DLRS: Deep Learning-based Recommender System for Smart Healthcare Ecosystem

Gagangeet Singh Aujla*, *Member, IEEE,* Anish Jindal†, *Member, IEEE,* Rajat Chaudhary‡, *Student Member, IEEE,*
Neeraj Kumar§, *Senior Member, IEEE,* Sahil Vashist‖, *Student Member, IEEE,*
Neeraj Sharma¶, and Mohammad S. Obaidat**, *Fellow of IEEE and Fellow of SCS*

* ¶Computer Science & Engineering Department, Chandigarh University, Gharuan (Punjab), India
‡§‖ Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala (Punjab), India
†School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, UK
¶ ECE Department, Nazarbayev University, Astana, Kazakhstan, King Abdullah II School of Information Technology,
University of Jordan, Jordan, and University of Science and Technology, Beijing, China
(e-mail: gagi_aujla82@yahoo.com, anishjindal90@gmail.com, rajatlibran@gmail.com, neeraj.kumar@thapar.edu,
sahil.vashist@thapar.edu, neerajkirti@gmail.com, msobaidat@gmail.com, m.s.obaidat@ieee.org).

Abstract—**Nowadays, the conventional healthcare domain has witnessed a paradigm shift towards patient-driven healthcare 4.0 ecosystem. In this direction, healthcare recommender systems provide ubiquitous healthcare services to the end users even on the move. However, there are various challenges for the design of patient driven healthcare recommender systems. Some of the major challenges are: a) handling huge amount of data generated by smart devices and sensors, b) dynamic network management for real-time data transmission, and c) lack of knowledge gathering and aggregation methods. For these reasons, in this paper; *DLRS*: *A Deep Learning based Recommender System using software defined networking* (SDN) is designed for smart healthcare ecosystem. *DLSR* works in the following phases: a) a tensor-based dimensionality reduction algorithm is proposed for removing unwanted dimensions in the acquired data, b) a decision tree-based classification scheme is presented for categorization of the patient queries on the basis of different diseases, and c) a convolutional neural network based system is designed for providing recommendations about the patient health. On evaluation, the results obtained prove the superiority of the proposed scheme in contrast to existing competing schemes.**

Index Terms—**Deep Learning, Dimensionality Reduction, Healthcare, Recommender Systems, Software-defined Networks.**

I. INTRODUCTION

With the migration of rural population to urban cities, an optimized usage of existing infrastructure paved the way for the design of various applications in the smart cities. One of such application is smart health care, which is essential for the mankind survival. The basic principle behind the smart healthcare ecosystem is the usage of smart devices, sensor and IoT to provide improved health care services at low cost to the smart city residents readily at any location and time [1]. Since the doctor-patient ratio in developing countries is very poor, so, smart healthcare can play a vital role for monitoring patient health from a remote location through body sensors, which can transmit the data to the cloud, which doctors can access and provide prescriptions [2]. These sensors are easily available in market, which are implantable or wearable and can be used for remote health monitoring. Some of the IoT based healthcare devices are based on Arduino and Raspberry Pi.

Nowadays, the healthcare systems transformed to a more effective patient driven platform known as Healthcare 4.0. This platform is patient driven concept, which enables personalization of patients' health through recommendations using cloud computing and IoT [3]. However, the design of Healthcare 4.0 ecosystem has to face various challenges [4] such as interoperability, decentralization, real-time capability, service orientation and many more. any authors have proposed recommender systems for providing healthcare solutions to the patients. For instance, Duan *et al.* [5] proposed a recommender system for nursing care applications to provide clinical decision support and nursing education to boost the performance of existing practices. Thong *et al.* [6] used fuzzy logic to create a recommender system for medical diagnosis to manage the uncertain behavior in clinical experiences of the users. In [7], the authors proposed a framework for personalized clinical prescription on the basis of information available in the previous medical records of the users. The authors combined artificial neural networks with case-based reasoning to improve the efficiency of recommender systems and to reduce the prediction errors. In addition to it, the use of clustering algorithms in healthcare recommender systems has been studied in [8].

However, the major challenges in these recommender systems still remain the same, i.e., data analytics and data management. The data in healthcare services is gathered at a rapid pace which requires a specialized process to handle. So, a data reduction technique needs to be incorporated in the existing recommender system which considers only the important data. Moreover, the advancements in machine learning and related techniques can be used by the healthcare recommendation systems to learn the hidden data patterns. Moreover, the underlying network resources can be optimized to gather and process the data at a rapid pace which results in minimum additional delay. Considering all these aspects, this paper proposes a software defined network (SDN)-enabled and deep leaning based healthcare recommender system for disease classification on the basis of gathered data for recommending the doctor about the treatment of a disease.

A. Contributions

After analysis of the above facts, *DLRS*: A Deep Learning based Recommender System using SDN is designed, which works in following phases.

- A Software defined healthcare ecosystem model comprising of three decoupled planes is designed for seamless transfer of healthcare data.
- A tensor-based dimensionality reduction is used to remove unwanted dimensions from heathcare data.
- A decision tree-based classification is used to categorize the patient queries based on different diseases.
- A convolutional neural network based system is designed for providing recommendations about the patient health.
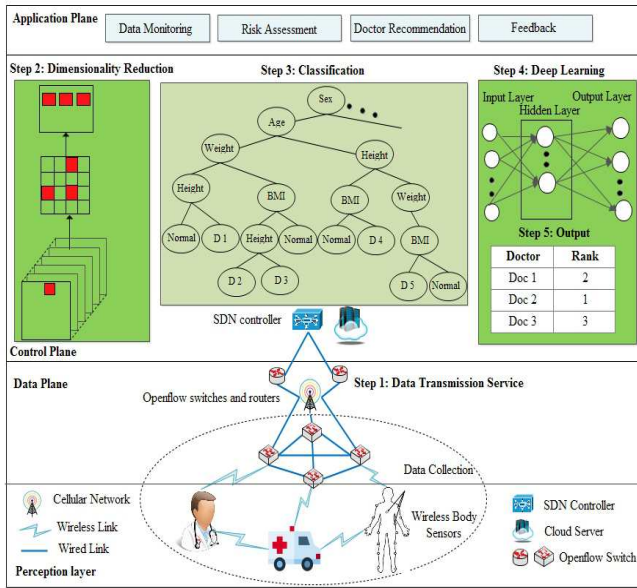

Fig. 1. SDN-based Healthcare scenario

II. SOFTWARE DEFINED HEALTHCARE ECOSYSTEM

Fig. 2 shows an SDN-enabled system model of a smart healthcare network. The model is divided into three planes: data, control, and application planes. The users are located at the perception layer below the data plane. The planes of the proposed architecture are discussed as below.

- *Data Plane:* It is also known as infrastructure layer, which consists of edge switches and the access points (cellular network). The first step is the data transmission service where patients data is collected with the help of wireless body sensors and is transmitted via a wireless link through secure channel to the cloud server [9].
- *Control Plane:* The network controller software resides at the control plane and intelligently handles the traffic of the cloud server [10]. The control logic executed by the network controller is initiated by step 2 i.e. the dimensionality reduction process. With the help of this process, the high dimensional tensor data is reduced to a smaller size data by extracting only the relevant features. Step 3 involves the data classification with the help of decision tree. Based on the input parameters of the patient, a decision tree is formed. The root node of

the tree is the most influencing feature of the patient. The patient sex is the root node of the tree so based on this input information, the output is stored in the left and right child node of the tree. Now, the same pattern is followed like the symptoms of the patient disease, which is classified and finally stored as the different classes pattern information at the leaf node. The next step is the deep learning technique in order to recommend the doctors for a particular disease classification.

- *Application Plane:* At the application plane, the multiple applications are running like data monitoring, risk assessment, doctor recommendation, and feedback. The last step is the output returned by the deep learning approach. The output is the list of doctors with their ranking. The patient have multiple choices for selecting the nearby or remote doctors based on their rankings. The final output is the feedback session to which doctor did the patient choose based on the feedback, the ranking gets updated.
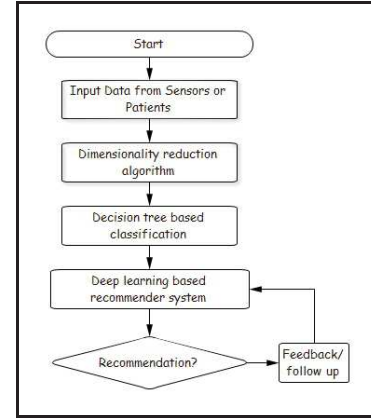

Fig. 2. Recommender system

III. *DLRS*: THE PROPOSED FRAMEWORK

The proposed framework (*DLRS*) works in three phases, which are described in the subsequent subsections. Moreover, the flow of work in *DLRS* is depicted in Fig. 2.

A. Phase I: Dimensionality Reduction Scheme

The data generated by smart devices (D^{gen}) and patients (D^{acq}) is represented in tensor form, i.e., as a multi-dimensional arrays. Generally, tensors are used to represent the data with higher dimensions and different characteristics. The representation of an *n*-order tensor T^n is shown below.

$$T^n \in R^{o_1 \times o_2 \times o_3 \dots \times o_n} \tag{1}$$

where, o_1, o_2,.., o_n represents the orders of a tensor, which describe the dimensionality of data characteristics.

The healthcare data is represented in tensorized form as a cross-product of *n* number of characteristics with multiple dimensions, which is depicted as below.

$$E[a_1 \otimes a_2 \otimes a_3 \otimes \dots a_n] = R^{o_1 \times a_2 \times o_3 \dots \times o_n} \tag{2}$$

where, $a_1, a_2, \dots, a_n$ denote different attributes of data (such as, symptoms, previous disease history, etc), which are independent of each other.

Now, D^{gen} and D^{acq} are converted into tensor form using Eqs. (1 and 2), and represented as shown below.

$$D^{gen} \to T^{gen}, D^{acq} \to T^{acq} \qquad (3)$$

where, T^{gen} and T^{acq} represents sub-tensors for generated and acquired data, respectively.

After converting the data into tensor form, the next step is to remove the redundancy in data. So, the sub-tensors are combined into a unified tensor (T^{un}) using *unified data tensorization operation* [11] and T^{un} is represented as below.

$$f : (D^{gen} \cup D^{acq}) \to T^{gen} \cup T^{acq} \qquad (4)$$

$$f(p,q) = v$$

where, $p \in T^{gen}$, $q \in T^{acq}$ and $u \in T^{un}$

Now, all the similar characteristics of tensors are combined, thereby removing all the redundant data values. But, the complexity of data is still high due to higher dimensionality. The high complexity can end up in inconsistency in healthcare recommendations. Therefore, to handle this problem, lower-order tensor representation could be beneficial. Hence, T^{un} is converted into lower-order tensors having limited dimensions, which are essential for healthcare recommendations. An nth order tensor can be reduced into *n* matrices using a *tensor unfolding or matricization operation* provided in [11]. For any tensor (T^n), the dimensions of unfolded *n*-order matrix into a mode-*i* matrix is shown below.

$$T^n \in R^{o_i \times (o_1 \times o_2 \times o_3 ... \times o_{i-1} \times o_{i+1} .... \times o_n)} \qquad (5)$$

s.t.

$$o_i, 1 \le i \le n$$

$$\prod_{j=1}^{n} o_j, i \ne j$$

where, o_i and o_j represent the number of rows and columns of each mode-*i* matrix, respectively.

After performing *matrix unfolding operation* on T^n, an unfolded mode-*i* matrix (M_i) is decomposed using singular value decomposition (SVD). The representation of M_i is shown below.

$$M_i = \mathring{U}_i \mathring{S}_i \mathring{V}_i^* \qquad (6)$$

where, $\mathring{U}_i$, $\mathring{S}_i$ and $\mathring{V}_i^*$ denote an orthogonal unitary matrix, a diagonal matrix and the conjugate transpose of orthogonal unitary matrix $\mathring{V}_i$, respectively.

In Eq. 6, $\mathring{S}_i$ with non-negative entries, contains *i* singular values represented as σ_i, which are shown below.

$$\mathring{S}_i = \begin{bmatrix} \sigma_1 & \cdots & \cdots & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \sigma_i & 0 \end{bmatrix}. \qquad (7)$$

s.t.

$$\sigma_1 \ge \sigma_2 \ge ...... \ge \sigma_i > 0$$

Following the above SVD process, the rank of $\mathring{S}_i$ of each mode-*i* matrix is approximated to a lower rank (r; s.t.

$r \le n$). This SVD process is applied on each mode-*i* matrix incrementally to achieve low rank approximation. Now, the orthogonal vectors of the truncated orthogonal unitary matrix are projected on the initial tensor (T^{un}) to obtain a reduced tensor (T^{rd}). An *n*-mode product operation is applied by $\mathring{U}_i$ to reduce T^{un}. The entire operation is shown as follows.

$$(T^{un} \times_n \mathring{U}_i)_{\epsilon_1 \epsilon_2 .... \epsilon_{k-1} \epsilon_k \epsilon_{k+1} .... \epsilon_n} \qquad (8)$$

where, $\epsilon_1, \epsilon_2 .... \epsilon_{k-1}, \epsilon_k, \epsilon_{k+1}, ... \epsilon_n$ are the dimensional attributes of T^n.

Now, Eq. 8 could be represent as shown below.

$$T^{rd} = T^{un} \times_{x=1}^{n} \mathring{U}_n^T \qquad (9)$$

$$T^{rd} = T \times_1 \mathring{U}_2^T \times_2 \mathring{U}_3^T ..... \times_n \mathring{U}_n^T \qquad (10)$$

Finally, an approximated tensor (T^{app}) can be reconstructed from T^{rd}, as and when required. The reconstruction process is shown as below.

$$T^{app} = T^{rd} \times_1 \mathring{U}_1 \times_2 \mathring{U}_2 \times_3 \mathring{U}_3 .... \times_n \mathring{U}_n \qquad (11)$$

Algorithm 1 SVD-based dimensionality reduction algorithm

Input: D^{gen} and D^{acq}
Output: T^{un}, T^{rd} and T^{app}

1: Input D^{gen} and D_{ac}
2: $D^{gen} \to T^{gen}$, $D^{acq} \to T^{acq}$
3: *Unified tensorization operation* $\to T^{gen}$ and T^{acq}
4: *Unfolding operation* $\to T^{un}$
5: **for** (i=1; i $\le$ n; i++) **do**
6:     $SVD(M_i) \to \mathring{U}_i$, $\mathring{S}_i$ and $\mathring{V}_i^*$
7:     Extract $\sigma_i \leftarrow \mathring{S}_i$
8:     $Rank(\mathring{S}_i)$ = n
9:     **while do**$(1 < r \le n)$
10:         Get $(\mathring{S})$ by pruning the smallest σ_i
11:         $Rank(\mathring{S})$
12:         *Reconstruct* (M_i) using Eq. (6).
13:         *Extract* $\mathring{U}_i, \mathring{S}_i, and \mathring{V}_i^*$
14:     **end while**
15:     *Store* $\to \mathring{U}_i$.
16:     Perform *n-mode product operation* of $\mathring{U}_i$ on T^{un}.
17:     Get T^{rd} using Eq.(10).
18: **end for**
19: Reconstruct T^{app} using Eq.(11).

The above described dimensionality reduction process has been presented in the form of Algorithm 1, which is elaborated as ahead. The data acquired from patients or generated by sensors, i.e., D^{gen} and D^{acq} is converted into tensorized form (T^{gen} and T^{acq}) using Eq. 1. Now, *unified tensorization operation* is applied on T^{gen} and T^{acq}. Here, the output received is a unified tensor, i.e, T^{un} (line 1-3). After this, *unfolding operation* is applied on T^{un}. Now, the unfolded tensor is decomposed using SVD process and three matrices ($\mathring{U}_i$, $\mathring{S}_i$ and $\mathring{V}_i^*$) are obtained using Eq. 6. From diagonal matrix ($\mathring{S}_i$), σ_i is extracted. After this, the rank of $\mathring{S}_i$ is computed (line 4-8). Now, the smallest σ_i is pruned to get new $\mathring{S}_i$ until the new rank r, which is lower than n is obtained. After this process, M_i is reconstructed using Eq. 6. The complete process is repeated till all the new matrices with lower rank are obtained (line 9-14). Once this process is over, $\mathring{U}_i$ is stored and *n-mode product operation* of $\mathring{U}_i$ is projected over T^{un}. Finally, T^{rd} is generated using Eq. 10 (line 15-17). Whenever required, T^{app} can be reconstructed using Eq. 11 (line 18).

Algorithm 2 Disease Specific Classification Scheme

Input: D: Training dataset, k: Disease attributes, A(k): all attributes of k.
Output: Decision_tree($\tau\{\}$).

```
 1: procedure FUNCTION
 2:     root[Tree] = Null;
 3:     a = 0;
 4:     extract A(x);
 5:     if (δ== Null || tuples in δ belongs to same class) then
 6:         return ι as a leaf node belonging to same class;
 7:     end if
 8:     while (A(k) > 1) do
 9:         Compute ε(D);
10:         for (℘ = 1; ℘ ≤ A(k); ℘++) do
11:             Compute ε(D|A), G̃(D, A), G̃_rat(D, A);
12:             if (G̃_rat[D, A] > a) then
13:                 a ← G̃_rat[D, A];
14:                 R{mx} = ℘;
15:             end if
16:         end for
17:         R{mx} = Max(G̃_rat(D, A));
18:         Build R{mx} as parent node;
19:         S_sub ← split S into k decision attributes;
20:         for (℘ = 1; ℘ ≤ k; ℘++) do
21:             τ_k ← (S_sub − R{max})
22:             Join τ_k with the corresponding τ
23:         end for
24:     end while
25:     return τ{};
26: end procedure
```

IV. PHASE II: DISEASE SPECIFIC CLASSIFICATION

The input data or patient query can relate to different diseases, which should be classified into various disease specific categories for improving the accuracy of the recommendations. In this direction, a decision tree based disease specific classification scheme for incoming healthcare data is designed, which classifies the data into different queues on the basis of different diseases. For example, all the data and queries related to heart disease are added to a single queue and handled separately by the recommender system. The incoming healthcare queries are classified according to different diseases from 1 to k classes defined by a decision tree. In the first step, the disease specific decision tree is trained with respect to different disease features. Now, the trained decision tree is implemented over an SDN controller located at the control plane. The SDN controller is responsible to identify the different disease classes. For simplicity, it is considered that the trained healthcare data D consists of k types of diseases. Accordingly, the entropy of D is described as follows.

$$\varepsilon(D) = -\sum_{i=1}^{k} \wp_i log_2 \wp_i \qquad (12)$$

where, $\wp_1$, $\wp_2$,..., $\wp_k$ represent the probability of each disease listed in D.

The uncertainty of information depends directly on the value of entropy. Therefore, a lower value of entropy means a lower uncertainty in the information. Now, D is distributed into m subsets based on the attribute A. Hence, the expected entropy of x for D is computed as below.

$$\varepsilon(D|A) = -\sum_{i=1}^{k} \wp(D_i)\varepsilon(D_i) \qquad (13)$$

Now, the information gain is defined on the basis of the original entropy and the entropy, which is shown as below.

$$\tilde{G}(D, A) = \varepsilon(D) - \varepsilon(D|A) \qquad (14)$$

After this, the value of $\tilde{G}(D, A)$ is used to calculate the information gain ratio, which is given as below.

$$\tilde{G}_{rat}(D, A) = \frac{\tilde{G}(D, A)}{\delta(D, A)} \qquad (15)$$

where, $\delta(D, A)$ represents the split information of A D, which is defined as below.

$$\delta = -\sum_{i=1}^{j} \wp(D_i)log_2 \wp(D_i) \qquad (16)$$

On the basis of above defined aspects, the disease specific classification Algorithm II is designed.

V. CNN MODEL FOR HEALTHCARE

The CNN is a deep learning counterpart of ANN in which each layer learns from the input neurons (rather than just adjusting the weights as the case with ANN). Due to their ability to learn hidden features, CNN's have already been used extensively for wide variety of applications such as image classification, traffic prediction, etc. [12], [13]. The basic architecture of CNN model for healthcare recommender system consists of an input layer, an output layer, and multiple hidden layers in-between. The hidden layers in the CNN model further consists of various rounds of convolutional and pooling layers which are finally connected to the fully connected layers before giving an output. As the healthcare data is in the form of text and CNN takes vector as an input, therefore, the first step in CNN processing is to convert the textual data into vector form. For this purpose, natural language processing is applied to the healthcare data to represent it in R^d dimensional vector, where the value of d is taken as 100 [14]. Once the input vector is created for the initial layer of CNN, this layer passes over the information to the hidden layers in the group of words. The advantage of grouping the words together is that they are locally connected to the hidden layer neurons for a particular region. This means that the traditional weight parameters used in the artificial neural networks for different locations are overlapped; thereby, these neurons share the same amount of information with much less complexity and connectivity.

In the convolution hidden layer, a mask (also known as function or kernel) is applied on the group of input words (also known as receptive field). It should be noted here that the size of the mask must be same as the size of the receptive field. So, if we choose 5 words as the size of the group, then the row vector will consist of $5 \times 100 = 500$ vector values. When this mask is applied to the receptive field, a weighted output of the complete receptive field is computed and is given as the output of the neuron which is later carried over to the pooling hidden layer. The mask is moved over the input layer one step at a time to cover all the values in the input vector space. So, if 100 convolution filters are to be applied on the input layer, the mask M would lie in $R^{100 \times 500}$ where the size of each receptive field in 500. For each receptive field, a weighted output is computed

by hidden neuron and this neuron also learns an overall bias for the respective receptive field. The output of this hidden neuron at $(a,b)^{th}$ position once the convolution operation is completed can be mathematically represented as:

$$O_{ab}^1 = \sigma \left(B + \sum_{i=0}^{n} \sum_{j=0}^{n} w_{(i,j)} \alpha_{(a+i,b+j)} \right) \quad (17)$$

where, σ is the activation function, B is the value of bias, $w_{(i,j)}$ is the weighted mask of dimension $n \times n$, and $\alpha_{(a,b)}$ represents the input activation at $(a,b)^{th}$ position.

There are various types of activation function which are used in this layer such as *tanh* or the sigmoid function which are mathematically represented as:

$$\sigma = \tanh(x) \quad (18)$$
$$\sigma = 1/(1 + exp(-x)) \quad (19)$$

where, x is an input to the neuron.

However, other functions also exist which have been proved very useful in the past for the deep networks; one of which is the rectified linear unit (ReLU) which is given by:

$$\sigma = max(0, a) \quad (20)$$

This function is the popular choice in deep learning architectures because it is very easy to implement and the training time with gradient descent is much less in this as compared to the *tanh* or sigmoid functions. Thus, ReLU is used as the activation function in the convolution layer to extract an output from the input layer. The output of this layer is also called as a feature map. Once the convolution operation is completed, the output is passed to the hidden pooling layer (which is immediate to the convolution layer) to further simplify the output. In general, it creates a condensed feature vector of different feature maps resulting from the convolution layer. To perform the task of pooling, max-pooling is used in the proposed scheme. In max-pooling, the maximum value of the complete region is given as an output. The output of 1-max pooling can be mathematically depicted as:

$$O_b^2 = \max_{1 \leq a \leq n} O_{ab}^1 \quad (21)$$

The advantage of max-pooling is that sometimes the role of each word in the data is not equal and max-pooling helps in choosing the words that play an important role.Another advantage of performing the pooling operation is that it makes the feature map less prone to small input variations and reduces number of parameters that are required in the later stages. Then, the convolution and pooling layers are repeated in many stages and in the last stage, the complete network becomes fully connected. Thus, at the last stage, every neuron at the end of pooling layer is connected to every neuron in the output layer. This is mathematically represented as:

$$O^3 = w^3 O^2 + B^3 \quad (22)$$

where, w^3 is the corresponding weight vector of the neurons and B^3 is the bias.

Once the CNN model is created using the training dataset, it is able to learn the hidden features in the data and is able to provide the recommendations based on the learned features.

VI. RESULTS AND DISCUSSIONS

A case study for healthcare system comprising of 100 patients, 100 sensors, 20 doctors is considered to evaluate the proposed scheme. The set of 20 doctors consists of 4 general physicians, 3 neurosurgeons, 3 heart specialists, 2 skin specialist, 2 dermatologist, 2 pediatricians, 2 gynecologists and 2 orthopedicians, which are geo-located in a smart city environment. The network topology using SDN (Openflow protocol) is designed in Mininet emulator [15]. The smart sensors are deployed to monitor heart-beat rate, body temperature, oxygen levels, and blood pressure. The evaluation parameters considered for evaluation purposes are discussed as below.

- *Dimensionality reduction ratio* (φ) is defined as the ratio of non-zero value ratio between T^{rd} and $\mathring{U}_i$ to the non-zero values of the T^{un}, which is shown as below.

$$\varphi = \frac{nz(T^{rd}) + \sum_{i=1}^{n} nz(\mathring{U}_i)}{nz(T^{un})} \quad (23)$$

- *Network latency* ($\mathfrak{t}$) is the delay experienced for the transmission of healthcare data from patient or sensor to the SDN controller, where recommender systems operates. $\mathfrak{t}$ is defined as below.

$$\mathfrak{t} = \mathfrak{t}_{pr} + \mathfrak{t}_q + \mathfrak{t}_t + \mathfrak{t}_{pg} + \jmath \quad (24)$$

where, $\mathfrak{t}_{pr}, \mathfrak{t}_q, \mathfrak{t}_t, \mathfrak{t}_{pg}$ and $\jmath$ represents processing, queuing, transmission, propagation delays and jitter.

- *Root mean squared error (RMSE)*: It is defined as below.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (D_{pr} - D_{ac})^2} \quad (25)$$

where, D_{pr} and D_{ac} are the predicted and actual classes of diseases and n is the number of predictions.

- *Mean absolute percentage error (MAPE)*: It is defined as:

$$MAPE = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{D_{pr} - D_{ac}}{D_{pr}} \right| \times 100 \quad (26)$$

A. Evaluation Results

The data acquired from patients and sensors is converted into sub-tensors, which are combined to form a T^{un} by the application of *unified data tensorization operation*. Now, SVD scheme is applied on the T^{un} followed by extraction of singular values, which are pruned to obtain T^{rd}. This reduced tensor is an approximate representation of T^{un} having all the essential dimensions of healthcare data. After evaluation of the case study, it is evident that the proposed framework achieves higher approximation ratio in contrast to principal component analysis algorithm. Fig. 3 shows the approximation ratio achieved for the proposed framework. Similarly, the reduction ratio achieved for the input data is also higher in contrast to the existing variant. Fig. 3 shows the reduction ratio achieved for the proposed framework. Here, an approximation ratio decreases from 98.2% to 88.7% in line with the reduction ratio depreciation from 87.2% to 11%. After analysis of these results, it is evident that higher level of originality is sustained for a higher reduction.
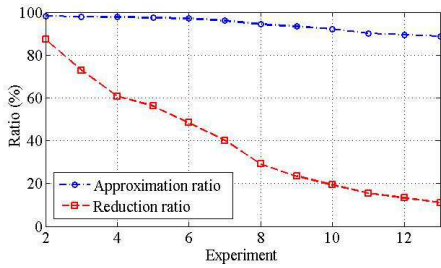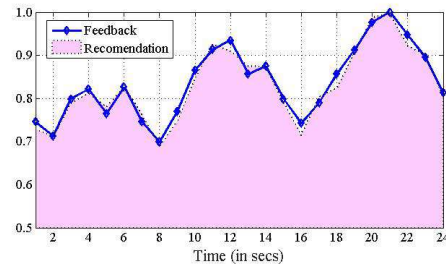
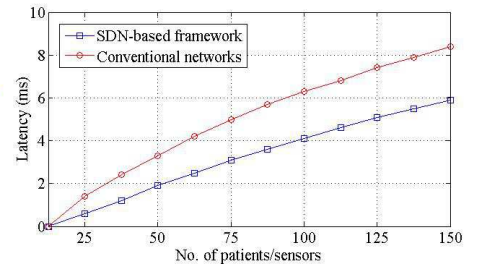Fig. 3. Reduction and approximation ratios

Fig. 4. Accuracy

Fig. 5. Latency

TABLE I
ERROR FOR DIFFERENT TECHNIQUES

Technique	RMSE	MAPE
Random forest	0.619	7.46 %
Decision tree	0.424	4.64 %

After the data is reduced, it is classified by constructing a disease specific decision tree. Once the disease data is classified, the RMSE and MAPE are computed for the D^{pr} and D^{ac}. Table I shows the values of RMSE and MAPE obtained for the proposed scheme. The results obtained show the superiority of the proposed scheme in contrast to other variants of its category. Finally, the healthcare data is provided to CNN model as an input, which uses convolution and pooling layers to learn the hidden patterns. Once, the healthcare profiles are extracted, then they are used to provide recommendation to the patients. After this process is complete, the patients provide the feedback for the recommendations received by them. Fig. 4 shows the evaluation results obtained for the DLRS framework. In the end, it is necessary to evaluate the latency for transmitting the healthcare data over SDN-enabled framework. Fig. 5 shows the overall latency incurred, which is lower in contrast to the conventional networks.

VII. CONCLUSION

In this paper, *DLRS: a deep learning based recommender system for SDN-enabled healthcare ecosystem* is designed. The major goal of *DLSR* is to provide a patient driven healthcare recommender systems, which tackles challenges such as; 1) handling huge amount of data generated by smart devices and sensors, 2) dynamic network management for real-time data transmission, and 3) lack of knowledge gathering and aggregation methods. Initially, the data generated be various sensors and queries entered by patients are collected and converted into tensorized form. After tensorization, the dimensionality reduction approach is applied on the data to obtain reduced data, which consists of less but valuable dimensions. After this step, the reduced data is classified using disease specific decision tree scheme. Finally, this data is provided to CNN model, which is responsible to provide the recommendations to patients. The proposed scheme is evaluated using a case study on the basis of various performance metrics such as-dimensionality reduction ratio, approximation ratio, accuracy, RMSE, MAPE and latency. The results obtained depict the

superiority of the proposed framework in comparison to the existing schemes of its category.

REFERENCES

[1] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das, and N. Saxena, "Lscsh: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 24–32, APRIL 2018.

[2] N. Kumar, K. Kaur, A. Jindal, and J. J. Rodrigues, "Providing healthcare services on-the-fly using multi-player cooperation game theory in internet of vehicles (iov) environment," *Digital Communications and Networks*, vol. 1, no. 3, pp. 191–203, 2015.

[3] A. Jindal, A. Dua, N. Kumar, A. K. Das, A. Vasilakos, and J. J. Rodrigues, "Providing healthcare-as-a-service using fuzzy rule-based big data analytics in cloud computing," *IEEE Journal of Biomedical and Health Informatics*, 2018, DOI: 10.1109/JBHI.2018.2799198.

[4] M. Wiesner and D. Pfeifer, "Health recommender systems: concepts, requirements, technical basics and challenges," *International journal of environmental research and public health*, vol. 11, no. 3, pp. 2580–2607, 2014.

[5] L. Duan, W. N. Street, and E. Xu, "Healthcare information systems: data mining methods in the creation of a clinical recommender system," *Enterprise Information Systems*, vol. 5, no. 2, pp. 169–181, 2011.

[6] N. T. Thong *et al.*, "HIFCF: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3682–3701, 2015.

[7] Q. Zhang, G. Zhang, J. Lu, and D. Wu, "A framework of hybrid recommender system for personalized clinical prescription," in *10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2015, pp. 189–195.

[8] N. D. Thanh, M. Ali *et al.*, "A novel clustering algorithm in a neutrosophic recommender system for medical diagnosis," *Cognitive Computation*, vol. 9, no. 4, pp. 526–544, 2017.

[9] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An sdn perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, Feb 2018.

[10] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data offloading in 5g-enabled software-defined vehicular networks: A stackelberg-game-based approach," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 100–108, 2017.

[11] D. Kaur, G. S. Aujla, N. Kumar, A. Zomaya, c. Perera, and R. Ranjan, "Tensor-based big data management scheme for dimensionality reduction problem in smart grid systems: Sdn perspective," *IEEE Transactions on Knowledge and Data Engineering*, 2018, doi: 10.1109/TKDE.2018.2809747.

[12] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[13] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1293–1298.

[14] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[15] F. Ongaro, "Enhancing quality of service in software-defined networks," Ph.D. dissertation, Alma Mater Studiorum-University of Bologna, 2014.