \min(4,4) \times 5 = 20$	$P_2 = 1$ at TE_1, $P_3 = 1$ at TE_1, $P_1 = 1$ at TE_2, $P_2 = 1$ at TE_2	1 at TE_1 (P_2 visits D_2 at T_2 and then D_3 at T_4), 1 at TE_2 (P_2 visits D_2 at T_1 and then D_4 at T_3)	−0.0167
D_3	Present in all three layers.$Total_m = \min(4,4) \times 5 = 20$	$P_3 = 1$ at TE_1, $P_2 = 1$ at TE_1, $P_3 = 1$ at TE_2, $P_4 = 1$ at TE_3, $P_2 = 1$ at TE_3	1 at TE_1 (P_3 visits D_3 at T_1 and then D_2 at T_3), 1 at TE_3 (P_4 visits D_3 at T_2 and then D_1 at T_4)	0.0833
D_4	Present in TE_2 only. $Total_m = \min(3,4) \times 4 = 12$	$P_2 = 1$ at TE_2, $P_3 = 1$ at TE_2	1 at TE_2 (P_3 visits D_4 at T_4 and then D_3 at T_5)	0.0833

6. Analysis of trust factor

As discussed in the previous section, *trust factor* for the doctors depends on the earlier visits(*ELT*), shift(*SFT*) visits, treatment episode layers and maximum possible visits at all layers. In this section some analysis is done for the *trust factor* to answer to the following questions.

1: How is *trust factor* of a doctor affected by increasing treatment episode layers?

2: How is *trust factor* of a doctor dependent on the earlier visits of patients to doctors?

3: How is *trust factor* affected if maximum visits are increased in an individual layer or in all layers?

4: How is *trust factor* affected if patient changes the doctor by visiting another doctor in the same layer? According to the algorithm, the *trust factor* can be calculated using the following equation.

$Trust\,factor = \frac{ELTm}{Total_m}$ -$\sum_1^M \frac{SFT_m}{Visit_{max}}$.

Suppose X = number of *Earlier* visits for all treatment episode layers(m) and Y_m = number of shift visits(*SFT*) in the mth treatment episode layer. That is, Y_1, Y_2, Y_3.....Y_M are shift(*SFT*) visits in each individual treatment episode layer. If R and r_m are the total maximum possible visits for all M individual layers and maximum possible visits in the mth individual layer respectively, then, the *trust factor* can be written by using the following equation: $T_f = \frac{X}{R} - \sum_1^M \frac{Y_m}{r_m}$

Based on this equation, it can be analyzed that how *trust factor* will vary for the above mentioned cases. For this, we discuss four corollaries and their proofs.

Corollary 1. *In case of a single layer graph, trust factor of a doctor will be positive if multiple visits occur in one treatment episode layer and if earlier(ELT) visits are more than shift (SFT) visits. In other cases, trust factor is negative.*

Proof. The *trust factor* can be calculated as $T_f = \frac{X}{R} - \sum_1^M \frac{Y_m}{r_m}$. Hence, if $M = 1$, it can be written as $T_f = \frac{X}{R} - \frac{Y}{r}$. In single layer, according to *Theorem* 1, $R = r$. Thus, the equation can be written as $T_f = \frac{1}{R}(X - Y)$.

Now it is concluded from this equation that if $X > Y$, then the *trust factor* must always have positive value.

Corollary 2. *The trust factor of any doctor will decrease gradually if the treatment episode layers are increased with fixed number of shift visits.*

Proof. Suppose, $r = r_1 + r_2 + r_3 + ...+r_M$. From Theorems *1* and *2*, it is clear that only $r > R$ can be possible if the value of $M > 1$, but $R = r$ when $M = 1$. Now the equation can be written as $T_f = \frac{X}{R} - (\frac{Y_1}{r_1} + \frac{Y_2}{r_2} + \frac{Y_3}{r_3} + \cdots .. + \frac{Y_M}{r_M})$ or $T_f = \frac{X}{R} - \sum_1^M \frac{Y_m}{r_m}$

According to the Algorithm 1 for *trust factor* calculation, the *trust factor* can be written as $T_{f_m} = \frac{X}{R_M} - \sum_1^M \frac{Y_m}{r_m}$ for M treatment episode layers and $T_{f_{M-1}} = \frac{X}{R_{M-1}} - \sum_1^{M-1} \frac{Y_m}{r_m}$ for $M - 1$ layers.

Hence, the change in *trust factor* with increased treatment episode layer can be written as: $T_f = T_{f_M} - T_{f_{M-1}}$. or $T_f = (\frac{X}{R_M} - \frac{X}{R_{M-1}}) + (\sum_1^{M-1} \frac{Y}{r} - \sum_1^M \frac{Y_m}{r_m})$ for any value of m. or $T_f = (\frac{X}{R_M} - \frac{X}{R_{M-1}}) + [\frac{Y_1}{r_1} + \frac{Y_2}{r_2} + ...\frac{y_{m-1}}{r_{m-1}}] - [\frac{Y_1}{r_1} + \frac{Y_2}{r_2} + \cdots .\frac{Y_{m-1}}{r_{m-1}} + \frac{y_m}{r_m}]$

$T_f = (\frac{X}{R_M} - \frac{X}{R_{M-1}}) - \frac{Y_m}{r_m}$.

The first part of this equation will always be negative ($\frac{X}{R_M} - \frac{X}{R_{M-1}} < 0$) as ($R_M > R_{M-1}$) and $\frac{y_m}{r_m}$ is always positive. Thus the *trust factor* T_f or $T_{f_M} - T_{f_{M-1}}$ is always negative for any value of M. Hence, it can be said that with the increasing number of treatment episode layers, the *trust factor* value always decreases when earlier (*ELT*) and shift (*SFT*) visits are fixed.

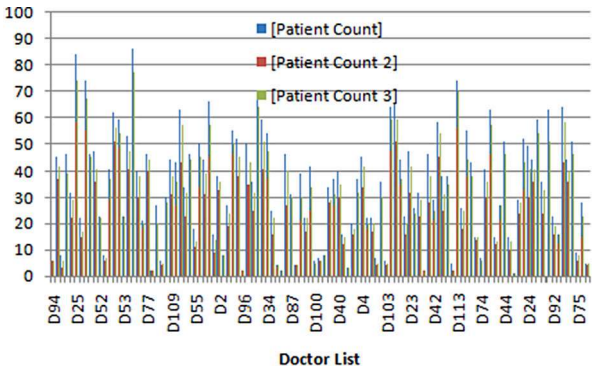

Fig. 18. Number of patients visited to the doctors.

Corollary 3. *In case of a multiple layer graph, the trust factor will decrease or will become negative if changed visits happen to be more than earlier visits considering all layers.*

Proof. The equation for the measure of *trust factor* is formulated as $T_f = \frac{1}{R}(X - \frac{R}{r}[Y_1 + Y_2 + Y_3 + \cdots .Y_M])$. If R and r are fixed, then with the increase in *shift* visits, $(Y_1 + Y_2 + Y_3 + \cdots .Y_M)$ will become more than the *earlier* visits or X, The *trust factor* must be decreasing or negative value.

Corollary 4. *The trust factor of a doctor always increases in single treatment episode layer with the increase of ELT visit of patients. But in most of the cases, trust factor will decrease with the increase of patients ELTvisit at multiple treatment episode layers.*

Proof. If $\frac{R}{r}$ is fixed, then *trust factor* will be positive for increase of X value. But in reality, it is not possible in multiple episode layers, because if the earlier visit increases for multiple treatment episode layers, then, the value of R will increase($R > X$) compared to the value X which will affect the *trust factor*. But if the r value is increased then it will not be affected too much (because $r = r_1 + r_2 + r_3 + ... + r_M$ and $r > R$). Hence, it is concluded that for the increase of patient *ELT* visits in multilayer treatment episode, *trust factor* will decrease.

7. Results on a real dataset

In this section, we consider a real dataset. This *patient–doctor multilayer* dataset has more than *5300* records which have been collected from different health centers and hospitals during a specific time period. The dataset is created after cleaning some unnecessary fields which are useless in our work and the patient names and contact numbers along with doctor names are changed using synthesized values for the sake of privacy and security. Each treatment episode layer is associated with a *suspected disease*. Suspected disease can be evaluated only on the basis of severity of *symptoms* or *signs of the symptoms*. With a *suspected disease*, patient can have multiple visits to the doctor for treatment of the disease. The *patient–doctor multilayer* dataset is attached as *supplementary material 1*.

Calculation and analysis of *trust factor* have been done using *qlikview* [69] data analytic tool which is used heavily as healthcare outcome solutions in different professional industries. The results are attached in a file as *supplementary material*. The list of doctors along with patient counts is shown in Fig. 18. Here, in addition to total number of patients to a doctor (*patient count*) in any treatment episode layer, *patient count 2* represents the number of patients who visited an individual doctor at least twice in any treatment episode layer, and *patient count 3* represents the number of patients who visited an individual doctor at least three times in any treatment episode layer. All these results are taken from *supplementary material 5*.

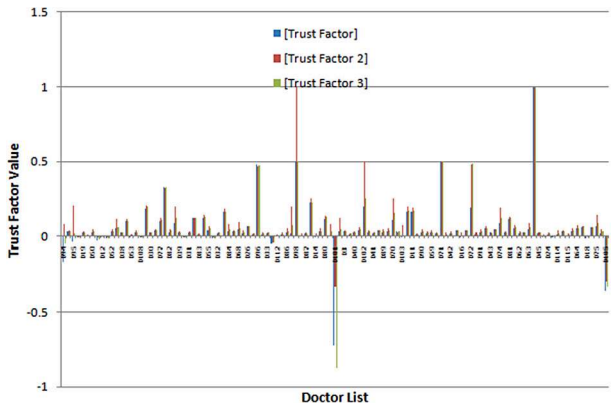

Fig. 19. Trust factor values for doctors.

Table 10
Trust Factor of doctors in one treatment layer (Supplementary material).

Doctors	Treatment Layer	Trust Values
D49	Hepatitis	0.01201
D53	Dengue	0.03635
D101	Dengue	0.023809
.	.	.
.	.	.
D53	Hepatitis	0.025892
D101	Dental Care	−0.3571428
D105	measles	−0.35714

The algorithm for computing *trust factors* is implemented using Python *3.7.3* language in a *64* bit windows machine with *Intel(R)core(TM)i3-5010U CPU@2.10 GHZ* processor and *4 GB RAM*. The list of doctors is shown along with their trust values in Fig. 19. In this result, only those doctors are chosen who have two or three multiple visits by the same patients in multiple treatment episode layers. The TF_2 value represents the *trust factor* of doctors who are visited at least two times by the same patient in any single treatment episode layer. The TF_3 value represents the *trust factor* of doctors who are visited at least three times by the same patient in any single treatment episode layer and the TF value represented the *trust factor* of doctors who are visited by any patient as a whole in any treatment episode layer. The results of the analysis is taken from *supplementary material 6*.

The corollaries stated in Section 6 are also supported with the results from the dataset. In order to demonstrate Corollary 1, Table 10 shows *trust factor* values of a few doctors (for the space constraint only few rows are given) based on the visits in a single treatment episode layer. It is shown that the values are always positive if multiple visits are made in only one treatment episode layer and if the number of earlier visits is more than the number of shift visits. The result of first three examples are taken from *supplementary material 2* and *supplementary material 3*. Also it is shown that under the same condition, if number of earlier visits is less than the number of shift visits, the trust value will be negative which is represented by the last two rows. Thus, all the results of the analysis (*supplementary material 2 and supplementary material 3*) are supportive of the Corollary 1. The proof of Corollary 2 is shown in Fig. 20. The results of the analysis are obtained from *supplementary material 2*. Here, it is shown how *trust factor* of a doctor decreases, if the number of treatment episode layers is increased. Corollary 3 is proved by analyzing the data given in *supplementary material 4*. Results are taken from a small portion of data from *supplementary material 4* and are shown as a graph in Fig. 21. It is evident that with the increase in shift visits (more than earlier visit), the *trust factor* of a doctor decreases as discussed in Corollary 3. The *supplementary material 8* and *supplementary material 9* support Corollary 4. *Supplementary material 8* contains the

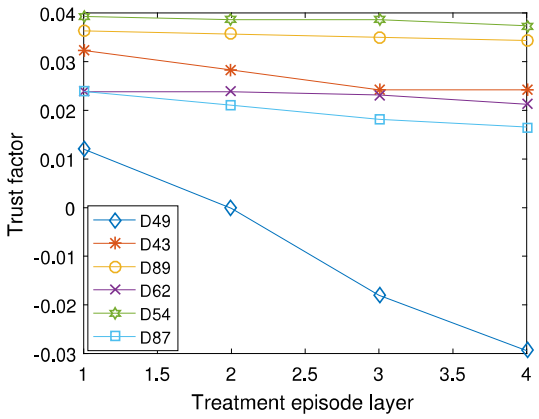

Fig. 20. The *Trust Value* with increasing treatment layer.

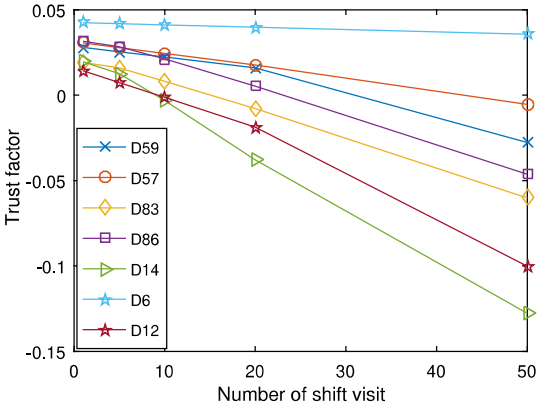

Fig. 21. The *Trust Value* with increasing shift visits.

data about the earlier visits of the patients having one or more than one visits. On the other hand, in *supplementary material 9*, suspected disease for each patient and the doctors' specializations are included making this data a multilayer database. Comparing these two datasets, it is shown in Table 11 that with the increase in earlier visits of patients in multiple layers, the *trust factor* decreases. Here also only a few rows are shown. As per the data obtained from *supplementary material 9*, there is one exception. In Table 11, the trust factor of doctor D6 is negative because in a single treatment episode layer (18 visits), the number of earlier (ELT) visits is less than the shift visits. But after increasing treatment episode layers (multiple layers), when 37 visits occur, trust factor increased due to increase in the number of earlier visits which has now become more than the number of shift visits in other layers.

8. Limitation

Trust of a patient toward a doctor is an expectation of future behavior, and satisfaction of patient toward a doctor is the past experience. Trust is an outcome based on satisfaction. Many research papers [70, 71] consider several factors which affect the patients' trust to doctors. However, in this article trust factor is evaluated only on the basis of number of visits to individual doctors and patients' decisions of changing doctors. But there are other valuable inputs which can affect the trust in patient–doctor relationships. These include doctors' behavior which can increase or decrease trust for the competency. Additionally, communication, care, honesty, and partnering of a doctor, simple and elegant appearances and cultural, ethnic, linguistic and social factors of doctors play significant role in building trust in healthcare. Moreover, patient continuity of care, the perception of risk, socio-economic status

Table 11
Trust Factor of doctors in multiple treatment episode layers for Corollary 4 (Supplementary material 8, 9).

Doctors	Single earlier visits	Trust Values	Increase earlier visits in increasing multiple layers	Trust Values
D53	13	0.076923077	19	0.033354471
D100	3	0.333333333	6	0.166666667
D26	37	0.083333333	59	−0.195454545
D133	43	0.0625	75	0.060876623
D6	18	−0.019621749	37	−0.125539887

of the patient and affordability of healthcare are some other factors which may affect the decisions related to visits as well as patient trust. These factors are not included in our doctor recommendation system to model the trust factor. An additional limitation in this article is that suitable method could not be devised to compare the trust model with other existing models.

9. Conclusion

The paper addresses two important issues of a doctor recommendation system — how to model the patient–doctor relationships for efficient access to data, and how to model trust factor on the basis of the underlying database. The patient–doctor relationship has been modeled as a multilayer graph in this paper. The *multilayer patient–doctor* graph data model is validated using suitable, efficient *NoSQL Neo4j* graph database. A relational model is also incorporated and compared with the multilayer graph model proposed by us. It has been shown that due to the complexity and irregularity in the patient–doctor relationships, the graph data model works more efficiently than the relational model in terms of data access. This is because, in this particular scenario, relationships between individual entities are important, not the relationships among similar types of entities as modeled in relational data model when mapped onto a table. Thus, in a relational data model, when the records are traversed to discover relationship between a patient and a doctor (such as, representing a treatment episode), *join* operations are required leading to increase in the access time. A graph model, on the other hand reduces this access time by mapping the relationship as edges between nodes.

The second contribution of this paper is the concept of trust factor used for doctor recommendation. The trust factor is modeled based on the current state of the database. Thus, no external information is required to measure the trust. Moreover, the trust factor is updated along with any changes in the database. The multilayer graph model and the trust factor have been implemented using real-life data.

CRediT authorship contribution statement

Safikureshi Mondal: Conceptualization, Methodology, Experiment, Writing. **Anwesha Basu:** Data collection, Experiment. **Nandini Mukherjee:** Supervision, Writing -review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jbi.2020.103549.

References

[1] Poly Sil Sen, Shabnam Banerjee, Nandini Mukherjee, Ontology-driven approach to health data management for remote healthcare delivery, in: Proceedings of the 7th ACM Workshop on ACM Mobile Health 2017, MobileHealth'17, ACM, New York, NY, USA, 2017, pp. 2:1–2:6.

[2] S. Mondal, N. Mukherjee, Mobile-assisted remote healthcare delivery, in: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing, PDGC, 2016, pp. 630–635.

[3] S. Mondal, N. Mukherjee, A BFS based pruning algorithm for disease-symptom knowledge graph database, in: 2018 the 3rd International Conference on Information and Communication Technology for Intelligent Systems, ICTIS 2018.

[4] Jaroslav Pokorný, Conceptual and database modelling of graph databases, in: Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS '16, ACM, New York, NY, USA, 2016, pp. 370–377.

[5] Renzo Angles, Claudio Gutierrez, Survey of graph database models, ACM Comput. Surv. 40 (1) (2008) 1:1–1:39.

[6] Yong Cai, Yunlong Wang, Dong Dai, Rare disease physician targeting: A factor graph approach, 2017, arXiv e-prints, arXiv:1701.05644.

[7] Alessio Cardillo, Massimiliano Zanin, Jesús Gómez-Gardeñes, Miguel Romance, Alejandro J. García del Amo, Stefano Boccaletti, Modeling the multi-layer nature of the European Air Transport Network: Resilience and passengers re-scheduling under random failures, Eur. Phys. J. Spec. Top. 215 (2013) 23–33.

[8] Regino Criado, Miguel Romance, M. Vela-Pérez, Hyperstructures, a new approach to complex systems, Int. J. Bifurcation Chaos 20 (3) (2010) 877–883.

[9] Regino Criado, Benito Hernández-Bermejo, Miguel Romance, Efficiency, vulnerability and cost: an overview with applications to subway networks worldwide, Int. J. Bifurcation Chaos 17 (7) (2007) 2289–2301.

[10] Robert Ferrer, Simon J. Hambidge, Rose C. Maly, The essential role of generalists in health care systems, Ann. Int. Med. 142 (2005) 691–699.

[11] Viroj Wiwanitkit, Family medicine in Thailand: System, training, and obstacles, Med. J. Dr. D.Y. Patil Univ. 9 (1) (2016) 4–6.

[12] Kevin Fiscella, Sean Meldrum, Peter Franks, Cleveland G. Shields, Paul Duberstein, Susan H. McDaniel, Ronald M. Epstein, Patient trust: Is it related to patient-centered behavior of primary care physicians? Med. Care 42 (11) (2004) 1049–1055.

[13] Johanna Birkhäuer, Jens Gaab, Joe Kossowsky, Sebastian Hasler, Peter Krummenacher, Christoph Werner, Heike Gerger, Trust in the health care professional and health outcome: A meta-analysis, PLoS One (2017).

[14] https://Neo4j.Com/.

[15] Thomas S. Szasz, March H. Hollender, A contribution to the philosophy of medicine: The basic models of the doctor-patient relationship, AMA Arch. Intern. Med. 97 (5) (1956) 585–592.

[16] R. Kaba, P. Sooriakumaran, The evolution of the doctor-patient relationship, Int. J. Surg. 5 (1) (2007) 57–65.

[17] Liviu Oprea, An analytic review of the doctor-patient relationship, Rom. J. Bioeth. 7 (2) (2009).

[18] Maulik Parekh, B. Saleena, Designing a cloud based framework for healthcare system and applying clustering techniques for region wise diagnosis, Procedia Comput. Sci. 50 (2015) 537–542, Big Data, Cloud and Computing Challenges.

[19] Maulik Parekh, B. Saleena, Designing a cloud based framework for healthcare system and applying clustering techniques for region wise diagnosis, Procedia Comput. Sci. 50 (2015) 537–542, Big Data, Cloud and Computing Challenges.

[20] Himmelstein D.S., A. Lizee, C. Hessler, L. Brueggeman, S.L. Chen, D. Hadley, Systematic integration of biomedical knowledge prioritizes drugs for repurposing, eLife, 6.

[21] Gligorijevic Vladimir, Prulj Nataa, Methods for biological data integration: perspectives and challenges, J. R. Soc. Interface (2015).

[22] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2013) 167–256.

[23] Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco Del Pozo, Stefano Boccaletti, Emergence of network features from multiplexity, Sci. Rep. 3 (2013) 1344, arXiv:1212.2153.

[24] Stanley Wasserman, Katherine Faust, Social Network Analysis: Methods and Applications, Structural Analysis in the Social Sciences, Cambridge University Press, 1994.

[25] S. Havlin, N.A.M. Araujo, S.V. Buldyrev, C.S. Dias, R. Parshani, G. Paul, H.E. Stanley, Catastrophic cascade of failures in interdependent networks, 2010, arXiv e-prints, arXiv:1012.0206.

[26] Dror Kenett, Jianxi Gao, Xuqing Huang, Shuai Shao, Irena Vodenska, Sergey Buldyrev, Gerald Paul, H Stanley, Shlomo Havlin, Network of interdependent networks: Overview of theory and applications, Underst. Complex Syst. (2014) 3–36.

[27] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, Dino Pedreschi, Foundations of multidimensional network analysis, in: Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining, ASONAM '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 485–489.

[28] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A. Porter, Sergio Gómez, Alex Arenas, Mathematical formulation of multilayer networks, Phys. Rev. X 3 (2013) 041022.

[29] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter, Multilayer networks, J. Complex Netw. 2 (3) (2014) 203–271.

[30] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter, Multilayer networks, J. Complex Netw. 2 (3) (2014) 203–271.

[31] Jonathan F. Donges, Hanna C.H. Schultz, Norbert Marwan, Yong Zou, Jürgen Kurths, Investigating the topology of interacting networks - Theory and application to coupled climate subnetworks, 2011, CoRR, abs/1102.3067.

[32] Himmelstein D.S., A. Lizee, C. Hessler, L. Brueggeman, S.L. Chen, D. Hadley, Systematic integration of biomedical knowledge prioritizes drugs for repurposing, eLife, 6.

[33] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, Shlomo Havlin, Catastrophic cascade of failures in interdependent networks, Nature 464 (7291) (2010) 1025–1028.

[34] Anna Saumell-Mendiola, M. Ángeles Serrano, Marián Boguñá, Epidemic spreading on interconnected networks, Phys. Rev. E 86 (2012) 026106.

[35] Alexei Vazquez, Spreading dynamics on heterogeneous populations: Multitype network approach, Phys. Rev. E 74 (2006) 066114.

[36] Peng Wang, Garry Robins, Philippa Pattison, Emmanuel Lazega, Exponential random graph models for multilevel networks, Social Networks 35 (1) (2013) 96–115.

[37] Gourab Ghoshal, Vinko Zlatić, Guido Caldarelli, M.E.J. Newman, Random hypergraphs and their applications, Phys. Rev. E 79 (2009) 066118.

[38] Petter Holme, Jari Saramäki, Temporal networks, Phys. Rep. 519 (3) (2012) 97–125, Temporal Networks.

[39] E. K. Lua, R. Chen, Z. Cai, Social trust and reputation in online social networks, in: 2011 IEEE 17th International Conference on Parallel and Distributed Systems, 2011, pp. 811–816.

[40] S. C. Sousa, D. Lamas, Emerging trust patterns in online communities, in: 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, 2011, pp. 313–316.

[41] Anupama Vohra, Neha Bhardwaj, From active participation to engagement in online communities: Analysing the mediating role of trust and commitment, J. Mark. Commun. 25 (1) (2019) 89–114.

[42] Wanita Sherchan, Surya Nepal, Cecile Paris, A survey of trust in social networks, ACM Comput. Surv. 45 (4) (2013) 47:1–47:33.

[43] Yefeng Ruan, Arjan Durresi, A survey of trust management systems for online social communities – Trust modeling, trust inference and attacks, Knowl.-Based Syst. 106 (2016) 150–163.

[44] Nicola Brennan, Oonagh Corrigan, Rebecca Barnes, Mike Calnan, Paul Dieppe, Vikki Entwistle, Trust in the health-care providerpatient relationship: a systematic mapping review of the evidence base, Int. J. Qual. Health Care 25 (6) (2013) 682–688.

[45] David H. Thom, Mark A. Hall, L. Gregory Pawlson, Measuring patients' trust in physicians when assessing quality of care, Health Aff. 23 (4) (2004) 124–132, PMID: 15318572.

[46] Nicola Brennan, Oonagh Corrigan, Rebecca Barnes, Mike Calnan, Paul Dieppe, Vikki Entwistle, Trust in the health-care providerpatient relationship: a systematic mapping review of the evidence base, Int. J. Qual. Health Care 25 (6) (2013) 682–688.

[47] Anuraag A Vazirani, Odhran O'Donoghue, David Brindley, Edward Meinert, Implementing blockchains for efficient health care: Systematic review, J. Med. Internet Res. 21 (2) (2019) e12439.

[48] P. Chomphoosang, A. Durresi, M. Durresi, L. Barolli, Trust management of social networks in health care, in: 2012 15th International Conference on Network-Based Information Systems, 2012, pp. 392–396.

[49] J. Matysiewicz, S. Smyczek, Consumer trust - challenge for E-healthcare, in: 2009 Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology, 2009, pp. 333–338.

[50] P. Cofta, Convergence and trust in ecommerce, BT Technol. J. 24 (2) (2006) 214–218.

[51] Jean Éric Pelet, Panagiota Papadopoulou, The effect of E-commerce websites' colors on customer trust, Int. J. E-Bus. Res. 7 (3) (2011) 1–18.

[52] Davide Di Fatta, Dean Patton, Giampaolo Viglia, The determinants of conversion rates in SME e-commerce websites, J. Retail. Consum. Serv. 41 (C) (2018) 161–168.

[53] Heli Hallikainen, Tommi Laukkanen, National culture and consumer trust in e-commerce, Int. J. Inf. Manag. 38 (1) (2018) 97–106.

[54] Matthew Richardson, Rakesh Agrawal, Pedro Domingos, Trust management for the semantic web, in: Proceedings of the Second International Conference on Semantic Web Conference, LNCS-ISWC'03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 351–368.

[55] Qi Gao, Towards trust in web content using semantic web technologies, in: Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, Tania Tudorache (Eds.), The Semantic Web: Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 457–461.

[56] D. Huang, V. Arasan, On measuring email-based social network trust, in: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, 2010, pp. 1–5.

[57] E. Sezgin, S. Özkan, A systematic literature review on Health Recommender Systems, in: 2013 E-Health and Bioengineering Conference, EHB, 2013, pp. 1–4.

[58] Li Guo, Bo Jin, Cuili Yao, Haoyu Yang, Degen Huang, Fei Wang, Which doctor to trust: A recommender system for identifying the right doctors, J. Med. Internet Res. (2016).

[59] Felix Gräßer, Stefanie Beckert, Denise Küster, Susanne Abraham, Hagen Malberg, Jochen Schmitt, Sebastian Zaunseder, Neighborhood-based collaborative filtering for therapy decision support, in: HealthRecSys@RecSys, 2017.

[60] Hanna Schäfer, Santiago Hors-Fraile, Raghav Pavan Karumur, André Calero Valdez, Alan Said, Helma Torkamaan, Tom Ulmer, Christoph Trattner, Towards health (Aware) recommender systems, in: Proceedings of the 2017 International Conference on Digital Health, in: DH 17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 157–161.

[61] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[62] Safikureshi Mondal Goutam Kundu, Building a graph database for storing heterogeneous healthcare data, in: Information and Communication Technology for Intelligent Systems (ICTIS 2020), Springer, 2020, in press.

[63] Vanesa Espín, María V. Hurtado, Manuel Noguera, Nutrition for Elder Care: a nutritional semantic recommender system for the elderly, Expert Syst. 33 (2) (2016) 201–210.

[64] André Calero Valdez, Martina Ziefle, Katrien Verbert, Alexander Felfernig, Andreas Holzinger, Recommender systems for health informatics: State-of-the-art and future perspectives, in: Machine Learning for Health Informatics, 2016.

[65] Victor Winberg, Jan Zubac, A comparison of relational and graph databases for CRM systems, Student Paper, LU-CS-EX 2019-09, 2019.

[66] MSc Dr. Steven D. Pearson, Lisa H. Raeke Ma, Patients trust in physicians: Many theories, few measures, and little data, J. Gen. Intern. Med. 15 (2000) 509–513.

[67] Suzanne McMurphy, Trust , distrust , and trustworthiness in argumentation : Virtues and fallacies, 2015.

[68] Jeffrey Xu Yu, Jiefeng Cheng, Graph reachability queries: A survey, in: Managing and Mining Graph Data, 2010.

[69] https://www.Qlik.Com/Us/Products/Qlikview.

[70] Fan V.S., M Burman, M.B. McDonell, S.D. Fihn, Continuity of care and other determinants of patient satisfaction with primary care, J. Gen. Intern. Med. 20(3) (11) (2005) 226–233.

[71] M.S. Jennifer Tabler, Debra L. Scammon, Jaewhan Kimand Timothy Farrell, Patient care experiences and perceptions of the patient-provider relationship: A mixed method study, Patient Exper. J. 1(1) (13) (2014) 226–233.