Data & Knowledge Engineering 148 (2023) 102233

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Effective healthcare service recommendation with network representation learning: A recursive neural network approach

Mouhamed Gaith Ayadi [a], Haithem Mezni [b], Rana Alnashwan [c,*], Hela Elmannai [c]

[a] Jendouba University, Tunisia
[b] Applied College, Taibah University, Madinah, Saudi Arabia
[c] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

A R T I C L E I N F O

Keywords:
Recommender system
Healthcare services
Smart medical object
Representation learning
Recursive neural networks
Cuboid space

A B S T R A C T

Recently, recommender systems have been combined with healthcare systems to recommend needed healthcare items for both patients and medical staff. By monitoring the patients' states, healthcare services and their consumed smart medical objects can be recommended to a medical team according to the patient's critical situation and requirements. However, a common drawback of the few existing solutions lies in the limited modeling of the healthcare information network. In addition, current solutions do not consider the typed nature of healthcare items. Moreover, existing healthcare recommender systems lack flexibility, and none of them offers re-configurable healthcare workflows to medical staff. In this paper, we take advantage of collaborative filtering and representation learning principles, by proposing a method for the recommendation of healthcare services. These latter follow a predefined execution pattern, i.e. treatment/medication workflow, that is determined by our framework depending on the patient's state. To achieve this goal, we model the healthcare information network as a *knowledge graph*. This latter, based on an *incremental learning* method, is then transformed into a cuboid space to facilitate its processing. That is by learning latent representations of its content (e.g., smart objects, healthcare services, patients symptoms, etc.). Finally, a *collaborative recommendation* method is defined to select the high-quality healthcare services that will be composed and executed according to a determined workflow model. Experimental results have proven the efficiency of our solution in terms of recommended services' quality.

1. Introduction

Recent years have witnessed a great interest in smart healthcare systems, which have taken advantage of the synergy between powerful paradigms, mainly service computing [1], sensor clouds [2], and Internet of Things (IoT) [3]. These pillar paradigms have endowed healthcare systems with smart monitoring, analysis, and recommendation capacities, making them as service-enabled systems [4], thus changing the way healthcare services are offered. This growth in smart healthcare systems has led to the emergence of "Internet of Healthcare Things" (IoHT) [5], as a large and heterogeneous network of smart medical devices and services (e.g., wearable biosensors, smart thermometers, rehabilitation devices). Such facilities have encouraged healthcare professionals to monitor patients remotely and deliver outpatient and long-term care as remote healthcare services. That also resulted in the

* Corresponding author.
E-mail addresses: mouhamed.gaith.ayadi@gmail.com (M.G. Ayadi), haithem.mezni@gmail.com (H. Mezni), roalnashwan@pnu.edu.sa (R. Alnashwan), hselmannai@pnu.edu.sa (H. Elmannai).

https://doi.org/10.1016/j.datak.2023.102233
Received 7 April 2023; Received in revised form 12 September 2023; Accepted 25 September 2023
Available online 29 September 2023

Basic symbols and notations

$\mathcal{G}$	Healthcare information network
$\mathcal{S}, \mathcal{H}, \mathcal{O}$	Sets of symptoms, healthcare services, and medical IoT objects
$\mathcal{W}$	Set of abstract workflows
$\mathcal{C}$	Cuboid space, i.e. representation of the healthcare network $\mathcal{G}$
x_i^r, x_i^c, x_i^p	$3-C$ (three-directional component) representation of a node v in $\mathcal{C}$
$g(v, i, j, k)$	Location of node v in the three-directional component
NLL	Negative log-likelihood
$\mathcal{T}$	Nodes sequence length
$P_r, P_c, P_p \vert a$	Row, column and page probabilities with a given node type a
$\mathcal{P}$	A meta-path in the healthcare information network
X^r, X^c, X^p	Embedding matrices for row, column, and page dimensions
$h_i^r, h_{i-1}^c, h_{i-1}^p$	Hidden state vectors produced using the recursive operations
$\mathcal{G}'$	Evolving healthcare network, with $\mathcal{G}' = \mathcal{G}(t+1)$
Δ	An increment of changes in the healthcare network
NLL'	Loss function for incremental learning
S'_d	Set of node changes in a directional component d, with d is a row r, a column c, or a page p

emergence of virtual hospitals[1] as a revolutionary kind of "without walls"-medical wards. To achieve this promising vision, IoT devices such as pulse oximeters and armpit patches were used by patients to measure oxygen saturation levels, heart rates, and body temperature. Then, the collected data is transmitted by patients to the virtual hospital to help the patient's care team decide on the appropriate medical treatment.

Despite this promising vision, most existing healthcare systems (e.g., [6–11]) are designed as traditional recommender systems that, based on historical user data, recommend specific treatment plans, without considering the patient-generated health data. Such approaches did not provide an understanding of healthcare entities (e.g., symptoms, IoT healthcare objects, treatment scenarios). As a result, they failed to correctly represent the patients' related information and their healthcare IoT environment. Bellatreche et al. [12] discussed the central role of data repositories and data models in developing data-driven smart systems. They stated that augmenting a system's input data (e.g., healthcare domain) by resources such as knowledge graphs may be an essential step towards efficient and advanced analytics. Besides the above challenges, existing healthcare recommendation systems adopt a static strategy when delivering medication plans (e.g., [6,8,13,14]), while in real-world scenarios, treatment plans should be reconfigurable according to the evolution of patients' health conditions. Finally, current healthcare systems generate medication plans that are executed in isolation, which may lead to conflicting treatments (e.g., undesirable reactions between medication and other drugs) that affect the patients' health.

Aiming to solve the above issues, we aim at providing a smart care delivery solution that recommends daily configurable healthcare workflows to patients. The proposed solution aims at cutting down unnecessary visits to doctors, hospitals, and clinics. It offers the following key advantages: rapid treatment of patients' states, better management of drugs and symptoms, enhanced patient experience, decreased care costs, faster disease diagnosis, proactive treatment, and cost reduction.

Our approach takes advantage of recent technologies and techniques, such as knowledge graphs [15], network representation learning [16], and deep neural networks [17] to propose a recommender system that guides medical staff in maintaining the patients' health states. Such technologies will be exploited to model and explore the healthcare network's complex structure, from which useful knowledge is inferred and exploited in the recommendation of smart healthcare services and their consumed medical objects. Taking advantage of the service paradigm [1], we abstract each treatment plan as a healthcare service. Examples include dietary plans, flu shots, pap smears, day-to-day fitness, treatment of Hepatitis C, nutrition, cancer treatments, physiological health, etc. In the modeled healthcare knowledge graph, users' (patients) and items' (e.g., treatments) interactions are expressed via typed connections. Then, using a recursive variant of neural networks [17,18], the vectorized forms of users and services are encoded as a *3-Component representation*, and arranged in a *cuboid space* to facilitate the recommendation task. Finally, the vector space is explored to locate, evaluate, and recommend the appropriate healthcare services.

The main contributions are summarized as follows:

- We model the *healthcare information network* as a multi-relational and semantic knowledge graph. This latter's content (i.e. healthcare services, IoT objects, relations, etc.) is periodically updated according to the sensed data.
- We facilitate the processing of the healthcare knowledge graph by incrementally transforming its content into a *cuboid* space, using a network representation learning model.

[1] https://econsultancy.com/internet-of-things-healthcare/.

- We define a *collaborative method* that, based on the patient's current state, recommends the suitable healthcare workflow, which consists of an optimal set of services together with their consumed smart medical objects.

The rest of this paper is organized as follows: Section 2 summarizes the current service recommendation solutions in the context of healthcare systems. Section 3 provides a formal definition of the smart healthcare services' recommendation problem. In Sections 4 and 5, we introduce the IoT-based healthcare knowledge graph as well as its parsing process for the purpose of services' and IoT objects' recommendation. A theoretical complexity study is conducted in Section 6. Experimental validation is provided in Section 7. In the last section, we give the concluding remarks and the future research.

2. Related work

2.1. Healthcare recommender systems

As the demand for remote health care is more encouraged than ever, providing smart solutions for healthcare monitoring and recommendation is becoming a necessity, to allow proactive measures and treatments. Healthcare recommendation systems exploit historical patient-generated healthcare to provide personalized services, such as remote diagnosis, remote medication, and healthcare plans [19].

In this context, several kinds of healthcare items have been focused on [19,20]. These include food (e.g., proper diets, food substitutes) recommendation [9–11,21–24], drug recommendation (e.g., for curing diseases) [8,14,25–27], physical activity recommendation [28], proper diets recommendation [29], medical professionals' recommendation [30,31], healthcare wearables' recommendation [6], doctor recommendation [7,13,32–34], health status prediction [35,36], etc. In these attempts, a panoply of techniques and methods have been adopted by researchers, such as graph theory [34], matrix factorization [6], machine learning techniques [27,35,36], recurrent and convolutional neural networks [23,33], graph convolutional networks [24], hierarchical attention networks [21], intuitionistic fuzzy sets and Bonferroni mean [13], classification [7], clustering [8], semantics and ontology [9–11], fuzzy logic [29], tensor decomposition [8], multi-layer graph data models [32], etc. Big data frameworks, such as Apache Spark, have also been applied to cope with the huge flow of healthcare data [35]. Regardless of the recommended item, the ultimate goal of those works is the recommendation of healthcare treatment solutions or the early disease detection. This is by addressing various constraints such as the trust [32], the risk level [27,35,36], and the health status [35]. It also should be noted that traditional filtering techniques have been instantiated in the context of healthcare items' recommendation, like Collaborative Filtering in [8,21].

The above approaches have been treated as a traditional class of recommender systems, where the relationships between patients/users and items (food, drugs, doctors, etc.) is mostly modeled as a simple user–item matrix view, thus, inheriting the well-known recommender systems' issues (data sparsity and cold start). Indeed, each patient usually consumes a very few number of medical items, new patient/items will not have sufficient historical data to be used in the prediction of patients' needs. Moreover, current healthcare recommender systems lack context-awareness, because they did not consider the changing or the past health status of the target patient. In fact, several contextual data have not been considered, such as spatio-temporal information and patient's medical restrictions (e.g., allergy cases). Finally, most researchers have not been addressed the problem of healthcare profiling (e.g., definition of patient profiles), which makes recommendation guided only by medical items' information.

2.2. Knowledge graph-based approaches

As a powerful technology to represent complex and heterogeneous information networks [15], knowledge graphs have been exploited to enhance the quality of healthcare services and management. Healthcare KGs have become a popular and preferred representation of the semantics and multiple relationships between healthcare entities. These latter can be patients, diseases, symptoms, events, drugs, side effects, food, healthy diet, devices, etc. The graph-like representation can improve the discovery of latent knowledge and clinical decision support systems. Healthcare KG usage includes drugs discovery, repurposing, prediction of adverse drug reactions, diseases and mental disorders, bio medicine, diagnosis and treatment of viral infections, topography and anatomy, human microbial metabolism network, etc. Healthcare KG usually are constructed following one of three methods: Named Entity Disambiguation, Named Entity Recognition, or Named Entity Linking [37]. The most common approaches to construct and augment healthcare KGs include classification, NLP techniques, LSTM, semantic analysis (ontology), cross-referencing, structured databases, rule-based methods, and fuzzy matching.

Recent attempts include the construction of healthcare knowledge graphs in some countries, such as Italy [38] and China [39]. KnowLife [40] is another knowledge graph that groups a wide range of entities, including symptoms, diseases, causes, risk factors, side effects, drugs, etc. Other attempts include Tumor-Biomarker Knowledge Graph (TBKM), Global Network of Biomedical Relationships (GNBR), and Microbe-Disease Knowledge Graph (MDKG). Knowledge graphs have also been exploited in several services including clinical search [41,42] and query understanding [42], healthcare-related question answering [43], chatbot healthcare applications [44], smart healthcare management [45,46].

Despite their great potential in supporting recommender systems [47], the adoption of knowledge graph technology in the healthcare domain is still in its infancy. In fact, knowledge graphs have been applied by only three recommendation approaches, with the goal to identify the suitable doctors [48], and foods [10], as well as enhancing disease prediction and treatment recommendation [49]. For example, Liu et al. [49] proposed a multi-task healthcare system that predicts diseases and recommends treatment programs and medical resources to mobile users. The system's healthcare data are collected from multiple channels (hospitals, communities, users) to construct a knowledge graph. This latter's structural, textual, and visual content is, then, embedded using TransD, Word2Vec, and ResNet backbone network, respectively. To extract relevant features and infer new relationships between the healthcare entities, the authors employed a panoply of methods, including Skip-Thought, GRU model, multilayer perceptron. Apart from the high complexity caused by feeding structurally heterogeneous knowledge in three different embedding models, the proposed systems does not provide details on the disease prediction and the recommendation process.

Following the same logic in [49], Yuan and Deng [48] have proposed a framework for healthcare consultation and doctor recommendation. In their approach, a health knowledge graph is built based on the content (e.g., patients, doctors) of a Chinese online healthcare platform. Then, the interactive (patient–doctor) and individual (doctors' service quality) features, extracted from the knowledge graph, were fed into an interpretable deep neural network (DNN) model with a Layer-wise Relevance Propagation (LRP) technique. By combining DNN and LRP, the authors aim at evaluating the features' contribution to the final recommendations.

Unlike the above works, the authors in [10] have focused on food information by constructing a semantic knowledge graph structure, called FoodKG. Then, ontology and linked data techniques have been employed to augment FoodKG structure and to establish the provenance of food-derived relationships. Finally, the healthier food is recommended, by a applying a SPARQL – or natural language – based question answering process.

Knowledge graphs have been also adopted in the service computing domain to model and recommend services based on the similarity between their context and that of the target user [50]. To achieve this goal, dilated recurrent neural networks have been adopted to process the services' usage history as a set of multi-hop sequences. These latter are inputted to a context-aware embedding model to measure the proximity degrees between entities (users, services, context) and recommend the top-rated services.

Unlike the traditional user–item representation (e.g., matrix view) that feeds less expressive knowledge (e.g., users/items similarities, multi-relational features) into the recommender system [19,20], the knowledge graph-based representation has enriched user–item interactions with auxiliary information, which helped reduce collaborative filtering (CF) problems (cold start and data sparsity) and improved the recommendations accuracy [47]. To cope with collaborative filtering issues, some researchers [51–53] have also exploited representation learning principles. The incorporation of network embedding and representation learning techniques has produced more efficient collaborative filtering models, such as neural CF [54], translation-based CF [55], deep generative models (e.g., collaborative variational autoencoders) [56], neural variational CF [57], and Graph attention-based CF [58]. For example, Deng et al. [52] have proposed a collaborative variational deep learning model (CVDL). The CVDL was designed as a Bayesian probabilistic generative model, which performs two main tasks: latent features' extraction and ratings prediction. The first task is achieved by a variational autoencoder, to learn implicit relationships between users (patients) and items (primary care doctors). Then, collaborative information, latent item variables, and the learned user profiles are unified to predict ratings and to generate the best recommendation in the primary care service. Despite their efficiency (e.g., Bayesian nature and non-linearity of VAE), the above models still suffer from poor latent user/item representations and only exploit rating information, which is always affected by the sparsity of user–item data.

Although knowledge graph-based approaches were able to represent healthcare-related data, they commonly suffer from a high complexity caused by the processing and projection of the healthcare network' content. Another major drawback is the followed static mode. In fact, most researchers state that the healthcare network does not undergo any change.

2.3. Current issues and our contributions

Analyzing the above approaches, we noticed the following challenges:

Healthcare network complexity. Current healthcare systems do not provide a full understanding of healthcare actors and entities, such as patients needs, virtual doctors/hospitals capacities, services features (e.g., usage, allergy cases), treatment plans, IoT objects. Even the graph-like representations of healthcare-related data, like in [10,48,49], are far from being complete and lack the semantics of relationships between graph nodes, i.e., healthcare entities. As a solution to these limitations, we exploit the strengths of knowledge graph technology [15], not only to model and construct a large multi-relational and multi-source healthcare information network, but also to infer additional knowledge (e.g., conflict between medication treatments, similarities between healthcare services and between IoT objects, co-occurrence between symptoms, etc.) regarding the relationships among its content (see Section 4).

Dynamicity of the healthcare network. Most existing healthcare recommender systems target static healthcare information networks, which is not realistic, especially with these latter's ever changing and evolving nature (patients states, medical resources' availability, treatments plans quality, etc.). Moreover, knowledge graph-based approaches [48,49] deal with static knowledge structures, which makes the semantic representation and embeddings of their entities not suitable to patients' current health state, and risk to recommend low-quality or inappropriate treatment plans (drugs, healthcare programs). Unlike the above solutions, we endow our recommendation system with incremental learning capabilities (see Section 4.2), to allow updating the healthcare system's knowledge and provide up-to-date and high-quality treatment recommendations.

3. Problem formulation

In this section, we define the problem (input, output, and constraints) of smart healthcare services' recommendation, based on the following key-terms:

Definition 3.1. Healthcare service, denoted by $h \in \mathcal{H}$, is a medical treatment defined as a tuple $\langle \mathcal{F}^h, \mathcal{Q}^h, \mathcal{O}^h \rangle$, where $\mathcal{F}^h = \{f_1^h, f_2^h, \ldots, f_y^h\}$ denotes the healthcare operations to handle a given patient case (e.g., physical and occupational therapy, laboratory and diagnostic care, pharmaceutical care, etc.), whereas $\mathcal{Q}^h = \{q_1^h, q_2^h, \ldots, q_z^h\}$ is the set of features and nonfunctional properties of the healthcare service h (e.g., cost, availability, reputation, etc.). As for $\mathcal{O}^h \subset \mathcal{O}$, it refers to the medical equipment (e.g., smart IoT object) needed to perform the healthcare service operations. Otherwise, $\mathcal{O}^h = \emptyset$.

Definition 3.2. Smart Object is presented by $o = \langle \mathcal{F}^o, \mathcal{S}^o \rangle$ ($o \in \mathcal{O}$), where $\mathcal{F}^o = \{f\}^+$ is a set of features f including connectivity, sensing, active engagements, safety, etc., whereas $\mathcal{S}^o \subset \mathcal{S}$ is a subset of symptoms that, when detected by a patient's monitoring sensors, trigger a healthcare service, as well as the use of the medical equipment o.

Definition 3.3. Abstract healthcare workflow $w \in \mathcal{W}$, also called treatment plan, is defined as a tuple $\mathcal{T}^w, \langle \mathcal{S}^w \rangle$, each healthcare task $t \in \mathcal{T}^w$ maps to a healthcare service that is responsible for handling a patient's given situation. The symptoms denoted by $\mathcal{S}^w \subset \mathcal{S}$ represent a medical case that requires the medical staff's (i.e., healthcare services) intervention.

Definition 3.4. Healthcare information network (HIN) is a complex graph structure $\mathcal{G} = \langle \mathcal{H} \cup \mathcal{O} \cup \mathcal{S} \rangle$ (see Section 4), that represents the semantics and multiple relationships between the healthcare network entities. The HIN is evolving and undergoes continuous changes, such as the availability of new healthcare services, the updates in patients' states, the removal of broken IoT devices (medical equipment).

The problem of healthcare service recommendation is defined as follows: *Given a set of smart IoT objects with their sensing and data management capacities, a set of healthcare services with their functional and quality (QoS) features, a set of workflow templates denoting the healthcare scenarios for a patient's critical situation, the goal is to find and aggregate the appropriate set of healthcare services together with their IoT objects, w.r.t a selected workflow execution pattern, and while taking into account patients' and services' constraints.*

3.1. Input

The smart healthcare environment consists of a set of healthcare services $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$, a set of medical IoT objects $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$, a set of symptoms $\mathcal{S} = \{s_1, s_2, \ldots, s_p\}$, and a set of abstract workflows $\mathcal{W} = \{w_1, w_2, \ldots, w_q\}$ ($w_i = \{(t_j)^+\}_{j=1}^x$, $1 \le x \le p$), i.e. template denoting the possible treatment plans (e.g., a medical staff) for a given patient situation. Each treatment task $t_g \in w_i (1 \le i \le q)$ is instantiated to a healthcare service, in order to offer a treatment operation to patients. Each healthcare service invocation is defined by the triple $\langle h, o, s \rangle$, to indicate that a healthcare treatment $h \in \mathcal{H}$ is applied with the help of some medical equipment $o \in \mathcal{O}$ when at least a symptom $s \in \mathcal{S}$ is detected.

3.2. Output

A patient's current situation is handled by a set of treatment tasks forming a *healthcare workflow* $\mathcal{W}$. This latter is instantiated to a set of services $\mathcal{W}^p = \langle \mathcal{H}^p \cup \mathcal{O}^p \rangle$, where $\mathcal{H}^p \subset \mathcal{H}$ and $\mathcal{O}^p \subset \mathcal{O}$ corresponds, respectively, to the subset of recommended healthcare services and their associated medical equipment, e.g., IoT objects.

3.3. Constraints

In this work, we focus on three types of constraints.

- *Healthcare cost:* indicates the maximum cost of the healthcare procedure triggered for a patient case ($cost_p$). This includes the cost of requested services and their associated medical equipment (e.g., IoT objects).

$$\begin{aligned} &\forall h \in \mathcal{H}^w, \forall o \in \mathcal{O}^h, \\ &\textstyle\sum_{h \in \mathcal{H}^w} \left(cost(h) + \sum_{o \in \mathcal{O}^h} cost(o) \right) \le cost_p \end{aligned} \tag{1}$$

- *Healthcare time:* indicates the maximum time $\mathcal{T}$ needed to completely treat a medical case.

$$\forall h \in \mathcal{H}^w, \textstyle\sum_{h \in \mathcal{H}^w} time(h) \le \mathcal{T} \tag{2}$$

- *Treatment conflicts:* refers to the conflicts between treatments tasks (e.g., allergy cases or conflicts between drugs) that risk to cause undesirable reactions between medication and other drugs and, consequently, may cause side effects on the patient's health. Let $A = \{a_1, a_2, \ldots, a_c\}$ the set of patient restrictions, such as a allergy cases that may trigger life-threatening reactions.

$$\begin{aligned} &\forall h = \langle \mathcal{F}^h, \mathcal{Q}^h, \mathcal{O}^h \rangle \in \mathcal{H}^w, \\ &A \cap F^h = \emptyset \wedge A \cap O^h = \emptyset \end{aligned} \tag{3}$$

4. Modeling of the healthcare IoT environment

As the number of IoT healthcare devices and medical facilities proliferate, the following key questions arise:

- **RQ1.** What technology offers a rich, semantic, and scalable modeling and processing of virtual hospitals' complex data (e.g., patients, symptoms, drugs, medical services and workflows, healthcare devices, etc.)?
- **RQ2.** How to deal with the dynamic and evolving nature of healthcare environments, and which method offers an efficient exploration of the healthcare information network with a reduced complexity?

To answer the above questions, we first modeled the healthcare information network as a *knowledge graph* [15]. This promising data structure is a recent kind of knowledge bases. Since its announcement by Google in 2012, leading companies such as LinkedIn, Facebook, Yahoo, and Wikimedia have exploited knowledge graphs to improve the quality of their offered services, including search engines, location-based services, question answering, etc. The motivation behind choosing a graph-like structure is that the healthcare information network has a multi-relation topology with typed entities (e.g., services, medical devices, symptoms, etc.) and relationships (e.g., between smart medical sensors and healthcare services).

4.1. Healthcare knowledge graph

The healthcare knowledge graph (see Fig. 1) can be seen as a heterogeneous and multi-relational information network that consists of various entities (see Definition 4.1), including patients, symptoms and their diseases, hospitals, healthcare services and their smart medical objects, etc. For simplicity reasons, we focus in this work on a subset of healthcare entities, mainly symptoms, healthcare services, and medical devices.

Definition 4.1 (*Smart Healthcare Knowledge Graph*)**.** a SHKG is a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{D}^+)$, where $\mathcal{E} = \langle \mathcal{E}_p, \mathcal{E}_s, \mathcal{E}_{hs}, \mathcal{E}_{ho}, \mathcal{E}_f \rangle$ is a set of entities denoting, respectively, the patients ($\mathcal{E}_p$), the symptoms ($\mathcal{E}_s$), the healthcare services ($\mathcal{E}_{hs}$), the smart healthcare objects ($\mathcal{E}_{ho}$), and the entities' features ($\mathcal{E}_f$), $\mathcal{R} \subseteq \mathcal{E} \times \mathcal{E}$ is a set of typed relations between entities, and $\mathcal{D}^+$ is a set of facts, which are labeled with a typed relation $t \in \mathcal{T}$.

Definition 4.2 (*Healthcare Fact*)**.** a fact in the SHKG is a 3-tuple $f = (e_i, r, e_j)$, where $e_i, e_j \in \mathcal{E}$ are the head and tail entities (e.g., patients, services, smart object, or features), and $r \in \mathcal{R}$ is a typed relation between e_i and e_j. The set of facts in $\mathcal{G}$ is denoted by $\mathcal{D}^+ = \{(e_i, r, e_j)\}$.

Definition 4.3 (*Relation*)**.** A relation $r \in \mathcal{R}$ in the SHKG is a typed link between entities e_i and e_j. It is defined as $r : e_i \xrightarrow{r} e_j$, where $e_i, e_j \in \mathcal{E}$ ($\mathcal{E} = \langle \mathcal{E}_p \cup \mathcal{E}_{hs} \cup \mathcal{E}_{ho}, \cup \mathcal{E}_f \rangle$). A relation function $f(r)$ returns one of the following values, depending on the types of nodes in the SHKG (see Fig. 1):

$$f(e_i, r_\tau, e_j) = \begin{cases} MONITOR & if\ e_i \in \mathcal{E}_h o \wedge e_j \in \mathcal{E}_p \\ BELONGTO & if\ e_i \in \mathcal{E}_f \wedge e_j \in \mathcal{E}_{hs} \\ INVOKE & if\ e_i \in \mathcal{E}_p \wedge e_j \in \mathcal{E}_{hs} \\ P-SIMILAR & if\ e_i, e_j \in \mathcal{E}_p \\ HS-SIMILAR & if\ e_i, e_j \in \mathcal{E}_{hs} \\ HO-SIMILAR & if\ e_i, e_j \in \mathcal{E}_{ho} \\ F-SIMILAR & if\ e_i, e_j \in \mathcal{E}_f \\ CONFLICTING & if\ e_i, e_j \in \mathcal{E}_{hs} \end{cases}$$

The healthcare knowledge graph contains two types of IoT devices. For patients, devices and wearables like fitness bands, blood pressure, heart rate monitoring cuffs, asthma tools, and glucometers, are responsible for collecting health information to keep track of patients' health. The collected information is exploited to understand the patients' health conditions and identify eventual symptoms, for the purpose of medical plan recommendation. The second type of IoT devices in the HKG is associated to healthcare services and exploited by medical staff to track patients' adherence to the recommended treatment plans. In this case, the produced data will guide the medical staff in taking immediate attention or the healthcare system in re-configuring the recommended healthcare workflow (medication plan).

To correctly classify and arrange healthcare services into groups of treatment cases and to efficiently guide the recommendation task, the healthcare network entities, as well as their features initially mined from available data, can be abstracted as requirement patterns (e.g., symptoms) and service patterns (e.g., medication plan). This helps exclude the services belonging to irrelevant patterns. As depicted in Fig. 2, the healthcare entities (e.g., services, symptoms, disease) are represented by one or multiple patterns that denote typical health situations and their treatment solutions. The mapping (arrows) between entities is exploited later to identify the candidate healthcare services. For example, the services {*Pediatric Derm., Cosmetic Derm., Light therapy, Surgical Derm., and General Derm.*} belong to the pattern *SP:Dermatology*. Therefore, they tend to be close in the vector embedding space. Likewise, the symptoms {*Hair loss, Eczema, Chickenpox, Skin, Skin cancer and psoriasis*} abstracted as a requirement pattern *RP:Dermatology* are, not only close in the embedding space, but also mapped close to *SP:Dermatology* services, thanks to the matching between the two patterns.

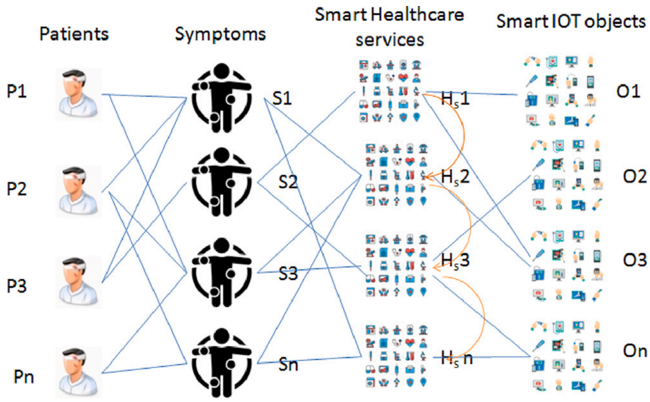

Fig. 1. IoT-based healthcare information network.

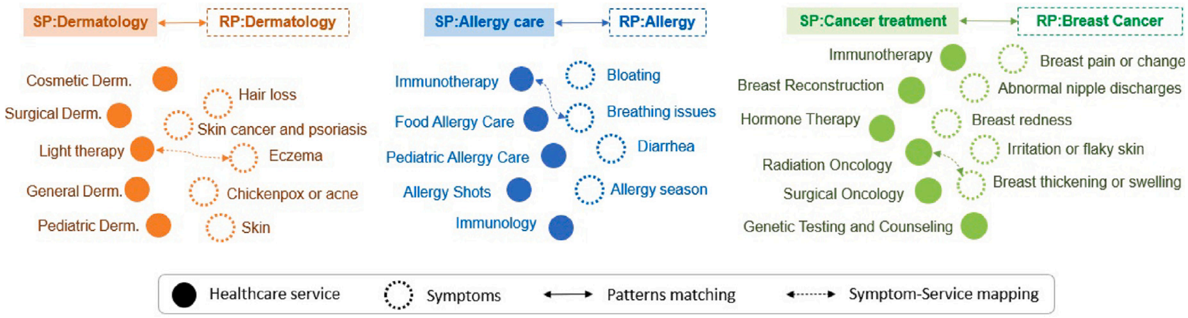

Fig. 2. Pattern-based classification of HKG entities.

4.2. Incremental representation learning of the healthcare network

Since information networks are large in size, heterogeneous, domain-oriented, complex and ever changing in their structure, exploring the semantic relationships among their entities is a challenging and time-consuming task. In our case, healthcare information networks contain different types of nodes (medical services, smart health objects, symptoms, patients' data, etc.). In addition, their structure changes over time, by the addition/removal of healthcare entities, updates of patients' related information, etc.

The drastically increasing number of information networks and their challenging characteristics have boosted the reasoning over information networks in a low-dimensional vector representation, while preserving the initial topology and content of information networks [16]. This manner of handling complex data is called *network representation learning*, and has been successfully applied in various fields, including context-aware service recommendation [50], citation recommendation [53], collaborative filtering [51], recommendations in smart cities [59], Web service classification [60], risk prediction [61], human mobility prediction [62], and rescue route selection [63]. Network representation learning has several advantages, mainly the improved performance of the features learning task, the better interpretation of latent knowledge, and the automatic feature learning even from unlabeled data.

In our approach, we exploit the strengths of network representation learning [16] to encode healthcare entities (e.g., similar services or smart health objects) with similar features and semantics in shared vectors. Inspired by the incremental learning method defined in [64], we map the healthcare information network into a *cuboid space*. In this structure with three directional components (rows, columns and pages), we adjust the location of healthcare entities in an iterative manner, based on their semantics and their similar features. This routine will arrange the healthcare network entities in the same directional components, which facilities their location, processing and evaluation during the healthcare workflow recommendation step.

Based on an existing healthcare taxonomy,[2] Fig. 3 shows an example of healthcare network mapping to a cuboid structure.

As depicted in Fig. 3, the entities of the healthcare knowledge graph are arranged on three cuboid dimensions, i.e. *rows*, *columns*, and *pages*. In this case, only $3\sqrt[3]{n}$ unique vectors are needed to encode the node embeddings instead of n vectors, unlike traditional representation learning methods [16].

[2] The healthcare taxonomy is collected from *Sutter Health*, a not-for-profit organization.

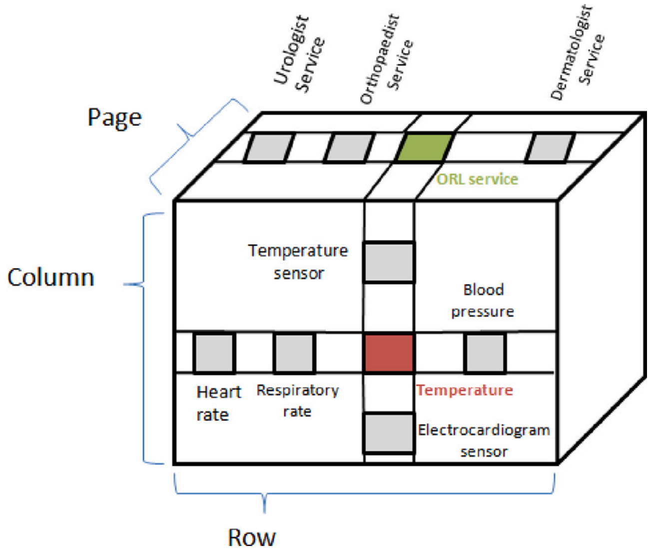

Fig. 3. Healthcare cuboid space.

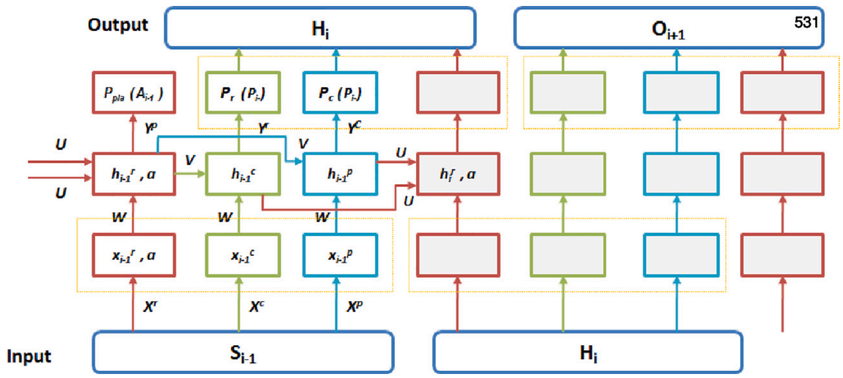

Fig. 4. RvNN architecture for healthcare network representation learning.

4.2.1. Learning component embeddings

The learning of healthcare entities' representation is, first, preceded by mapping the content of $\mathcal{G}(t)$ into an initial cuboid structure. $\mathcal{G}(t)$ can be seen as the first, i.e. static, snapshot of the healthcare network. Then, the cuboid space will undergo continuous adjustments by applying an incremental learning w.r.t. the healthcare network evolution, i.e. $\mathcal{G}(t+1)$. To determine the semantic and structural relationships between the healthcare network's entities and to extract each of these latter's neighborhood, we adopted a meta-path guided random walks. In this strategy, a meta-path is a sequence of typed relations that connects pairs of nodes. The sequences that result from the guided random walks are, then, employed to learn the node embeddings using a recursive variant of neural networks, namely RvNN [17,18]. This choice is justified by RvNN ability to model complex hierarchical structures. As a kind of tree-structured neural networks [17], RvNN is a hierarchical learning model, where input data are treated in a tree fashion, that is, each node's representation is deduced from its direct children (see Fig. 4).

Consider the example of smart healthcare network in Fig. 5. The network schema (see Fig. 5.*a*) includes three types of nodes: smart medical objects $\mathcal{O}$, symptoms $\mathcal{S}$, and smart healthcare services $\mathcal{H}$. Nodes are connected via different types of edges, such as *Treated-By*, *Require*, and *Cause*. For example, the metapath $\mathcal{P}_1$ in Fig. 5.*b* indicates that a symptom $s \in \mathcal{S}$ is treated by a healthcare service $h \in \mathcal{H}$, and requires the use of a medical equipment $o \in \mathcal{O}$. Whereas the metapath $\mathcal{P}_2$ refers to the occurrence of a symptom $s \in \mathcal{S}$ and the application of a specific treatment $h \in \mathcal{H}$ that may cause new symptoms.

To build the initial cuboid structure, *row, column* and *page* vectors are randomly initialized. Then, the nodes that form the HKG are randomly assigned to these three dimensions of vectors, to have a $3-C$ representation. By considering the order of typed nodes in the above meta-paths, a *cuboid space* will be constructed by an iterative adjustment of the nodes locations. This process can be seen as a node placement problem. In fact, given a node vector and a metapath $\mathcal{P}$ in the cuboid C, we employ an encoder–decoder scheme to compute the probability that a node is the next one according to $\mathcal{P}$. Taking the metapath $\mathcal{P}$*: "S-H-O"* in Fig. 5.*b* as example. To determine the location of a symptom $s \in \mathcal{S}$ in the cuboid space, the idea is to predict the probability of each healthcare

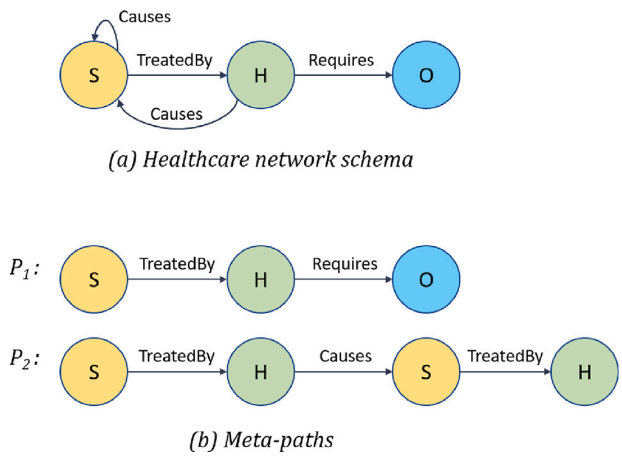

Fig. 5. Example of meta-paths in the healthcare knowledge graph.

service ($h \in \mathcal{H}$) to be the next node when seeing s. In the same fashion, the probability of a smart medical equipment $o \in \mathcal{O}$) to be the next node when seeing h is computed w.r.t. the metapath $\mathcal{P}$: *"S-H-O"*.

A node will take place in the cuboid C by maximizing the likelihood of being a next node in a metapath $\mathcal{P}$ and, in the same direction, by minimizing the negative log-likelihood (NLL) of the next node in the node sequence. That is achieved through the optimization of the cross-entropy between the target probability distribution and the predicted one, where:

$$NLL = \sum_{t=1}^{T} -\log P_r(N_t) - \log P_c(N_t) - \log P_{p|a}(N_t) \tag{4}$$

Here, P_r, P_c, and $P_{p|a}$ denote, respectively, the row, column and page probabilities with a given node type, T is the length of node sequences, and N_t refers to the tth element in the node sequences.

Learning of healthcare entities' embeddings. Let n be the input vector's dimension, be it a row, column or page vector, a the type of a healthcare network entity (symptom, healthcare service, or smart medical object), m the dimension of a hidden state vector. We learn the representation of a healthcare service node h, given a symptom node s, as follows: for the input node S_{i-1} of meta-path P_1 : *"**S-H**-O"*, estimating the probability of the second node in P_1 depends on several parameters (see Fig. 4), including (i) the column, page, and row vectors $x^c_{i-1} \in R_n$, $x^p_{i-1} \in R_n$, $x^r_i \in R_n$, which are derived from the input embedding matrices $X^c, X^r, X^p \in R^{n\times\sqrt[3]{|\mathcal{V}|}}$, respectively; (ii) the hidden state vectors $h^c_{i-1} \in R_m$, $h^p_{i-1} \in R_m$, $h^r_i \in R_m$; and (iii) the state value of the node type a. The above hidden state vectors can be produced using the recursive operations below:

$$h^c_{i-1} = \sigma(W x^c_{i-1} + V h^r_{i-1} + b) \tag{5}$$

$$h^p_{i-1} = \sigma(W x^p_{i-1} + V h^r_{i-1} + b) \tag{6}$$

$$h^r_i = \sigma(W x^r_i + U(h^c_{i-1} + h^p_{i-1}) + b) \tag{7}$$

where $W \in R^{m\times n}$, $U, V \in R^{m\times n}$, $b \in R^m$ denote the parameters of affine transformations, while σ is a non-linear activation sigmoid function.

The probability $P(v_i)$ of a node v_i is computed using three conditional probability values: $P_r(v_i)$ (row probability), $P_c(v_i)$ (column probability), $P_{p|a}(v_i)$ (page probability) [64], with the same type a as P_i:

$$P_r(v_i) = \frac{exp(h^c_{i-1}.y^r_{r(v_i)})}{\sum_{i\in S_r} exp(h^c_{i-1}.y^r_i)} \tag{8}$$

$$P_c(v_i) = \frac{exp(h^p_{i-1}.y^c_{c(v_i)})}{\sum_{i\in S_c} exp(h^p_{i-1}.y^c_i)} \tag{9}$$

$$P_{p|a}(v_i) = \frac{exp(h^r_i.y^p_{p(v_i)})}{\sum_{i\in S_{p|a}} exp(h^r_i.y^p_i)} \tag{10}$$

$$P(P_i) = P_r(v_i).P_c(v_i).P_{p|a}(v_i) \tag{11}$$

where $r(N_i), c(N_i), p(N_i)$ denote respectively the row, column, and page indices of N_i, whereas $y^r_i, y^c_i, y^p_i \in R^m$ are, respectively the ith vectors of $Y^r, Y^c, Y^p \in R^{m\times\sqrt[3]{|V|}}$. The rows, columns, and pages, of the cuboid are denoted by S_r, S_c, and S_p.

4.2.2. Optimization of healthcare nodes' placement

Once placed in the cuboid space, the location of each healthcare node could be adjusted, so to optimize the loss function and reduce the information loss. This could be achieved by grouping the nodes with similar semantics, such as the healthcare services which treat the same symptoms or require similar medical equipment.

Take the example in Fig. 3. During training, we move the *Pulmonology* node from its initial position to the jth column, which is shared by other treatment tasks, i.e. healthcare services, like *DLCO*, *CPET*, *Plethysmography*. Likewise, the node *General Dermatology*, which was initially placed at the jth column shares similar characteristics with *Cosmetic Dermatology*, *Pediatric Dermatology*, *Surgical Dermatology*, and *Light Therapy* services, which belong to another column. Such similarities include the commonly treated symptoms (e.g., skin, hair loss, eczema, skin cancer and psoriasis, chickenpox or acne) or the smart medical objects. So, the node *General Dermatology* can be moved and placed together with those nodes. Doing so, similar services, symptoms, or smart medical objects will be placed in the same directional component, being a row, column or a page component.

Based on the above example, the idea is to move a node v in the HIN from $(r(v), c(v), p(v))$ to (i, j, k), which denote, respectively, the node's initial and new location in the cuboid. That is by computing the new loss values $l_r(v,i)$, $l_c(v,j)$, and $l_p(v,k)$ for the row, column and page components, respectively. By this way, the total loss for a moved node is computed as: $l(v,i,j,k) = l_r(v,i)+l_c(v,j)+l_p(v,k)$. The loss at each dimension could be computed to predict the next node based on the probabilities computed for each object in the node sequences. For instance, $l_r(v,i)$ is the sum of $-log(\frac{exp(h^c_{t-1}\cdot y^r_{r(n)})}{\sum_{i\in S_r} exp(h^c_{t-1}\cdot y^r_i)})$. After calculating the total loss for all the possible new movements, the nodes placement can be translated into the following optimization problem [64]:

$$\begin{aligned}
&min_g \sum_{(v,i,j,k)} l(v,i,j,k)g(v,i,j,k), \;\; subject\ to \\
&\sum_{(v,i,j,k)} g(v,i,j,k) = 1, \;\; \forall v \in V \\
&\sum_{(v)} g(v,i,j,k) = 1, \;\; \forall i \in S_r, j \in S_c, k \in S_p, \\
&g(v,i,j,k) \in (0,1), \;\; \forall v \in V, i \in S_r, j \in S_c, k \in S_p
\end{aligned} \tag{12}$$

Here $g(v,i,j,k) = 1$ indicates the location of a node, v represent the three directional components (row, column and page) (i,j,k), whereas S_r, S_c and S_p denote, respectively, the sets of nodes in those three directions.

To solve the healthcare nodes' reallocation in the cuboid space, we translated this optimization problem into a minimum weight perfect matching (MWPM) problem [65]. MWPM could be realized using different algorithms (e.g., MCMF, IPGA). We choose the Improved Path Growing Algorithm [66] thanks to its accuracy and high computational efficiency. So given the set of healthcare nodes $v \in \mathcal{V}$, their locations $(i,j,k) \in S_r \times S_c \times S_p$, and their loss values $l(v,i,j,k)$, we constructed a weighted bipartite graph $\mathcal{B} = (\mathbb{V}, \mathcal{E})$, with $\mathbb{V} = (\mathcal{V}, S_r \times S_c \times S_p)$ and $l(v,i,j,k)$ is the edge weight. Based on that, the goal is to find the set of edges that satisfy two conditions: (i), all vertices in $\mathcal{B}$ are matched, and (ii) the sum of weights $(\sum_{v,v_{ijk}} l(v,i,j,k))$ is as small as possible.

4.2.3. Modeling and learning of changes in the healthcare network

In front of the ever changing healthcare network, we need to continuously update the existing embeddings w.r.t. the nodes/edges added, removed or updated in the healthcare network. This task is achieved by applying dynamic meta-path guided random walks, in order to handle the changes seen at time $t+1$ (e.g., new medical objects, unavailability of healthcare services, changes in service cost, etc.). Such changes are formulated as a nodes subset $\mathcal{V}'$ and a new sequence length T', given an increment Δ, wherein $\mathcal{V}' = \mathcal{V} + \Delta\mathcal{V}$, $T' = T + \Delta T$. T and $\mathcal{V}$ are, respectively, the sequence length and the healthcare network content seen at time t.

With the evolution of the healthcare network topology and content, the cuboid structure must be adjusted to correctly learn the new node embeddings. To do so, the loss function is adapted by incorporating the amount of captured changes in $\mathcal{G}(t+1)$:

$$NLL' = \sum_{t=1}^{T} -\log P(N_t) + \sum_{t=1}^{T+\Delta T} -\log P(N_t) \tag{13}$$

where N_t refers to the tth node in the new generated node sequences.

At this stage, we do not need to map the whole network content into a new cuboid space. Rather, we inherit the parameters and vectors for the healthcare network $\mathcal{G}(t)$, observed before capturing the new changes ($\Delta\mathcal{G}$). These parameters include the input embedding matrices (X^c, X^r, X^p), the vectors sets for the row, column, and page components (X^c, X^r, X^p), and the parameters of affine transformations ($\boldsymbol{W,U,V}$, and b). The goal is to initialize the newly added vectors (representing healthcare entities) with the same settings in the loss function.

Since the above parameters and embedding vectors may introduce large errors when used in the first part of Eq. (13) ($\sum_{t=1}^{T} -\log P(N_t)$), we minimize the loss function by calculating the difference $\Delta NLL'$, which is then reduced and calibrated using the standard SGD method:

$$\begin{aligned}
\Delta NLL' = | \sum_{t=1}^{T} (\log \sum_{i\in S'_r} exp(h^c_{t-1} y^r_i) + \log \sum_{i\in S'_c} exp(h^p_{t-1} y^c_i) + \\
\log \sum_{i\in S'_p} exp(h^r_t y^p_i)) - \log RCP|,
\end{aligned} \tag{14}$$

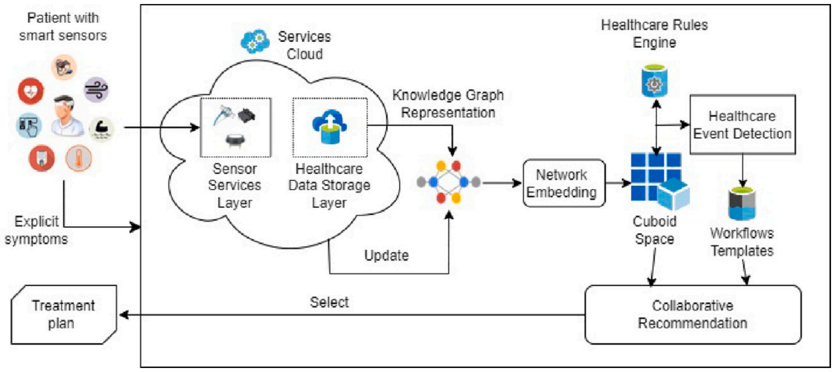

Fig. 6. Smart services recommendation process.

where $R = \sum_{t=1}^{T}\sum_{i\in S_r} exp(h^c_{t-1} y^r_i)$ denotes the *rows* directional components in the cuboid, $C = \sum_{t=1}^{T}\sum_{i\in S_c} exp(h^p_{t-1} y^c_i)$ denotes the *columns* components, whereas $P = \sum_{t=1}^{T}\sum_{i\in S_r} exp(h^r_t y^p_i)$ corresponds to *pages* components. Some of the parameters and vectors in R, C, and P ($y^c_i \in S'_{c'}$, $y^r_i \in S'_{r'}$ and $y^p_i \in S'_{p'}$) are updated via SGD, with keeping the rest of inherited parameters unchanged. This accelerates the learning and updating of component embedding vectors.

As for the second part of Eq. (13) ($\sum_{t=1}^{T+\Delta T} - \log P(N_t)$), the new changes ($\Delta T$) in the HKG are employed to train and update the parameters and vectors with SGD, following the same logic above.

Dynamic node placement in the healthcare cuboid. Like in the learning of static embeddings from the initial cuboid structure (see Section 4.2.2), the nodes placement after a captured change in the healthcare network should be incrementally adjusted. Since IPGA' algorithm is not designed for dynamic bipartite graphs, we adopt a *Edmonds-Karp Shortest Path Faster Algorithm* (EK-SPFA) [67]. Using this algorithm, in addition to the objective function Eq. (13) defined for the incremental learning, the nodes placement are adjusted taking into account the healthcare network changes.

To incrementally learn the new representations of healthcare entities, we consider these latter's changes in each row, column, and page in C. Such changes can be formulated as: $S'_d = S_d + \Delta S_d$, where d denotes a row, column or page components, and S_d represents the set of nodes in each of those components. The whole parameters are employed in the computing of ΔT in the loss function (Eq. (13)).

Once constructed, the cuboid space is inputted to the recommendation algorithm (See Fig. 6), rather than exploring the complex graph-like healthcare network.

5. Smart services recommendation

The aim of this section is to propose a smart healthcare recommendation algorithm that, given a set of symptoms denoting the patient's health state and a cuboid space representing the healthcare network, generates an executable workflow of treatment services, together with their associated smart medical objects.

As depicted in Fig. 6, the core actors of the healthcare recommendation systems are (i) sensors to collect patients' data, (ii) sensor cloud services to analyze, filter, and aggregate patient-generated health data, (iii) smart healthcare services to recommend care actions to patients, and (iv) smart medical objects which are needed by healthcare services, for example to track patients' adherence to treatment or medication plans. The patient-generate data include treatment history and symptoms, which can either be collected using IoT wearables or reported by patients.

The three-step process in Fig. 6 indicates that *network embedding* is the first step that consists in transforming the healthcare knowledge graph into a cuboid space, to learn a rich representation of its entities (medical services, healthcare smart objects, patients' symptoms, etc.) and to efficiently filter the recommended healthcare entities (i.e. services and smart objects). Then, a *selection phase* is triggered in order to choose the appropriate healthcare workflow according to the patient's current state, i.e. detected symptoms. Finally, a *collaborative recommendation* is performed to explore the cuboid space and find the suitable services and healthcare IoT objects for each abstract task in the healthcare workflow.

By analyzing the detected symptoms and the relationships between them, the healthcare recommendation system can deduce the related diseases (e.g., cancer, obesity, heart and cardiovascular problems, diabetes) and recommend the appropriate healthcare plan for the medical staff, before patients' status get complicated. The proposed system will provide patients with personalized guidance (through predefined medication plans) in response to the observed symptoms. In our work, we design medication plans as abstract workflows, i.e., sets of healthcare tasks. Each of these latter will be instantiated to a healthcare service which consumes specific IoHT devices, depending on the patient's health status. Examples of healthcare workflows include preventive care, short-term care, home-based care, proactive treatment plans, etc.

As we mentioned in Section 3, the goal of our framework is to recommend a set of healthcare services, together with their needed medical IoT objects, given a set of symptoms and an abstract workflow representing the recommended execution pattern of the required services. To meet this goal, we defined an algorithm that explores the cuboid space in order to locate each detected symptom and evaluate its corresponding healthcare services/objects to, finally, aggregate the best ones w.r.t. to the recommended execution pattern.

Algorithm 1 takes, as input, a set $\mathcal{P}$ of symptoms and the cuboid $\mathcal{C}$. It returns a composite healthcare service $\mathcal{W} = \langle \mathcal{O}_p \cup \mathcal{S}_p \rangle$, with $\mathcal{O}_p$ and $\mathcal{S}_p$ denoting, respectively the selected smart medical objects and services that fit the patient's health state.

Algorithm 1 Selection of treatment plan

1: **Input :** $\mathcal{S}_p = \{s_1, s_2, ..., s_n\}$: set of detected symptoms, $\mathcal{C}$: cuboid space
2: **Output:** $\mathcal{W}^p =< \mathcal{H}^p, \mathcal{O}^p >$: Treatment workflow of healthcare services ($\mathcal{H}$) and smart medical objects ($\mathcal{O}$).
3: $\mathcal{H}^p \leftarrow \emptyset$, $\mathcal{O}^p \leftarrow \emptyset$
4: **for** each $s \in \mathcal{S}_p$ **do** ▷ For each detected symptom
5: Locate $s(x_i^r, x_j^c, x_k^p)$ in $\mathcal{C}$ ▷ Locate the symptom's $3-\mathcal{C}$ representation in the cuboid space
6: $\mathcal{H}_s \leftarrow \mathcal{H}(x_i^r) \cup \mathcal{H}(x_j^c) \cup \mathcal{H}(x_k^p)$ ▷ Group all the candidate healthcare services that have the symptom s in their descriptions
7: $\mathcal{O}s \leftarrow \mathcal{O}(x_i^r) \cup \mathcal{O}(x_j^c) \cup \mathcal{O}(x_k^p)$ ▷ Group all the smart medical objects that can be used in the treatment of the symptom s
8: **for** each $h < \mathcal{F}^h, \mathcal{Q}^h, \mathcal{O}^h > \in \mathcal{H}_s$ **do**
9: **if** $\mathcal{A} \cap \mathcal{F}^h \neq \emptyset \vee \mathcal{A} \cap \mathcal{O}^h \neq \emptyset$ ▷ There is some treatment conflicts **then**
10: Remove h from $\mathcal{H}_s$
11: **else**
12: $h_{score} \leftarrow$ Compute score of h using Eq. (15)
13: **end if**
14: **end for**
15: Sort $\mathcal{H}_s$ in a decreasing order of h_{score}
16: $h_{best} < \mathcal{F}^h, \mathcal{Q}^h, \mathcal{O}^h > \leftarrow \mathcal{H}_s[1]$
17: **if** $h_{best} \notin \mathcal{H}$ **then**
18: add h_{best} to $\mathcal{H}^p$
19: $\mathcal{O}^p \leftarrow \mathcal{O}^p \cup O^h \cap \mathcal{O}^s$
20: **end if**
21: **end for**
22: Return $\mathcal{W}^p < \mathcal{H}^p, \mathcal{O}^p >$

Algorithm 1, first, explores the cuboid space to locate each detected symptom $s \in \mathcal{S}_p$ (lines 4–5). Then based on these latter's $3-C$ representations (x_i^r, x_j^c, x_k^p), the close healthcare services that have the symptom s in their descriptions, as well as the medical equipment (smart medical objects) that can be used in the treatment of the symptom s, are saved in the sets $\mathcal{H}_s$ and $\mathcal{O}_s$, respectively (lines 6–7). After that, candidate healthcare services in $\mathcal{H}_s$ are checked against the patient's constraints (e.g., allergy, conflicts between drugs), and the ones that violate those constraints are eliminated from $\mathcal{H}_s$ (lines 8–10). Otherwise, a score is calculated for each of the remaining services (line 12), which will be, next, sorted in a decreasing order of their probability likelihoods and their adequacy to the patient's constraints (line 15). These values denote the degree of closeness between a candidate service/object and the detected symptom $s \in \mathcal{S}_p$ in the cuboid space $\mathcal{C}$, which reflects their ability to treat the symptom. Therefore, the healthcare service $h_{best} \leftarrow H_s[1]$ (respectively smart medical object $\mathcal{O}^h \cap \mathcal{O}^s$) with the highest score will be added to the final set H (respectively O_p), if it does not already exit in $\mathcal{H}$ (lines 17–20). The whole routine is repeated for each detected symptom, and the best healthcare services/objects that already belong to H and O will be added only once, as they will handle more than one symptom (line 17). Finally, the appropriate healthcare services together with their smart medical equipment are outputted as the composite treatment workflow $(\mathcal{W}\langle \mathcal{H}, \mathcal{O} \rangle)$ for the patient's current health situation $(\mathcal{S}_p = \{s_1, s_2, \ldots, s_x\})$.

To ensure a trade-offs between the patient's constraints, the time and cost of each treatment task are normalized and associated with two weights (α and β), denoting their importance degrees. Eqs. (15) and (16) compute, respectively, the score of each recommended healthcare service h and the whole treatment plan $\mathcal{W}^p$.

$$score(h) = \alpha.\mathcal{T}_h + \beta.\big(cost(h) + \sum_{o \in \mathcal{O}^h} cost(o)\big) \tag{15}$$

$$score(\mathcal{W}^p) = \alpha. \sum_{h \in \mathcal{H}^p} \mathcal{T}_h + \beta. \sum_{h \in \mathcal{H}^p} \big(cost(h) + \sum_{o \in \mathcal{O}^h} cost(o)\big) \tag{16}$$

In the above equations, $cost(h)$ and $cost(o)$ denote, respectively, the costs of a healthcare treatment h and the medical objects $\mathcal{O}^h$ required by h. $\mathcal{T}_h$ is the required treatment time to reach a stable health state for a patient.

6. Theoretical complexity study

The adopted network representation learning method allowed putting the healthcare network entities (services, smart medical equipment, symptoms), with similar semantics, in the same directional component, to be mapped as close as possible in the cuboid space. This strategy has optimized the process of exploring the healthcare network and recommending the suitable treatment plan according to the patient's health situation. However, this process comes with a cost, as the frequent changes in the healthcare network require a continuous re-adjusting the nodes location in the cuboid space. In this section, we study the complexity of incremental learning and its impact on the healthcare service recommendation task.

- *Incremental learning:* The time complexity of the embedding model depends on the number of nodes (n) in the healthcare knowledge graph, as well as the length of node sequences (T). The complexity comes from (i) learning on the component embeddings in the cuboid representation: $\mathcal{O}(\sqrt[3]{n}.T)$, and (ii) nodes reallocation: $\mathcal{O}(n^2)$. Taking into account the number of training epochs (k), the overall complexity of the embedding process is in the order of $\mathcal{O}((\sqrt[3]{n}.T + n^2)K)$, i.e. $\mathcal{O}(n^2)$ if ($n > T$). Since the incremental learning step, given the changes ($\Delta\mathcal{G}$) in the healthcare network, is based on the shortest path faster algorithm (SPFA), this latter's time complexity depends on the healthcare network size (n) as well as the number of connections ($\mathcal{R}$) in G. Also, the nodes adjustment in the cuboid structure is treated as a standard minimum weight perfect matching problem, with linear-time approximation. Therefore, the overall time complexity is in the order of $\mathcal{O}(n^2.|\mathcal{R}|)$.
- *Treatment plan's recommendation:* The complexity of recommendation mainly depends on the number of symptoms ($|\mathcal{S}_p|$) and the cost of exploring n nodes (services, smart medical equipment, symptoms) in the cuboid space $\mathcal{C}$. The algorithm runs over the cuboid space to locate the $3-C$ vector representation of each symptom $s \in \mathcal{S}_p$, and to extract the candidate healthcare services that have the symptom s in their description. This operation takes $O(|\mathcal{S}_p|.\sqrt[3]{n})$, as the embedding vectors in $\mathcal{C}$ are shared among nodes in the same directional component: $s(x_i^r, x_j^c, x_k^p)$. In the second part of Algorithm 1, each candidate service is evaluated against the patient constraints ($\mathcal{A}$), scored, ranked, and the best one (h_{best}) is finally saved in the set $\mathcal{H}$ of recommended healthcare services. This block of instructions takes $O(|\mathcal{H}_s|.|\mathcal{A}|)$ in the worst cases, where $\mathcal{H}_s \subset \mathcal{H}$. The whole complexity is in the order of $O(|\mathcal{S}_p|.|\mathcal{H}_s|.|\mathcal{A}|.\sqrt[3]{n})$.

7. Experimental results

To implement the whole recommendation process, we used Google Colab as a Python development environment that runs on Google Cloud. Due to the large number of healthcare datasets and the diversity of their content, we started by collecting useful data on healthcare services, smart medical objects and devices, and symptoms from various sources. Then, we combined the collected data to create the healthcare information network (denoted as $\mathcal{G}$), following the meta-paths in Fig. 5. The used data was in csv format, essentially based on PrimeKG[3] and other resources like UMLS[4] and the IoT Healthcare Security Dataset.[5] PrimeKG manages to combine 20 high-quality biomedical resources to define 17 080 diseases, including rare diseases, and 4 050 249 relationships across ten major biological scales supplemented with text descriptions of clinical guidelines for drugs and diseases. The used dataset comprises 3000 samples with various parameters describing the relations between diseases patterns, symptoms patterns, treatment patterns and associated objects.

To conduct our experiments while taking into account the amount of changes in the healthcare network, we generated five sub-networks (denoted as $\mathcal{G}'$) by extracting $(100 - \Delta)\%$ from the collected dataset, i.e. the build knowledge graph, where Δ is the change rate (in $[5\%, 10\%, 15\%, 20\%, 25\%]$) in the healthcare network. To evaluate the effectiveness of our solution, we compared the static and incremental version with a traditional recommendation approach that adopts a matrix view.[6] We also used four metrics to record accuracy results: Precision, Recall, MRR, and F1.

7.1. Recommendation quality evaluation

7.1.1. Impact of user request

In these tests, we evaluated the ability of our approach to return accurate and high-score treatment plans. To do so, we varied the number of symptoms between 2 and 10 for different cuboid sizes and change rates (in [5%,25%]). Results are depicted in Figs. 7 and 8.

It is obvious, from Fig. 7, that the recommendation by mining the cuboid space is better than the traditional recommendation method, which follows the simple matrix representation. That is could be attributed to the ability of our method to map similar healthcare entities into the same vector components (3-C: row, column, page). We also notice that accuracy measures are quite close for both static and incremental embedding of the healthcare network. Fig. 7 also shows the degradation of recommendation quality with larger cuboid spaces. In fact, such behavior is explained by the iterative adjustment of each healthcare entity's placement in the cuboid space. But in the end, we could gradually improve accuracy with a higher number of training iterations.

We also studied the impact of user requests' complexity (symptoms number) on the recommendation accuracy, as depicted in Fig. 8. It is clear that our algorithm performs well regardless of the complexity of the patient's query. We also notice that accuracy results slightly decreased with higher numbers of symptoms, but without a real influence on the algorithm's performance. Fig. 8 also shows that the captured changes on the cuboid have a little impact on the recommendation quality, as the placement of most nodes was not affected by the changes. Add to that, the incremental embedding of the captured changes will inherit the same parameters, which provides accurate results for the entities that remain unchanged.

3 https://github.com/mims-harvard/PrimeKG/tree/main.
4 https://github.com/dhchenx/umls-graph.
5 https://github.com/imfaisalmalik/IoT-Healthcare-Security-Dataset/tree/main.
6 CARSKit - https://github.com/irecsys/CARSKit.

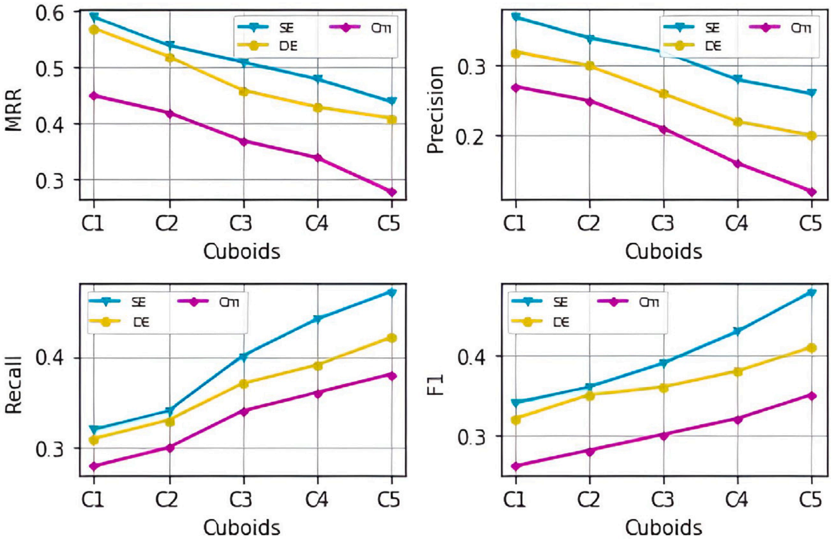

Fig. 7. Recommendation quality with different cuboid sizes.

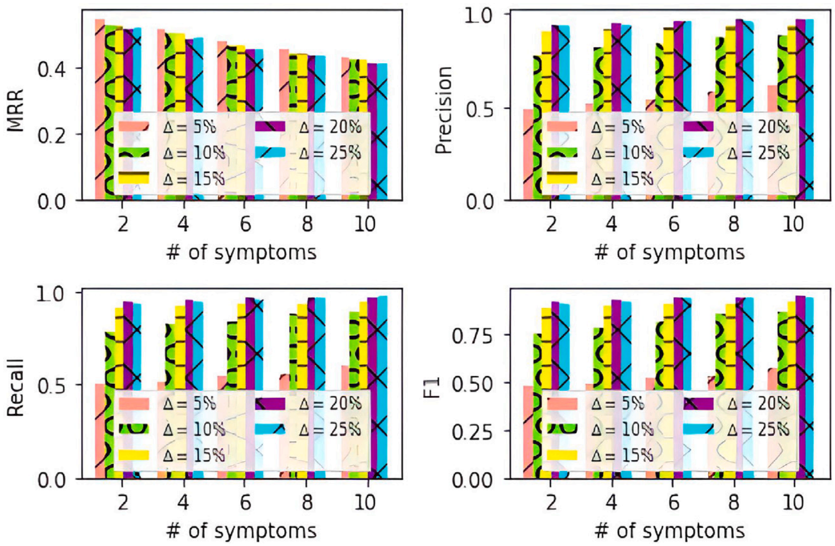

Fig. 8. Recommendation quality with different cuboid sizes.

7.1.2. Impact of 3-C representation

The goal of these test series is to study the recommendation accuracy based on two parameters: the number of returned services (k) and the cuboid size ($|C|$). Accuracy measures were recorded for both embedding-based and graph-based methods (see Fig. 9).

As expected, the cuboid representation has improved the arrangement of the healthcare network elements, and produced better results compared to the graph-based method. Our approach performed well, regardless of the number of returned services (k). However, we noticed a decrease in the recommendation accuracy, when k tends to be higher. For example, the MRR and precision measures have reached 0.346, for two services, and did not go below 0.319 for $k \leq 5$. Contrariwise, a higher k value means that more healthcare services with low quality will be included in the recommendations list. The cuboid size adds another factor of inaccuracy, as depicted in Fig. 9. That is the case of $C1$ and $C2$ test cases, for which the *3-C* representation was influenced by the lack of sufficient features for each node's neighborhood. That was not the case with large cuboid spaces, where better semantics of the learned embeddings were reached thanks to the metapath-guided generation of node sequences.

We should mention that the cuboid size reflects the amount of knowledge in the healthcare information network. If this latter is small, the *3-C* representation of healthcare entities will suffer from inaccurate semantics, which will affect the recommendation's precision. Contrariwise, the HKG density will translate into a higher number of complete metapath-based node sequences to be fed into the embedding model, thus, correctly learning the *3-C* representations of HKG entities. As a result, the node placement in the

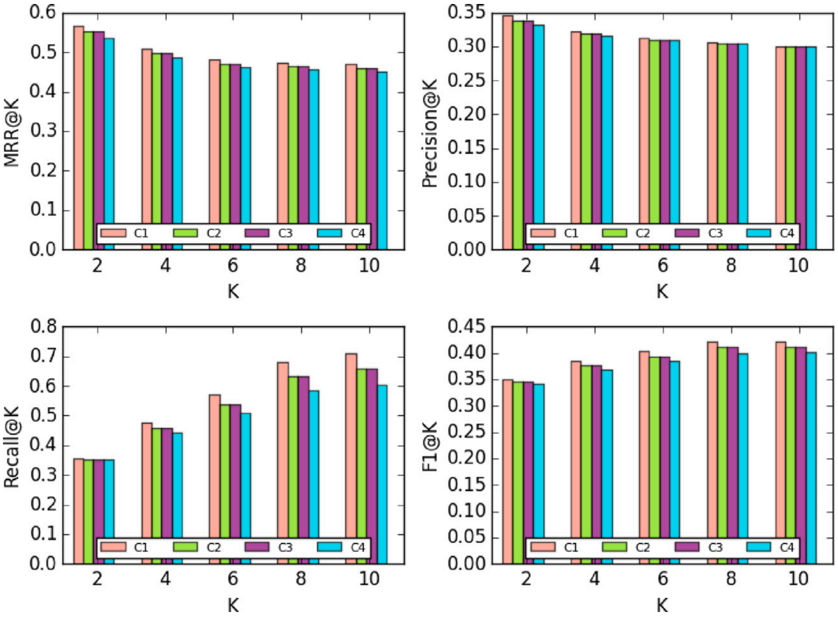

Fig. 9. Recommendation accuracy with different numbers of *top-k* services.

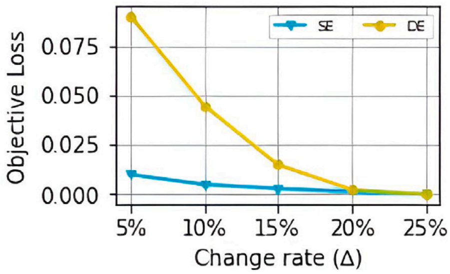

Fig. 10. Impact of the change rate on the model effectiveness.

cuboid will be optimal and the suitable healthcare services will be closer to their associated symptoms, as connected in the original HKG. That was noticed from the accuracy measures, even with increased k values.

7.1.3. Impact of change rate

In this section, tests with and without the incremental learning step were conducted, to measure the impact of the change rate on the solution quality. The objective loss, for both static (SE) and incremental (DE) embedding, are depicted in Fig. 10.

It is obvious that the embedding model performs better with a static structure of the healthcare network. In fact, the loss when learning the representation of each entity did not exceed 0.011 for all the test cases. Such behavior was noticed only for high degrees of changes in the case of incremental learning. Indeed, a low degree of change (e.g., 5% and 10%) does not provide a good training base. Consequently, the optimization scheme of the nodes placement will almost consider the same *3-C* representation for the healthcare entities that have been affected by changes. That is why the static and incremental models tend to similar performance as the number of affected nodes soars. That produces different *3-C* representations which will result in a correct placement optimization scheme.

7.2. Computation time complexity

In these experiments, the recommendation time depends on three main parameters: (i) the patient's symptoms ($|S_p|$), (ii) the cuboid size ($|\mathcal{C}|$), and (iii) the amount of changes in $\mathcal{C}$ (Δ). The total computation time is defined as $\mathcal{T} = \mathcal{T}_l + \mathcal{T}_r$, where $\mathcal{T}_l$ and $\mathcal{T}_r$ denote, respectively, the learning and recommendation times. We varied the number of symptoms in [2..10] and we recorded the computation time for 5 cuboid structures $\mathcal{G}(t)$, which we extracted from the original healthcare information network. Δ was varied in $[5\%, 10\%, 15\%, 20\%, 25\%]$, with $\mathcal{G}(t) = \mathcal{G}(t-1) + \Delta\mathcal{G}(t-1)$. For example, when $\Delta = 25\%$, this means that tests were conducted with

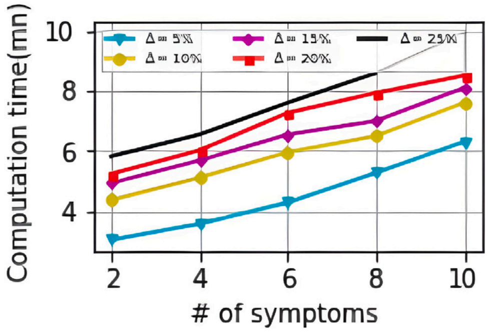

Fig. 11. Computation times with different numbers of symptoms and cuboid spaces.

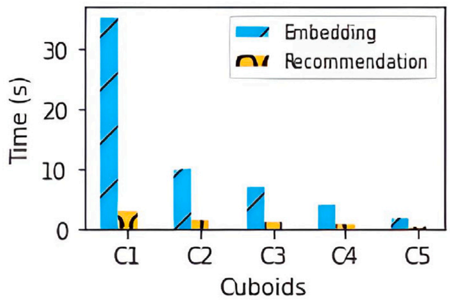

Fig. 12. Training and recommendation performance with different cuboid sizes.

75% of the healthcare-related nodes, in addition to the randomly eliminated Δ content (incremental embedding). Then, tests were repeated for the whole healthcare information network (static embedding).

Fig. 11 shows that the computation time is proportional to the number of treated symptoms. Indeed, each of these latter must be located in the cuboid, so that its related healthcare services are evaluated. On the other side, the higher is the change rate in the cuboid space, the costly is the task of adjusting the placement of healthcare entities and locating the input symptoms to perform recommendation.

We also evaluated the computational complexity in terms of incremental model training and exploring the generated cuboid to locate the detected symptoms and their treatment plans. Fig. 12 depicts the training and recommendation times for different cuboid structures.

We can clearly notice that the training step takes most of the computation time. Also, the incremental behavior of the proposed model represents an additional factor of complexity. In fact, the model involves various operations, including the learning of component embeddings, the optimization of nodes' placement, and the learning of changes in the healthcare network. Fig. 12 also shows that the training time is proportional to the healthcare network's size ($|C|$) and the captured changes (Δ). That is due, not only to the costly operation of processing the different nodes and their neighborhood, but also to the adjusting of their placement in the cuboid. The training times with small-size cuboids are generally close. On the other side, the cuboid size had no significant impact on the recommendation time, which always was in $[0.35, 1.15]$ s. Regardless of the cuboid content and organization, the recommendation algorithm locates the *3-C* vector representation of each input symptom, and limits the evaluation of candidate services and smart medical objects to the corresponding row, column, and page components. Therefore the recommender system's performance is almost dependent on the incremental learning phase.

8. Future directions

8.1. Domain-specific healthcare recommendation

In the present work, we addressed the general aspects of healthcare service recommendation, regardless of the targeted item (e.g., medications, proper diets, doctors, healthy roads). However, each recommended item has its specificity and own features, which is the case of walking routes. These latter can be recommended to solve mobility issues of the citizens that suffer from high concentrations of allergens. Route recommendation can also help shorten the rescue time of ambulances, especially for emergency medical services [63]. As a healthcare recommendation case, the rescue route selection can be incorporated into our healthcare recommender system to suggest the suitable medical services w.r.t. the patient context (i.e. spatio-temporal data). By this way, additional features will be incorporated into the healthcare information network, such as the route length, the roads priority, the roads maximum speed, etc. Add to that, the route sequences can be generated by guided random walks to learn their corresponding representation vectors (encoded 3-C), which will facilitate their arrangement in the cuboid space and, consequently, the selection of roads that form the best rescue route.

8.2. Risk assessment in healthcare recommendation

Besides the specificity of the healthcare items, the costs and risks that may be caused by these latter should be predicted before recommending a specific healthcare service. In fact, the recommended items' evaluation often is performed based on missing, inaccurate, or even conflicting knowledge regarding the patient situation [68], which may cause dangerous health situations to patients (e.g., undesirable reactions between medication and other drugs). Also to avoid serious situations or inappropriate medications that may lead to life-threatening reactions, the recommendation and aggregation of healthcare services cannot always be guided by experts from the healthcare domain, because patients have their own context and treatment restrictions, which makes the generation of a treatment plan for each of them, by experts or medical teams, a complex and burdening task. To reduce the risks on patients' health (drug allergies or contraindications), prediction and uncertainty handling techniques should be incorporated in the recommendation task. That will help prevent drug side effects, before proceeding to recommendation.

8.3. Data analytics for healthcare recommendation

Although the knowledge graph-based modeling of the healthcare network has covered the limitations of the traditional representation of user–item interactions (see Section 4), the healthcare recommendation task should be reinforced by additional analytical capabilities of domain-specific knowledge and patient-generated data. These latter (e.g., sequences of visits over time, usage history of medical treatments, sensors data) may include structured/unstructured and poorly annotated data and medical concepts (e.g., diagnosis, procedures, medical codes). Besides these characteristics, the high dimensionality, complexity, heterogeneity, and massiveness of healthcare data require advanced deep learning models to correctly learn their semantics (e.g., vector-based patient representations) and trigger various clinical tasks, such as medical text analysis, prediction tasks (e.g., diseases, mortality, length of stay, medical cost), patients sub-typing and clustering, cancer classification, etc. Most used deep learning models include RNN (LSTM, GRU), CNN, DAN, RBM, auto-encoders, GNN, CBOW, Skip-Gram, Deep Averaging Network, Transformer based models (e.g., convolutional transformer), meta-learning [69]. For example, patients' clinical events over time can be fed into Transformers and RNN models as an entire patient trajectory (i.e. events sequence) to predict the next clinical state of a patient, while meta-learning can be applied to deal with label/annotation insufficiency. Such clinical outcomes help understand patients states and construct complete user profiles, which will improve the recommendations accuracy.

8.4. Privacy-preserving healthcare recommendation

Healthcare information networks represent a typical scenario, in which user (e.g., patient) and item (e.g., medical treatment and services) data are distributed across multiple virtual hospitals (e.g., patient visits/treatments in different hospitals), and are explicitly represented using various data structures or privacy restrictions. Moreover, users' and items' features are partially made available to recommender systems, which leads to an incomplete view on users' preferences and quality of experience.

To cope with these issues, federated recommender systems [70] can be a promising solution to the privacy concerns caused by the distribution of user–item data, but also the heterogeneity of multiple sources of information. The combination of embedding models and federated learning [71] can be an elegant way to deal with healthcare service recommendation, while preserving users privacy. In fact, representation learning is employed to learn rich representations of entities (e.g., patients, services, IoT objects) in a federated setting, by encoding healthcare entities across multiple information network. In this case, federated learning will help collaboratively infer a unified profile of users and items, while taking into consideration the restrictions on the virtual hospitals' information networks. The federated embeddings, as well as the unified representations, will finally be fed into a federated recommendation algorithm to select the top-rated healthcare services.

9. Conclusion

Exhibiting the same behavior and principles of the service paradigm [1], healthcare systems' operations have recently been exposed as services. In this context, we have applied to principles of service reuse in order to aggregate smart healthcare services into treatment plans together with their required medical objects. This servicelization and encapsulation of healthcare operations is driven by patients' needs and current states, which are continuously monitored in a sensor cloud-based environment. To model the healthcare information network (e.g., healthcare services, symptoms, smart medical equipment, patient data, etc.), we exploited knowledge graphs as a powerful technology, allowing a multi-relational and semantic representation of healthcare entities. Then, we applied an incremental embedding method to learn the representation of the network content, and to identify the required healthcare service from a cuboid space of 3-C vectors. Finally, based on a repository of healthcare workflow templates, a recommendation module was implemented to select and combine the suitable services according to the patient's current state (input symptoms). Experimental results with real-world datasets have proven the advantages of treating healthcare data in a cuboid space, rather than parsing a complex graph structure, hence a better accuracy in service recommendations.

The present work will be extended by considering the uncertainty factors of the healthcare network. These include the fuzzy states or missing information of some patients (e.g., allergy cases), the incompleteness of services information (e.g., availability, usage restrictions), the behavioral deviations in smart medical objects (e.g., failures, performance), etc. Such uncertainty risks to affect the decision on the appropriate healthcare services. To solve these issues, probabilistic models [72,73] could be applied in the construction/update of the healthcare information network, as well as the learning of its cuboid structure. The representation of uncertain facts with confidence scores will provide an effective means of handling the uncertainty of patients', services' and smart objects' related knowledge, allowing a more accurate prediction of the recommended healthcare workflows.

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.

Data availability

The authors do not have permission to share data.

Acknowledgment

The authors express their gratitude to Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R408), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

References

[1] A. Bouguettaya, M. Singh, M. Huhns, Q.Z. Sheng, H. Dong, Q. Yu, A.G. Neiat, S. Mistry, B. Benatallah, B. Medjahed, et al., A service computing manifesto: the next 10 years, Commun. ACM 60 (4) (2017) 64–72.
[2] A. Alamri, W.S. Ansari, M.M. Hassan, M.S. Hossain, A. Alelaiwi, M.A. Hossain, A survey on sensor-cloud: architecture, applications, and approaches, Int. J. Distrib. Sens. Netw. 9 (2) (2013) 917923.
[3] X. Li, L. Da Xu, A review of internet of things—resource allocation, IEEE Internet Things J. (2020).
[4] K. Benouaret, P.C. Hung, L. Bellatreche, Service-enabled systems and applications: current and future trends, Serv. Orient. Comput. Appl. 15 (2021) 171–173.
[5] S. Ketu, P.K. Mishra, Internet of healthcare things: A contemporary survey, J. Netw. Comput. Appl. 192 (2021) 103179.
[6] X. Yu, D. Zhan, L. Liu, H. Lv, L. Xu, J. Du, A privacy-preserving cross-domain healthcare wearables recommendation algorithm based on domain-dependent and domain-independent feature fusion, IEEE J. Biomed. Health Inf. (2021).
[7] Y.-F. Huang, P. Liu, Q. Pan, J.-S. Lin, A doctor recommendation algorithm based on doctor performances and patient preferences, in: 2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP), IEEE, 2012, pp. 92–95.
[8] Y. Zhang, D. Zhang, M.M. Hassan, A. Alamri, L. Peng, CADRE: Cloud-assisted drug recommendation service for online pharmacies, Mob. Netw. Appl. 20 (3) (2015) 348–355.
[9] V. Subramaniyaswamy, G. Manogaran, R. Logesh, V. Vijayakumar, N. Chilamkurti, D. Malathi, N. Senthilselvan, An ontology-driven personalized food recommendation in IoT-based healthcare system, J. Supercomput. 75 (6) (2019) 3184–3216.
[10] S. Haussmann, O. Seneviratne, Y. Chen, Y. Ne'eman, J. Codella, C.-H. Chen, D.L. McGuinness, M.J. Zaki, Foodkg: a semantics-driven knowledge graph for food recommendation, in: International Semantic Web Conference, Springer, 2019, pp. 146–162.
[11] R. Chivukula, T.J. Lakshmi, S. Sumalatha, K.L.R. Reddy, Ontology based food recommendation, in: IOT with Smart Systems, Springer, 2022, pp. 751–759.
[12] L. Bellatreche, C. Ordonez, D. Méry, M. Golfarelli, et al., The central role of data repositories and data models in data science and advanced analytics, 2022.
[13] Y. Yang, J. Hu, Y. Liu, X. Chen, Doctor recommendation based on an intuitionistic normal cloud model considering patient preferences, Cognit. Comput. 12 (2) (2020) 460–478.
[14] Z.R. Rosenberg-Yunger, H. Thorsteinsdóttir, A.S. Daar, D.K. Martin, Stakeholder involvement in expensive drug recommendation decisions: an international perspective, Health Policy 105 (2–3) (2012) 226–235.
[15] S. Ji, S. Pan, E. Cambria, P. Marttinen, S.Y. Philip, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[16] D. Zhang, J. Yin, X. Zhu, C. Zhang, Network representation learning: A survey, IEEE Trans. Big Data 6 (1) (2018) 3–28.
[17] J. Ma, W. Gao, S. Joty, K.-F. Wong, An attention-based rumor detection model with tree-structured recursive neural networks, ACM Trans. Intell. Syst. Technol. 11 (4) (2020) 1–28.
[18] V. Di Massa, G. Monfardini, L. Sarti, F. Scarselli, M. Maggini, M. Gori, A comparison between recursive neural networks and graph neural networks, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 778–785.
[19] A.O. Afolabi, P. Toivanen, K. Haataja, J. Mykkänen, Systematic literature review on empirical results and practical implementations of healthcare recommender systems: lessons learned and a novel proposal, Int. J. Healthc. Inf. Syst. Inform. (IJHISI) 10 (4) (2015) 1–21.
[20] T.N.T. Tran, A. Felfernig, C. Trattner, A. Holzinger, Recommender systems in the healthcare domain: state-of-the-art and research issues, J. Intell. Inf. Syst. (2020) 1–31.
[21] X. Gao, F. Feng, X. He, H. Huang, X. Guan, C. Feng, Z. Ming, T.-S. Chua, Hierarchical attention network for visually-aware food recommendation, IEEE Trans. Multimed. 22 (6) (2019) 1647–1659.
[22] W. Min, S. Jiang, R. Jain, Food recommendation: Framework, existing solutions, and challenges, IEEE Trans. Multimed. 22 (10) (2019) 2659–2671.
[23] W. Wang, L.-Y. Duan, H. Jiang, P. Jing, X. Song, L. Nie, Market2Dish: Health-aware food recommendation, ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) 17 (1) (2021) 1–19.
[24] X. Gao, F. Feng, H. Huang, X.-L. Mao, T. Lan, Z. Chi, Food recommendation with graph convolutional network, Inform. Sci. 584 (2022) 170–183.
[25] K. Shimada, H. Takada, S. Mitsuyama, H. Ban, H. Matsuo, H. Otake, H. Kunishima, K. Kanemitsu, M. Kaku, Drug-recommendation system for patients with infectious diseases, in: AMIA Annual Symposium Proceedings, Vol. 2005, American Medical Informatics Association, 2005, p. 1112.
[26] H. Liu, H. Lin, C. Shen, L. Yang, Y. Lin, B. Xu, Z. Yang, J. Wang, Y. Sun, A network representation approach for COVID-19 drug recommendation, Methods (2021).

[27] N.K. Kumar, D. Vigneswari, A drug recommendation system for multi-disease in health care using machine learning, in: Advances in Communication and Computational Technology, Springer, 2021, pp. 1–12.
[28] H.F. Badawi, H. Dong, A. El Saddik, Mobile cloud-based physical activity advisory system using biofeedback sensors, Future Gener. Comput. Syst. 66 (2017) 59–70.
[29] A. Gupta, S.K. Dubey, Fuzzy nutrition recommendation system for diabetic patients, in: Intelligent Communication, Control and Devices, Springer, 2018, pp. 1381–1390.
[30] K.-c. Choi, H.Y. Chan, D.N. Chan, W.W. Lam, C.W. Chan, S.S. Ho, K.K. Cheng, W.B. Goggins, A.T. Shiu, W.K. So, The mediating role of health professionals' recommendation in the uptake of colorectal cancer testing among older c hinese adults, Int. J. Nurs. Pract. 20 (2) (2014) 170–178.
[31] A.S. Axe, R. Varghese, M. Bosma, N. Kitson, D.J. Bradshaw, Dental health professional recommendation and consumer habits in denture cleansing, J. Prosthet. Dent. 115 (2) (2016) 183–188.
[32] S. Mondal, A. Basu, N. Mukherjee, Building a trust-based doctor recommendation system on top of multilayer graph database, J. Biomed. Inform. 110 (2020) 103549.
[33] Y. Yan, G. Yu, X. Yan, Online doctor recommendation with convolutional neural network and sparse inputs, Comput. Intell. Neurosci. 2020 (2020).
[34] Q. Meng, H. Xiong, A doctor recommendation based on graph computing and LDA topic model, Int. J. Comput. Intell. Syst. 14 (1) (2021) 808–817.
[35] L.R. Nair, S.D. Shetty, S.D. Shetty, Applying spark based machine learning model on streaming big data for health status prediction, Comput. Electr. Eng. 65 (2018) 393–399.
[36] A. Akbulut, E. Ertugrul, V. Topcu, Fetal health status prediction based on maternal clinical history using machine learning techniques, Comput. Methods Programs Biomed. 163 (2018) 87–100.
[37] B. Abu-Salih, M. Al-Qurishi, M. Alweshah, M. Al-Smadi, R. Alfayez, H. Saadeh, Healthcare knowledge graph construction: A systematic review of the state-of-the-art, open issues, and opportunities, J. Big Data 10 (1) (2023) 81.
[38] M. Postiglione, Towards an Italian healthcare knowledge graph, in: International Conference on Similarity Search and Applications, Springer, 2021, pp. 387–394.
[39] W. Liu, L. Yin, C. Wang, F. Liu, Z. Ni, Medical knowledge graph in Chinese using deep semantic mobile computation based on IoT and WoT, Wirel. Commun. Mob. Comput. 2021 (2021).
[40] P. Ernst, C. Meng, A. Siu, G. Weikum, Knowlife: a knowledge graph for health and life sciences, in: 2014 IEEE 30th International Conference on Data Engineering, IEEE, 2014, pp. 1254–1257.
[41] M.R. Kamdar, W. Dowling, M. Carroll, C. Fitzgerald, S. Pal, S. Ross, K. Scranton, D. Henke, M. Samarasinghe, A healthcare knowledge graph-based approach to enable focused clinical search, 2021.
[42] M. Kamdar, M. Carroll, W. Dowling, L. Wogulis, C. Fitzgerald, M. Corkum, D. Walsh, D. Conrad, C. Stanley, S. Ross, et al., Focused clinical search through query intent interpretation and a healthcare knowledge graph, 2020, Available at SSRN 3775468.
[43] J. Park, Y. Cho, H. Lee, J. Choo, E. Choi, Knowledge graph-based question answering with electronic health records, in: Machine Learning for Healthcare Conference, PMLR, 2021, pp. 36–53.
[44] H.Q. Yu, Dynamic causality knowledge graph generation for supporting the chatbot healthcare system, in: Proceedings of the Future Technologies Conference, Springer, 2020, pp. 30–45.
[45] L. Huang, C. Yu, Y. Chi, X. Qi, H. Xu, Towards smart healthcare management based on knowledge graph technology, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019, pp. 330–337.
[46] K.M. Malik, M. Krishnamurthy, M. Alobaidi, M. Hussain, F. Alam, G. Malik, Automated domain-specific healthcare knowledge graph curation framework: Subarachnoid hemorrhage as phenotype, Expert Syst. Appl. 145 (2020) 113120.
[47] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020).
[48] H. Yuan, W. Deng, Doctor recommendation on healthcare consultation platforms: an integrated framework of knowledge graph and deep learning, Internet Res. (2021).
[49] W. Liu, L. Yin, C. Wang, F. Liu, Z. Ni, Multitask healthcare management recommendation system leveraging knowledge graph, J. Healthc. Eng. 2021 (2021).
[50] H. Mezni, D. Benslimane, L. Bellatreche, Context-aware service recommendation based on knowledge graph embedding, IEEE Trans. Knowl. Data Eng. 34 (11) (2021) 5225–5238.
[51] Y. Zhang, J. Wu, H. Wang, Neural binary representation learning for large-scale collaborative filtering, IEEE Access 7 (2019) 60752–60763.
[52] X. Deng, F. Huangfu, Collaborative variational deep learning for healthcare recommendation, IEEE Access 7 (2019) 55679–55688.
[53] W. Wang, T. Tang, F. Xia, Z. Gong, Z. Chen, H. Liu, Collaborative filtering with network representation learning for citation recommendation, IEEE Trans. Big Data 8 (5) (2020) 1233–1246.
[54] B. Swaminathan, S. Palani, S. Vairavasundaram, Feature fusion based deep neural collaborative filtering model for fertilizer prediction, Expert Syst. Appl. 216 (2023) 119441.
[55] Y. Tay, L. Anh Tuan, S.C. Hui, Latent relational metric learning via memory-based attention for collaborative ranking, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 729–739.
[56] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 305–314.
[57] X. Deng, F. Zhuang, Z. Zhu, Neural variational collaborative filtering with side information for top-k recommendation, Int. J. Mach. Learn. Cybern. 10 (2019) 3273–3284.
[58] E. Elahi, Z. Halim, Graph attention-based collaborative filtering for user-specific recommender system using knowledge graph and deep neural networks, Knowl. Inf. Syst. 64 (9) (2022) 2457–2480.
[59] H.-X. Hu, Z.-W. Jiang, Y.-F. Zhao, Y. Zhang, H. Wang, W. Wang, Network representation learning-enhanced multisource information fusion model for POI recommendation in smart city, IEEE Internet Things J. 8 (12) (2020) 9539–9548.
[60] B. Tang, M. Yan, N. Zhang, L. Xu, X. Zhang, H. Ren, Co-attentive representation learning for web services classification, Expert Syst. Appl. 180 (2021) 115070.
[61] Y. Li, Z. Zhu, X. Guo, S. Li, Y. Yang, Y. Zhao, HGV4risk: Hierarchical global view-guided sequence representation learning for risk prediction, ACM Trans. Knowl. Discov. Data (2022).
[62] X. Zhang, Q. Wang, Z. Ye, H. Ying, D. Yu, Federated representation learning with data heterogeneity for human mobility prediction, IEEE Trans. Intell. Transp. Syst. (2023).
[63] X. Li, X. Niu, G. Liu, Spatiotemporal representation learning for rescue route selection: An optimized regularization based method, Electron. Commer. Res. Appl. 48 (2021) 101065.
[64] H. Peng, R. Yang, Z. Wang, J. Li, L. He, P. Yu, A. Zomaya, R. Ranjan, Lime: Low-cost incremental learning for dynamic heterogeneous information networks, IEEE Trans. Comput. (2021).
[65] W. Cook, A. Rohe, Computing minimum-weight perfect matchings, INFORMS J. Comput. 11 (2) (1999) 138–148.
[66] D.E. Drake, S. Hougardy, Linear time local improvements for weighted matchings in graphs, in: International Workshop on Experimental and Efficient Algorithms, Springer, 2003, pp. 107–119.

[67] P. Lammich, S.R. Sefidgar, Formalizing the edmonds-karp algorithm, in: Interactive Theorem Proving: 7th International Conference, ITP 2016, Nancy, France, August 22-25, 2016, Proceedings 7, Springer, 2016, pp. 219–234.
[68] M. Kuanr, P. Mohapatra, Outranking relations based multi-criteria recommender system for analysis of health risk using multi-objective feature selection approach, Data Knowl. Eng. 145 (2023) 102144.
[69] Y. Si, J. Du, Z. Li, X. Jiang, T. Miller, F. Wang, W.J. Zheng, K. Roberts, Deep representation learning of patient data from electronic health records (EHR): A systematic review, J. Biomed. Inform. 115 (2021) 103671.
[70] Z. Alamgir, F.K. Khan, S. Karim, Federated recommenders: methods, challenges and future, Cluster Comput. 25 (6) (2022) 4075–4096.
[71] R.S. Antunes, C. André da Costa, A. Küderle, I.A. Yari, B. Eskofier, Federated learning for healthcare: Systematic review and architecture proposal, ACM Trans. Intell. Syst. Technol. 13 (4) (2022) 1–23.
[72] D. Kim, L. Xie, C.S. Ong, Probabilistic knowledge graph construction: Compositional and incremental approaches, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 2257–2262.
[73] X. Chen, M. Chen, W. Shi, Y. Sun, C. Zaniolo, Embedding uncertain knowledge graphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 3363–3370.

Mouhamed Gaith Ayadi received his Ph.D. degree in computer sciences from the Higher Institute of Management in Tunis (Tunisia). He is currently an assistant professor with the University of Jendouba, Tunisia and a member of BESTMOD research laboratory. His research interests mainly include information retrieval and data mining and analytics, with application to healthcare systems.

Haithem Mezni received the Ph.D. degree in computer science from Manouba University in 2014. He is currently an associate professor with Jendouba University and a member of SMART Research Laboratory, Tunisia. His current research interests include Service lifecycle management, Cloud computing and Big data. Haithem Mezni has also supervised several research works on Web service adaptation, Cloud service provisioning and recommendation, Cloud resource management, service composition and process fragments reuse, Big data services, etc. His publication record includes articles in peer-reviewed journals such as IEEE TKDE, TSC, DKE, FGCS, SoCo, JSS, IJAR, AIHC, JNCA, etc. Haithem Mezni has also served as a guest editor in J. Approximate Reasoning and has co-organized some international conferences and workshops.

Rana Alnashwan received a Ph.D. degree in computer science from University College Cork, Cork, Ireland. She is an assistant professor at the Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia. Her research interests include Artificial Intelligence, Data Science and Analytics.

Hela Elmannai received her Ph.D. degree in Information Technology from SUPCOM in Tunisia. She is currently an assistant professor at the department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia. Her research interests include artificial intelligence, networking, blockchain and engineering applications.