

Gated Recurrent Unit Architecture for Context-Aware Recommendations with improved Similarity Measures

Kala K. U.^{1*} and M. Nandhini²

^{1,2} Department of Computer Science, Pondicherry University
Puducherry-605014, India

[E-mail : ¹ kalasarin88.res@pondiuni.edu.in, ² mnandhini.csc@pondiuni.edu.in]

*Corresponding author : Kala K. U.

*Received May 27, 2019; revised September 16, 2019; accepted September 20, 2019;
published February 29, 2020*

Abstract

Recommender Systems (RecSys) have a major role in e-commerce for recommending products, which they may like for every user and thus improve their business aspects. Although many types of RecSys are there in the research field, the state of the art RecSys has focused on finding the user similarity based on sequence (e.g. purchase history, movie-watching history) analyzing and prediction techniques like Recurrent Neural Network in Deep learning. That is RecSys has considered as a sequence prediction problem. However, evaluation of similarities among the customers is challenging while considering temporal aspects, context and multi-component ratings of the item-records in the customer sequences. For addressing this issue, we are proposing a Deep Learning based model which learns customer similarity directly from the sequence to sequence similarity as well as item to item similarity by considering all features of the item, contexts, and rating components using Dynamic Temporal Warping(DTW) distance measure for dynamic temporal matching and 2D-GRU (Two Dimensional-Gated Recurrent Unit) architecture. This will overcome the limitation of non-linearity in the time dimension while measuring the similarity, and the find patterns more accurately and speedily from temporal and spatial contexts. Experiment on the real world movie data set LDOS-CoMoDa demonstrates the efficacy and promising utility of the proposed personalized RecSys architecture.

Keywords: Context Aware Recommender System, Recurrent Neural Network, Sequence Aware Recommender System, Dynamic Temporal Matching, Deep Recommender System Models.

1. Introduction

A Recommender System refers to a system that is capable of predicting the future preference of a set of items for a user, and recommends the top items [1]. The growth of RecSys has been progressed from the traditional RecSys based on missing rating findings using collaborative filtering [2], content-based filtering [3] and hybrid RecSys [4], to context aware [5], cross-domain [6] RecSys and their complexities in nature leads to Deep Learning based RecSys models [7]. Those systems provide a more personalized way of finding items of interest of each user within a huge collection of products. It has a wide range of applications in ecommerce, media streaming and for the improvement of our automated daily online experience. Such systems process the past data from the user community and analyses it for finding the patterns in the data and thus the probability of their interest toward the items. The items may vary depending upon the applications. It may be purchase history, watched movies, their clicks on advertisements, sites visited, etc.

The research in this field has traditionally based on the completion of matrix formulation. The interactions between each user and items (e.g., rating) from the past is given as a matrix, the major role of RecSys is to predict the missing interaction among them [2-4]. This will be suited for user's long-term behavior prediction with a view of there is one and only one interaction between each user and items exists. While we are considering user's short-term preferences, the traditional matrix method is not that much successful in RecSys like Session based ones [8].

The main drawbacks found in traditional RecSys, which utilize matrix formulation is that they only consider long terms sequence and only one interaction between user and item. In many real life applications user interactions may vary depends on contexts and time and needs to consider short-term interest also. In this scenario the concept and scope of Sequence aware recSys is arrived. Those consider recommendation as a sequence prediction problem [9].

The sequence modeling for RecSys utilizes one among Markov Models [10], Reinforcement Learning and Recurrent Neural Networks [11-15] for modeling the user sequences. Due to the complexity and data sparsity problems Markov models are not suited for every application. The sequence aware RecSys is one among the most successful application of Deep Learning which is in exploration now because of its powerfulness in computing very large and variety datasets. The recent researches show that from among the various DL techniques Recurrent Neural Networks (RNN) is well suited for Sequence Aware RecSys Models, especially the RNN variants such as LSTM and GRU.

The existing research works considers user sequences without considering individual users and items. That is, they are trying to find the similarity patterns from the one-dimensional sequences, there exists no consideration about users and their context and the features of the objects. Here we are proposing a RNN model, which helps to find the patterns from two-dimensional sequences, which includes both user and item characteristics for an improved personalized recommendation. However, evaluation of similarities among the customers is challenging while considering temporal aspects, context and multi-component ratings of the item record sequences in the overall customer sequences. For addressing this issue, we are proposing a Deep Learning based Model, which learns customer similarity directly from the item-item similarity as well as sequence-to-sequence similarity by considering all features of the item, contexts, and rating components using 2DRNN (Two Dimensional Recurrent Neural Network) Architecture [16,17]. This will learn the similarity between two item record

sequences through Dynamic Temporal pattern matching in customer sequences. Experiment on real world movie Data set LDOS-COMODA demonstrates the efficacy and promising utility of the proposed personalized RecSys Architecture.

The remaining sections of the paper has organized as follows: first, we discuss the relevant works related to sequential recommendation, application of RNN in different domains. Following this our proposed approach has depicted in detail which is followed by the experimental and evaluation part of our work. Finally, we conclude this paper and discuss potential future work.

2. Related Work

The recent researches in the area of RecSys have considered recommendation problem as a sequence prediction problem by measuring the similarity between various user sequences. Deep Learning techniques such as RNN is well suited for Sequence Aware RecSys Models, especially the RNN variants such as LSTM and GRU. Among the two gated RNNs, GRU is the one with less complicated recurrent cell and solves the vanishing gradient problem. While considering multiple contexts along with items in user sequences, there is a scope of 2D-GRU along with DTW for measuring the user similarity instead of Euclidean distance.

2.1. Sequence-Aware Recommendation

The state of the art RecSys has focused on finding the user similarity based on sequence (e.g. purchase history, movie-watching history) analyzing and prediction techniques like Recurrent Neural Network in Deep learning [11-15]. It utilizes the rich set of information maintained in a sequentially ordered user interaction logs of the applications for finding the user similarity. That is RecSys has considered as a sequence prediction problem to predict the next object in the sequence [9]. It is well suited for considering multiple interactions and both long and short-term patterns.

The relevance of sequence-aware RecSys is high in many practical applications and many works has proposed recently in this area. Sequence modeling is one among them. The input of this system is sequential and of time stamped list of user interactions in the past. The computational tasks have mainly focused on finding the user similarity patterns among the user's sequence [10]. In addition, the output is an ordered list of items that the user may interested in future.

2.2. Recurrent Neural Network

RNN based models have been widely used in many DL tasks when both inputs and outputs are of variable length in speech recognition, natural language processing etc. [11-15]. The uniqueness of RNN is that hidden states of the network is connected with both current and previous inputs of the network, which makes it suitable for sequence or time series based models. The structure of RNN is depicted in Fig. 1 and calculates hidden state h_t at time t from current input x_t , previous hidden state h_{t-1} as :

$$h_t = \tanh(Wx_t + Uh_{t-1}), \quad (1)$$

Where the matrices W and U are the parameters of the model.

In the conventional feed-forward neural networks, all test cases are considered to be independent. That is when fitting the model for a particular time, there is no consideration

for the data on the previous time steps. This dependency on time is achieved via Recurrent Neural Networks. Due to the problem of gradient explosion and vanishing, standard RNN has not been successful in finding long-term dependency well. For solving these issues, gated mechanism is embedded with RNNs and it leads to the development of Long Short Term Memory (LSTM) [18,19] and Gated Recurrent Unit (GRU) [20]. Fig. 1 Compares the structure of RNN variants.

2.2.1. Long-Short Term Memory(LSTM)

LSTM incorporates memory units for solving the vanishing gradient problem, which gives the ability to the network to learn when to forget and when to update the previous hidden states while considering a new information [18,19]. A traditional LSTM calculates hidden state h_t at time t from the current input x_t , previous hidden state h_{t-1} and the activation function \hat{h}_t calculated using a memory cell c_t , input gate i_t , a forget gate f_t , and an output gate o_t as:

$$i_t = \sigma(U_i x_t + W_i h_{t-1}) \quad (2)$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1}) \quad (3)$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1}) \quad (4)$$

$$\hat{c}_t = \tanh(U_g x_t + W_g h_{t-1}) \quad (5)$$

$$c_t = \sigma(f_t \odot c_{t-1} + i_t \odot \hat{c}_t) \quad (6)$$

$$h_t = \tanh(c_t) \odot o_t \quad (7)$$

Where σ a logistic is is function and \odot is an elementary multiplication operation.

2.2.2. Gated recurrent unit (GRU)

While LSTM has shown to be a viable option to avoid exploding/vanishing gradient problem, the memory cells in the architecture leads to increased memory requirement. GRU is also similar to LSTM, where as separate memory cell doesn't include in its architecture. GRU has an update and reset gate in the network, which deals with the update degree of each hidden states, that is it decides which information has to pass to the next state and which are not needed to be passed [9,10]. GRU calculates hidden state h_t at time t from the output of the update gate z_t , reset gate r_t , current input x_t , previous hidden state h_{t-1} is calculated as :

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (8)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (9)$$

$$\hat{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1})) \quad (10)$$

$$h_t = (1 - z_t)h_{t-1} + z_t \hat{h}_t \quad (11)$$

Where σ a logistic is function and \odot is an elementary multiplication operation.

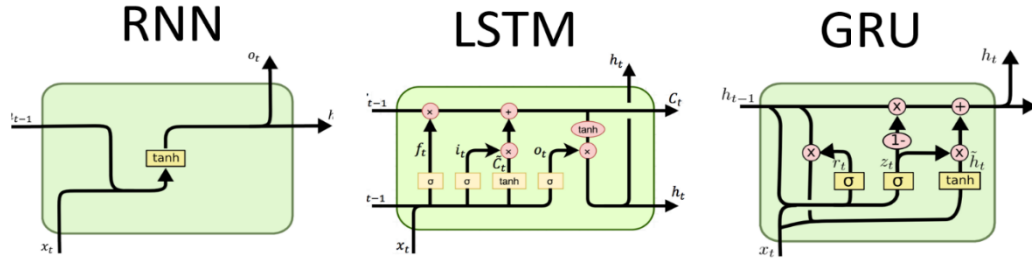


Fig. 1. Structure of vanilla RNN, LSTM and GRU [11,18,9]

2.3. Role of Gated RNNs in various Application Domains

RNN is a type of artificial deep learning neural network designed to process sequential data and recognize patterns in it (that's where the term "recurrent" comes from). Gated RNNs such as LSTM and GRU stand at the foundation of the modern-day marvels of artificial intelligence. They provide solid foundations for deep learning applications to be more efficient, flexible in its accessibility and most importantly, more convenient to use. Despite of the traditional sequence prediction applications, gated RNN models can effectively utilized for the state of the art AI applications. The trending application domains which are being effectively utilized and few recent papers are listed out in [Table 1](#).

Table 1. Application Domains of Gated RNNs

Application Domains	Author Name & paper ID	Sub-Domains	Gated-RNN Architecture
Natural Language Processing	Zheng et.al.[21]	Machine Translation	GRU
	Yao et al.[22]	Machine Translation	Depth Gated LSTM
	Viswanadan et al.[23]	Machine Comprehension	LSTM & GRU
	Jiao et al.[24]	Lexical Analysis	BI-GRU
	Chen et al.[25]	Relation Extraction	Bi-Tree-GRU
	Yao et al.[26]	NLP Analysis	Improved LSTM
	Mohammed et.al.[27]	Speaker, Language, and Gender Identification	LSTM
Speech/Audio/ Music Analysis & Synthesis	Garcia et al.[28]	Music Generation	LSTM & GRU
	Madhok et al.[29]	Music Generation	Stacked LSTM
	Song et al.[30]	Music tagging	GRU
	Xie e al.[31]	Speech emotion classification	LSTM
	Kang et al.[32]	Speech recognition	LSTM
	Nakyama et al.[33]	Audio Chord classification	LSTM & GRU
	Chen et al.[34]	Voice detection	GRU
Human Action/Interaction Recognition	Sho et al.[35]	Human interaction recognition	Hierarchichal LSTM
	Sho et al.[36]	Person-Person Action Recognition	Concurrent LSTSM
	Tang et al.[37]	Group Activity recognition	Coherence Constrained Graph LSTM
	Yan et al.[38]	Group Activity recognition	Bi-LSTM
	Vahora et al.[39]	Group Activity recognition	CNN, GRU/LSTM

	Vu et al.[40]	Human Activity Recognition	Self-Gated RNN(LSTM/GRU)
Bioinformatics	zhong et al.[41]	RNA- Sequencing	LSTM & GRU
	Anu et al.[42]	Protein Family classification	Vanila RNN,LSTM &GRU
	Shi et al.[43]	Protein Domain Boundary prediction	CNN & GRU
	Quoc et al.[44]	Protein identification	GRU
	Hanson et al.[45]	Protein disorder Prediction	Bi-LSTM
	Liu et al.[46]	Protein Function Prediction	LSTM
Business Process Management	Hinka et al.[47]	Process Instance Classification	GRU & LSTM
	Camargo et al.[48]	Business Process Monitoring	LSTM
	Nolle et al.[49]	Anomaly classification	LSTM & GRU
	Yan et al.[50]	Credit Risk Management	BI-GRU
Health Care Management	Jaganatha et al.[51]	Medical event detection	LSTM & GRU
	Ma et al.[52]	Risk Prediction	LSTM
	Pavithra et al.[53]	Prediction of Disease Progression	GRU
	Maragatham et al.[54]	Prediction of diseases	LSTM
	Lee et al.[55]	Clinical event prediction	LSTM
	Xu et al.[56]	Inpatient health care	LSTM
	Cheng et al.[57]	Treatment Migration prediction	GRU
	Che et al.[58]	Clinical event prediction	GRU
Recommender System	Zhu et al.[59]	Product recommendation	GRU
	You et al.[60]	Event recommendation	GRU & TCN
	Huang et al. [61]	POI recommendation	LSTM
	Douligeris et al. [62]	Tag recommendation	Bi-LSTM
	Livne et al. [63]	POI & Product recommendation	Encoder Decoder LSTM
	Doan et al. [64]	POI recommendation	LSTM
	Hu et al. [65]	POI recommendation	BiGRU
	Huang et al. [66]	Product recommendation	LSTM & GRU
	Zheng et al. [67]	Music recommendation	LSTM
	Li et al. [68]	Movie recommendation	LSTM & CNN
	Heinz et al. [69]	Fashion recommendation	LSTM
	Tanveer et al. [70]	User activity recommendattion	Stacked LSTM

From the [Table 1](#), it is clear that most of the applications are using LSTM or GRU alternatively for solving the Vanishing Gradient problem. Some researchers experiment their model with both network and compare their results [\[23,28,33,39,40-42,47,49,51\]](#). The prediction of which model will be good is difficult. We choose GRU for our application because of its less complicated recurrent cell (it uses two gates instead of 3, like an LSTM). Also, in contrast to LSTM, the GRU exposes the whole state at each time step and computes a linear sum between the existing state and the newly computed state.

2.4. GRU for RecSys models

In this section, we briefly discuss about the various ResSys models, which utilizes GRU architecture for finding user similarity and predicting the next items in the sequence aware RecSys.

The specific properties of RNN make them suitable for sequence modeling applications [76]. Gated architectures of RNN includes gating units for controlling information flow over the network and makes it suitable for processing long term sequences. Thus, GRU came in action for such systems. As a first attempts GRU has used for sequential recommending without any modifications [77-82]. Global behavior among the users has measured by treating every sequence equivalently using any similarity measuring techniques.

The proposed approaches in literature more often focus only the user consumption sequences without considering content, context and user specific information. The focus of GRU Model proposed in [77] is pair wise learning for session-based systems. Hidasi et al. [78] enhances this pair wise approach to include feature rich content information simultaneously as input to the GRU layer. GRU4Rec is the basic GRU for RecSys. In [82] sequence learning and user characteristics learning have been done independently by using RNN and feed forward network respectively. In [83] they utilize two separate RNN for training and modeling sequence and user information. The outputs of both RNN have used further for processing auxiliary parameters.

Tan et al. [80] deals content information in preprocessing step to reduce the effect of outdated features and the resultant subset is processed by the second model. Session based RecSys have implemented effectively with GRU [85]. Recently contexts have also considered as input along with the sequence data [86-89].

It is important to deal with high-dimensional content information in this big data era. While most of the sequential models developed so far has designed for low-dimensional data. High dimensional data contents leads to more accurate and personalized recommendation. 2D GRU is a variant of GRU to model 2 dimensional (matrix or tensor) data, which utilizes a different sequential model for matching signals instead of the hierarchical approach in traditional GRU. The concept of 2D GRU, introduced in Match-SRNN [16], recursively scans from top left to bottom right in a spatial data. Later in Information Retrieval, DeepRank [17] architecture effectively utilizes this concept and compares with Convolutional Neural Networks(CNN).

Most of the models find similarity based on Euclidean distance measures, here we are introducing DTW algorithm for that purpose. It overcomes the problem of sensitivity to distortion along time axis. DTW is distance measure algorithm for pattern detection. It measures the similarity between two speed varied temporal sequences. Dynamic programming approach has used in DTW for minimizing the distance measure such as Euclidean distance. The approximate optimal alignment of two sequences has carried out by a warping path. DTW has already proved his role in the area of speech recognition [71], DNA Sequence mining [72], online streaming monitoring [73] and entertainment [74]. It overcomes the weakness of Euclidean measure (sensitivity to distortion in time axis) thus; it achieves great success in many time series pattern-matching applications [75].

We are proposing 2D-GRU for efficiently handling two dimensional data sequences, which contain user contents, item contents, ratings and their contexts along with DTW for similarity measures.

This work progresses along three directions:

- 1) A Deep learning model for directly learning customer similarity from all aspects of available contexts using the GRU architecture.
- 2) Personalized RecSys model that is able to adapt various classification techniques in the recommendation step over a few further timestamps,
- 3) Modification of the model by adding a preprocessing module makes it suitable for multi-domain applications.

3. Proposed GRU Architecture for Context Aware Recommendations

To cope up with the challenges of personalized sequence aware predictions, this paper proposing a GRU architecture for computing the similarity between user data sequences by using a structure like DTW, which brings adequate temporal dynamics for user sequences. The three step temporal matching structure has shown in Fig. 2. In first step the distance between each records of two user data sequences are measuring using DTW [71-75], in the second step 2D-GRU is applying for calculating the overall similarity among two customer data matrix/tensors [16,17] and the third step focus on calculating the final distance by applying linear scoring function.

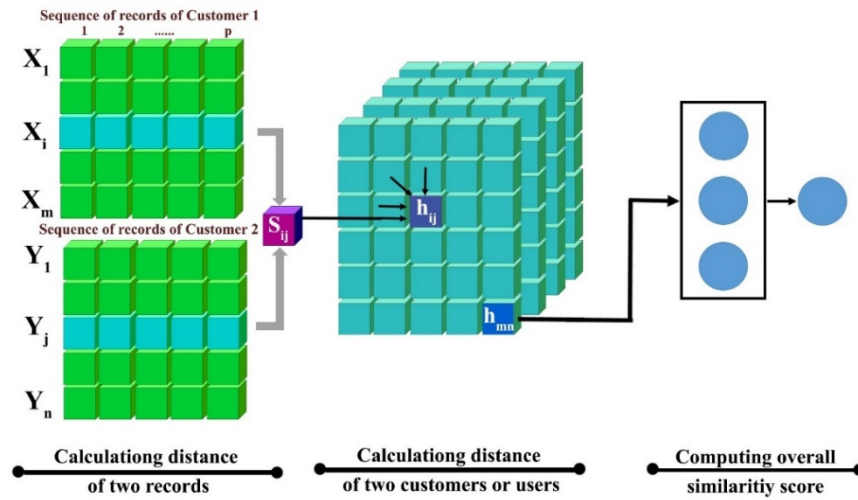


Fig. 2. Proposed GRU Architecture

3.1. Setup and Input

Let C is the set of N customers represented as $C = \{C_1, C_2, \dots, C_N\}$. The customers buying history is organized as two-dimensional sequences, which includes both user and item characteristics for an improved personalized recommendation. Each customer data sequence has represented as set of unified vectors, which represents each item purchase records of them. Given two customer sequences, $C_1 = \{X_1, X_2, \dots, X\}$ and $C_2 = \{Y, Y_2, \dots, Y_n\}$, where X_i and Y_j represents the i^{th} and j^{th} purchase vector of customer C_1 and C_2 respectively and it is represented as $C_1(X_i) = (X_{i1}, X_{i2}, \dots, X_{ip})$ and $C_2(Y_i) = (Y_{i1}, Y_{i2}, \dots, Y_{ip})$. Thus the

problem is to learn customer similarity directly from the item-item similarity as well as sequence-to-sequence similarity by considering all features of the item, contexts, and rating components.

3.2. Phases of Proposed GRU Architecture

Our proposed model has 3 stages for measuring the similarity between multimodality customer purchase sequences as shown in [Fig. 2](#):

Phase1: Calculating the distance of two records with improved temporal matching based similarity measures.

Phase 2: Calculating the distance between each pair of customer sequences using 2D-GRU.

Phase 3: Computing the overall similarity score using linear scoring functions.

3.2.1. Phase 1: Improvements in Record-Similarity Measuring Technique using DTW

For each customer, after normalizing the data to a unified vector from multiple modalities, the similarity between them is calculated. Typically, by applying Euclidean distance measure [\[96\]](#), the similarity is calculated by measuring the distance between the sequences.

Euclidean distance (d_{ij}) is calculated using equation (7) and thus the similarity is measured.

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - y_{ik})^2} \quad (6)$$

For calculating sequence similarity, first we have to calculate current item-record similarity and then previous sequence similarity. The term previous sequence denotes the sequence of purchase records from 1st to i^{th} . For measuring the similarity between the prefixes, $C_1[1:i]$ and $C_2[1:j]$ has to be calculated from the distance between the sub prefixes $\hat{h}_{i-1,j}$, $\hat{h}_{i,j-1}$, $\hat{h}_{i-1,j-1}$ and the current record distance as

$$\hat{h}_{i,j} = f(\hat{h}_{i-1,j}, \hat{h}_{i,j-1}, \hat{h}_{i-1,j-1}, \hat{d}(x_i, y_j)) \quad (7)$$

Where $\hat{d}(x_i, y_j)$ represent the distance between i^{th} item record of C_1 and j^{th} item record of C_2 and $\hat{h}_{i-1,j}$ denotes the distance among the prefixes $C_1[1:i-1]$ and $C_2[1:j]$.

For modeling temporal dynamics, the direct similarity measuring techniques is not efficient. The temporality of events (purchasing, watching movie) in their sequence may differ in local arrangement and speed, although the customers show overall similarity trends. That is two customers may buy similar products or watching similar movies, but the timestamps of their purchase may differ or frequency may differ, all those things leads to noise in learning. To overcome the limitation of non-linearity in the time dimension, a deep learning structure, inspired by DTW is designed. The DTW distance is calculated using the below equation (7) in a recursive way.

$$dtw(i, j) = d(x_i, y_j) + \min \{dtw(i-1, j), dtw(i, j-1), dtw(i-1, j-1)\} \quad (8)$$

Where $d(x_i, y_j)$ is the distance of two observations x_i and y_j .

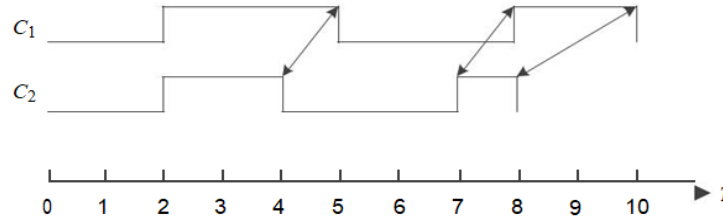


Fig. 3. The DTW matching Process

Fig. 3 illustrates the matching process between two customers C1 and C2 using the design inspired by DTW [72].

For the simplicity of illustration, only binary features are considered. Given customers $C1 = (0, 0, 1, 1, 1, 0, 0, 0, 1, 1)$ and $C2 = (0, 0, 1, 1, 0, 0, 0, 1)$, the i^{th} feature value of C1 is denoted as x_i and the j^{th} feature value of C2 is denoted as y_j . The two waveforms in **Fig. 3** represent the two customers C1 and C2. There exists overall similarity between the waveforms of C1 and C2, because their difference in dimensionality Euclidean distance will not apt. By aligning C1 and C2 in time dimension, similarity can measure using DTW. At timestamp 5 as in **Fig. 3**, the time axis is “Warped” by measuring the distance of (x_5, y_4) instead of (x_5, y_5) since (x_5, y_4) is shorter in comparison. The necessity of warping is also exists at (x_8, y_7) and (x_{10}, y_8) .

3.2.2. Phase 2: Calculating the distance between two Customer sequences

For measuring the similarity customer purchase record sequences, an RNN architecture is introduced along with a DTW to improve the temporal dynamics in the customer sequences. The concepts of DTW and 2D-GRU along with ranking loss functions can be combined and form a deep architecture for temporal sequence similarity learning. **Fig. 4** depicts the structure of a 2D-GRU, which has chosen as function f .

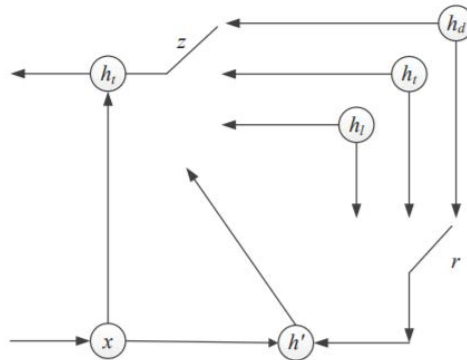


Fig. 4. Structure of a 2D-GRU unit[16]

The two gates of GRU, update \vec{z} and reset \vec{r} , controls the three directional inputs from previous units. The value of x is directly inputting to $h_{i,j}$.

The vanishing gradient problem of RNN has resolved by GRU with two gates of it. Here we are utilizing 2D-GRU, the enhanced version of traditional GRU. In this the information to be discarded is decided by the reset gate \vec{r} and the storage of the information in the hidden-state is decided by the update gate \vec{z} . The input from previous unit comes along three

directions, (i, j-1), (i-1, j) and (i-1, j-1), for the particular position (i, j) and it is represented as l, t and d. so both gates has the control of information along 3 directions. The unit also has another input along with the previous three, that is d_{ij} , the distance. At time T, by concatenating the vectors $\hat{h}_{i-1,j}^t, \hat{h}_{i,j-1}^t, \hat{h}_{i-1,j-1}^t, d_{i,j}$, and the input vector \hat{q}^t is formed by the following formulae.

$$\hat{q}^t = ([\hat{h}]_{i-1,j}^t, \hat{h}_{i,j-1}^t, \hat{h}_{i-1,j-1}^t, d_{i,j}) \quad (9)$$

The value of the two gates are computed as

$$\hat{r}^t = \sigma(W^r \hat{q} + \hat{b}^r) \quad (10)$$

$$\hat{z}^t = \sigma(W^z \hat{q} + \hat{b}^z) \quad (11)$$

Where W^r and W^z are the coefficient of weights corresponding to both gates, and the threshold of both gate is represented as \hat{b}^r and \hat{b}^z

Overall matching score \hat{h}'_{ij} is calculates as:

$$\hat{h}'_{ij} = \tanh(\hat{w}d_{ij} + U(\hat{r} \odot ([\hat{h}]_{i-1,j}^t, \hat{h}_{i,j-1}^t, \hat{h}_{i-1,j-1}^t)^T) + \hat{b}) \quad (12)$$

The hidden state is computed as

$$\hat{h}_{i,j} = W^m(\hat{z} \odot ([\hat{h}]_{i-1,j}^t, \hat{h}_{i,j-1}^t, \hat{h}_{i-1,j-1}^t)^T) + U^m(1 - \hat{z})\hat{h}'_{ij} + \hat{w}d_{ij} \quad (13)$$

Where W^m and U^m are the reset gate parameters, \hat{w} weight corresponding to distance between customers. \odot Position vice multiplication of elements (Hadamard product)

3.2.3. Phase 3: Overall similarity Measure and Optimization

The customers overall similarity can be computed from the last output of the 2D-GRU model, i.e. \hat{h}_{mn} , using a linear function, since the model starts scanning the inputs from along the direction (0,0) to (m, n).

The overall similarity has computed using the formula:

$$S(C_1, C_2) = W^s \hat{h}_{mn} + \hat{b}^s \quad (14)$$

Where W^s and \hat{b}^s represents linear function parameters.

As a first step of the optimization, we select the loss function as pairwise ranking loss. If the matching score of (C_1, C_2^+) is less than that of (C_1, C_2^-) in the triplet (C_1, C_2^+, C_2^-) , the pairwise loss function is defined as

$$L(C_1, C_2^+, C_2^-) = \max(0, 1 + M(C_1, C_2^+) - M(C_1, C_2^-)) + \lambda \|\theta\|_2^2 \quad (15)$$

Where θ indicates the parameters of the GRU, constitutes the reset gate parameters W^r, \hat{b}^r , update gate parameters W^z, \hat{b}^z , memory gate parameters \hat{w}, U, \hat{b} , and the dimension transformation parameters W^s, \hat{b}^s .

$$\theta = \{W^r, \hat{b}^r, W^z, \hat{b}^z, \hat{w}, U, \hat{b}, W^s, \hat{b}^s\} \quad (16)$$

Back-propagation along with mini-batch SGD (Stochastic Gradient Descent) with AdaGrad is used to train the parameters of the GRU network.

3.3. Personalized Recommendation

Customer similarity learned from the model has utilized for personalized recommendation of items of their preference under various contexts. For a queried customer, first N most similar customers have retrieved to create a sub-population. Here we recommend ordered items that the user may purchase on next few time-periods, the recommendation task has been considering as a task of ordinal classification. For the classifier the model is capable of admitting any classifier with multiclass classification. In this work, we are using multi class

Support Vector Machine (SVM) classifier using one versus rest strategy and the classifier model has trained using the sub-population of the similar patients.

4. Results and Discussion

The main difficulty for experimenting with our proposed architecture is the lack of available context rich data set. A recent survey has conducted that focused on the dataset available for the context aware RecSyss[90]. From that survey, we found 5 datasets, which focus on different types of items and include corresponding contexts: movies in movies in DePaulMovie and LDOS-CoMoDa songs in InCarMusic apps in Frapp´ e Points of Interest (POIs) in STS and hotels in the TripAdvisor datasets for implementing DTW-2DGRU model. We choose one among them by looking various statistics of the datasets. We evaluate the performance of DTW-2DGRU model with various evaluation metrics used in RecSyss such as Precision, Recall, F-Measure, MSE, RMSE. Then compare the results with various existing recurrent context aware recommender systems for showing the improvements in performance and accuracy of our DTW-2DGRU Model.

4.1. Datasets

We evaluate the performance of our architecture for context aware recommendation; we use one among the six available context aware datasets, which focus on different types of items such as movies, music, apps, Point of Interest and hotels. **Table 2** shows statistics and contextual attributes of each dataset, which made it suitable for our purpose.

For our architecture implementation and evaluation, we select LDOS-CoMoDa[91] Dataset because it has adequate number of contextual attributes and less percentage of unknown contextual values as shown in **Table 2**. It is a movie dataset focused on specific use cases of movies. LDOS- CoMoDa consists of both dynamic contexts (varies according to time and state of the user) and static contexts (information related to items that will never change). Our architecture is suitable for both type of contexts, it deals all the contexts in the form of a matrix. There are 4% of missing values exists in the database it is marked as -1.

Table 2. Statistics of the Data Sets along with contextual attributes and percentage of unknown context values

Dataset	Brief Description	Context attributes	#users	#Items	#Ratings	% of unknown values
DePaulMovie	Movie	Time, Location, Companion	97	79	5043	28.71 %
LDOS-CoMoDa	Movie	Time, Day type, Season, Location, Weather, Social environment, End emotion, dominant emotion, Mood, Physical status, Decision point, Interaction status	121	1232	2296	4.00%
InCarMusic	Music	Landscape, Mood, Natural phenomena, Road type, Sleepiness, Traffic conditions, Weather	42	139	4012	90.54 %
Frappe	Apps	Daytime, Weekday, Homework, Weather, Country, City	957	4082	96203	23.09 %

STS	Point of Interest	Distance, Time available, Temperature, Crowdedness, Knowledge of surroundings, season, Budget, Daytime, Weather, Companion, Mood, Weekday, Travel goal, Transport	325	249	2534	89.37 %
TripAdvisor	Hotels	Trip type	1202 , 2371	1890 , 2269	14175 , 28350	0%

Table 3. Contextual details of LDOS - CoMoDa dataset

Type of Context	Data Fields	Values	Contextual variables
Dynamic	itemID	1 -4138, some missing	
	rating	1-5	
	time	1-4	Morning, Afternoon, Evening, Night
	daytype	1-3	Working day, Weekend, Holiday
	Season	1-4	Spring, Summer, Autumn, Winter
	location	1-3	Home, Public place, Friend's house
	weather	1-5	Sunny / clear, Rainy, Stormy, Snowy, Cloudy
	social	1-7	Alone, My partner, Friends, Colleagues, Parents, Public, My family
	endEmo	1-7	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
	dominantEmo	1-7	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
	mood	1-3	Positive, Neutral, Negative
	physical	1-2	Healthy, Ill
	decision	1-2	User decided which movie to watch, User was given a movie
	interaction	1-2	First interaction with a movie, n-th interaction with a movie
Static	movie director	1-820	Names of directors
	movie's country	1-37	Country name
	movie's language	1-25	English, French, etc.
	movie's year	Any year	Any year
	genre1	1-25	Action, comedy, Drama, etc
	Genre2	1-25	Action, comedy, Drama, etc
	Genre3	1-25	Action, comedy, Drama, etc
	actor1	1-2000	Name of the actor
	Actor2	1-2000	Name of the actor
	Actor3	1-2000	Name of the actor
	movie's budget	Amount	Budget of the movie

4.2. Experimental setup

The data in our dataset has missing value with both static and temporal contexts and needs temporal record wise mapping for finding similarity. For preprocessing the data, we choose all attributes that are dynamically varying and those are static related to movies along with the movie ids as in [Table 3](#). The last occurrence carry-forward strategy has used for handling missing values. The first record of a customer has filled with the first ever-observed record when it is missing. We removed the users with less interaction and movies that are less watched and if ratings are missing it is allotted 1. If all values corresponding to one context of a customer is not exists, it is filled with the mean of all observed values.

For implementing the architecture, the size of the batch used in SGD (Stochastic Gradient Descent) is set to be 10, the value has chosen as small because there are only 200 customers/users in CoMoDa dataset. Uniform distributions used for randomly initializing all the other parameters. As we use pair wise loss function, we have to find a triplet (C, C_2^+, C_2^-) by finding the set of positive customers and negative customers (C_2^+ and C_2^-) for each customer (C). All customers are ranked by similarities with C , and then 5 most similar customers are selected as positive and select 5 customers with highest rank as negative. Thus, 32 triplets are for each customer and total training set has increased 32 times of the number of customers. Thus, it resolves over-fitting issue to some extent.

4.3. Evaluation metrics

To evaluate the performance of the sequence aware RecSys, standard classification and ranking metrics has used in many application scenarios. The various reviews papers include the evaluation metrics such as Precision, Recall, Mean Average Rank (MAR), Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Mean Average Precision (MAP) and F1 [\[92\]](#). The performance of the overall personalized recommendation task is measured using Recall, MAP, NDCG and HR. Since, Context aware sequence recommendations are highly correlated with classification problem so classification metrics such as precision and recall will be more suitable. MAP and NDCG also measured when it is a time step ahead ranking type prediction. So here, we are evaluating the performance of our proposed architecture with Precision, Recall, MAP and NDCG.

For evaluating the precision of recommendation, we use precision@K and Recall and for evaluating the ranking precision of recommended items MAP and nDCG metrics are measuring through offline experiments using the dataset LDOS –CoMoDa. More details of the used evaluation metrics have described in [Table 4](#).

Table 4. Evaluation metrics and their description used for DTW-2DGRU Architecture

Evaluation Metric	Description	Equation	Parameters
Precision [92]	The proportion of user-preferred items in recommendation system	$\text{Precision} = \frac{N_{rs}}{N_s}$ $\text{Precision@n} = \frac{N_{rs@n}}{n}$	N_{rs} – No. of recommended items that user prefer N_s - No. of recommended items n - No. of first k recommended items
Recall [92]	The proportion of user-favorite items will not missed to recommend is described	$\text{Recall} = \frac{N_{rs}}{N_r}$	N_r - No. of items that user prefer

MAP(Mean Average Precision)[92]	The average of multiple recommendation precision	$MAP = \frac{\sum_{q=1}^Q Avgp(q)}{Q}$ $Avgp(q) = \frac{\sum_{k=1}^n P(k) * rel(k)}{\#relevant\ item}$	Q- No. of recommendation k- rank rel(k)- the relativity function given rank k p(k)- the precision given rank k
nDCG[92]	The importance of each position of the recommended items is represented	$nDCG = \frac{DCG(r)}{DCG(r_{perfect})}$ $DCG(r) = \sum disc(r(i))u(i)$	Desc(r(i))- discounted function which makes previous items more important. U(i)- utility of items in the list DCG($r_{perfect}$) - perfect ranking of discounted cumulative function

4.4. Evaluation results of DTW-2DGRU Model

Evaluation of DTW-2DGRU architecture has to do in two perspectives: first focus is customer similarity learning and second is on personalized context aware recommendation. For learning the similarity of customers with dynamic temporal matching, we use DTW concept in our model. For evaluating this, we compare it by setting Euclidean measure as the baseline for measuring the similarity. In this, the Euclidean distance between the feature vectors has calculated as a similarity measure. To evaluate the performance of similarity learning, our method has compared with baseline Euclidean approach. The result has shown in Fig. 5.

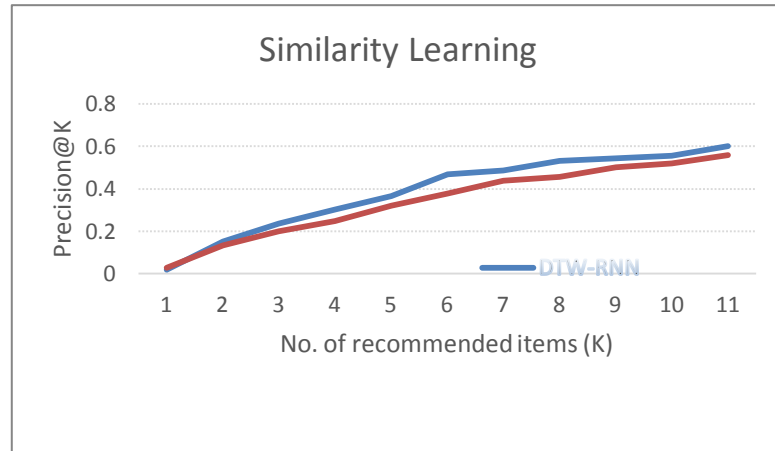


Fig. 5. Precision@K for DTW-RNN and Euclidean Distance

Fig. 5 depicts the similarity learning performance evaluation results of our method by drawing the line graph with Precision@k corresponding to k (number of recommended items). DTW-RNN performs better than the baseline. For evaluating the recommendation efficiency, we took various GRU based Context Aware Recommendation architectures as baseline and compared them with our model using the metrics recall, MAP and nDCG.

4.4.1. Baselines

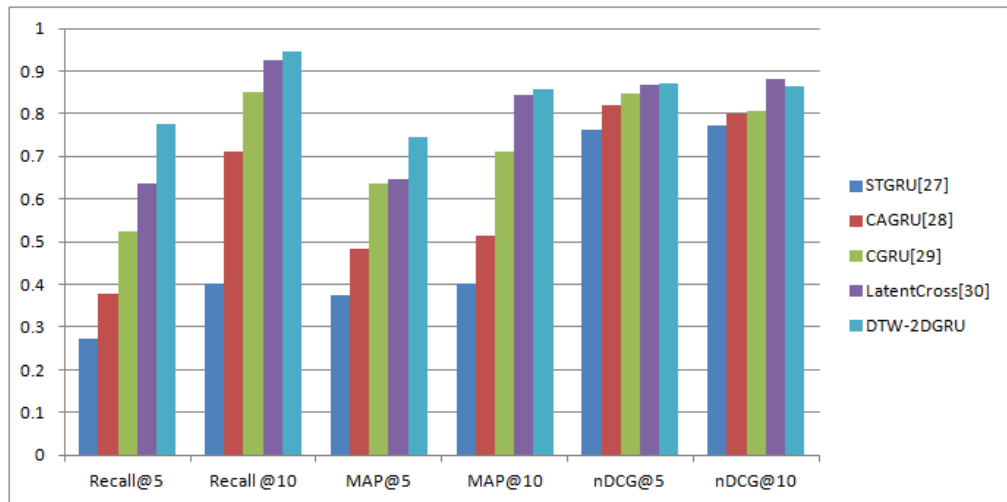
The base of all GRU models for RecSys is GRU4REC [20], but it only deals with the sequence of items, contexts are not considering for recommendation. We choose different GRU based RecSys architectures, which deals with at least one context as baseline. The features of various existing GRU RecSys have shown in Table 5.

Table 5. Features of baseline architectures

Features	GRU ARCHITECTURES					
	STGRU[88]	CAGRU[86]	CGRU[87]	LatentCross[93]	DTW-2DGRU	Cross DTW-GRU
Sequential-based	✓	✓	✓	✓	✓	✓
Context Aware	✓	✓	✓	✓	✓	✓
Static/Dynamic contexts	Only dynamic	One context	One context	✓	✓	✓
Multi-Component Rating	×	×	×	×	✓	✓
Cross-Domain	×	×	×	×	✓	✓

Table 6. Performance in terms of Recall MAP and nDCG metrics among various approaches

Metrics	GRU ARCHITECTURES				
	STGRU[88]	CAGRU[86]	CGRU[87]	LatentCross[93]	DTW-2DGRU
Recall@5	0.2728	0.3776	0.5236	0.6347	0.775
Recall @10	0.4013	0.7102	0.8491	0.9265	0.946
MAP@5	0.3756	0.485	0.636	0.646	0.7461
MAP@10	0.4013	0.5149	0.7124	0.8453	0.8563
nDCG@5	0.7604	0.8208	0.8484	0.8686	0.8725
nDCG@10	0.7717	0.7991	0.807	0.8802	0.8648

**Fig. 6.** Performance Comparison in terms of different metrics for the GRU- Architectures

On re-implementing various existing architecture baselines along with our proposed approach on LDS-CoMoDa data set, our approach shows significant improvement on different metrics recall and MAP as shown in Fig. 6. However, in terms of nDCG latent cross outperforms our approach. The values of various metrics have shown in Table 6. These results imply that our proposed DTW-2DGRU architecture with improved similarity measure and 2D input sets is more efficient than the various state of the art GRU architectures in modeling user sequences.

5. Modification for Multi-Domain Applications

Our major difficulty during training the model is the lack of rows in the dataset, i.e. the sequences created using the dataset is not large enough to model. We extend our model to multi-domain so that we can utilize the same user's data from multiple domains/datasets. Our proposed architecture makes it suitable for Cross-Domain (Multi-Domain) RecSys by adding a preprocessing module, which selects contexts common to selected modules and then merging records from multiple domains of a single user into a single sequence.

The baseline for this is CCCFNet (Content-Boosted Collaborative Filtering Network for Cross Domain Recommender Systems) [94] and compare our cross-DTW architecture with this. Fig. 7 shows the modified DTW-2DGRU architecture for multi-domain recommendations. And Table 7 compares its performance with existing CCCFNet Architecture by considering LDOS-CoMoDa as dataset of one domain and STS [95] as another domain by assuming both has common contexts and users, so that we can extend our user sequences. Due to the lack of availability of the data set, we slightly modified the STS Dataset so that common users exist in both.

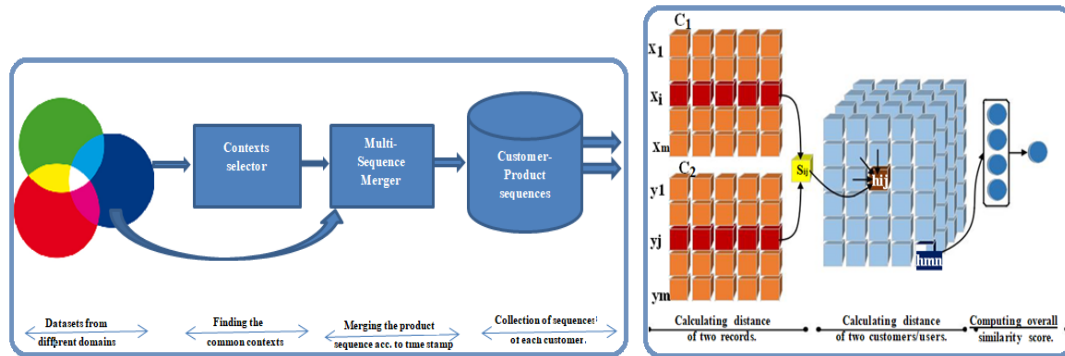


Fig. 7. Cross-DTW-2DGRU Architecture with Additional Pre-processing Module

Table 7. Performance measures for CCCFNet and Cross-DTW-2DGRU

Metrics	Multi-Domain architectures	
	CCCFNet	Cross-DTW-2DGRU
Recall@5	0.3586	0.5456
Recall @ 10	0.8103	0.9482
MAP@5	0.4348	0.6786
MAP@10	0.6582	0.7125
nDCG@5	0.8245	0.8487
nDCG@10	0.7674	0.7872

6. Discussion

We find the following information from the experiments

- The performance of the personalized recommendation system models is highly influenced by the size of similar user sub-group.
- The DTW-GRU model outperforms with a moderate performance gain because of the effectiveness the adopted similarity learning technique.
- Similar customers found by our model, increases the performance of the recommended model than those models, which use Euclidean distance.
- After finding sub-population of similar customers, recommender system problem has considered as an ordinal classifier problem, we tried with KNN also, from among SVM gives high performance.
- Our Multi-domain context aware model gives the ability to combine data from of a user from multiple domains, by considering their contexts too.

7. Conclusion and Future Works

This research work initially focused on the development of an effective deep model for learning the customer similarity by considering both static and dynamic contexts. The proposed DTW-GRU model directly learns customer similarity from all aspects of available contexts by dynamically matching the temporal patterns in the user data sequences. We further designed a personalized recommender system model that is able to adapt various classification techniques in the recommendation step. We again modify this model by adding a preprocessing module, which made it suitable for multi-domain applications. This study mainly focused on two directions: similarity learning and context aware recommendation. The potential future-work may focus on in any of these directions for better accuracy and performance of the recommender systems.

Acknowledgement

This research work is performed as part of the Ph.D work in the area of Context-Aware Recommender System. There is no funding source(s) involved in this research.

References

- [1] Francesco Ricci, Lior Rokach and Bracha Shapira, "Introduction to Recommender Systems Handbook," *Recommender Systems Handbook*, Springer, pp. 1-35, 2011. [Article \(CrossRef Link\)](#)
- [2] Xiaoyuan Su, Taghi M. and Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence archive*, 2009. [Article \(CrossRef Link\)](#)
- [3] Macedo AA, Pollettini JT, Baranauskas JA and Chaves JC, "A Health Surveillance Software Framework to deliver information on preventive healthcare strategies," *J Biomed Inform.*, 62, 159–170, 2016. [Article \(CrossRef Link\)](#)
- [4] Robin Burke, "Hybrid Web Recommender Systems," *the Wayback Machine*, pp. 377-408, 2014. [Article \(CrossRef Link\)](#)

- [5] Imen Ben Sassi, Sehl Mellouli and Sadok Ben Yahia, "Context-aware recommender systems in mobile environment: On the road of future research Information Systems," *Information Systems*, Vol. 72, pp. 27-61, 2017. [Article \(CrossRef Link\)](#)
- [6] P. Cremonesi, A. Tripodi and R. Turrin, "Cross-Domain Recommender Systems," in *Proc. of 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC*, pp. 496-503, 2011. [Article \(CrossRef Link\)](#)
- [7] Shuai Zhang, Lina Yao, Aixin Sun and Yi Tay, "Deep Learning based Recommender System: A Survey and New Perspectives," *ACM Comput. Surv.*, vol.52, no.1, 2019. [Article \(CrossRef Link\)](#)
- [8] C. Chen and C. Chang, "Evaluation of Session-Based Recommendation Systems for Social Networks," in *Proc. of 2013 IEEE 13th International Conference on Data Mining Workshops, Dallas, TX*, pp. 758-765, 2013. [Article \(CrossRef Link\)](#)
- [9] Massimo Quadrona, Paolo Cremonesi and Dietmar Jannach, "Sequence-aware Recommender Systems," in *Proc. of UMAP*, 373-374, 2018. [Article \(CrossRef Link\)](#)
- [10] Rendle S., Freudenthaler C. and Schmidt-Thieme L., "Factorizing personalized markov chains for next basket recommendation," in *Proc. of WWW*, 811-820, 2010. [Article \(CrossRef Link\)](#)
- [11] X. Zhang and M. Lapata, "Chinese poetry generation with recurrent neural networks," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25- 29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. *ACL*, pp. 670-680, 2014. [Article \(CrossRef Link\)](#)
- [12] Q. Wang, T. Luo, D. Wang and C. Xing, "Chinese song iambics generation with neural attention-based model," in *Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA*, pp. 2943-2949, 9-15 July 2016.
- [13] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds. *Association for Computational Linguistics*, pp. 1657-1668, 2017. [Article \(CrossRef Link\)](#)
- [14] V. Tran and L. Nguyen, "Semantic refinement gru-based neural language generation for spoken dialogue systems," in *Proc. of International Conference of the Pacific Association for Computational Linguistics*, pp. 63-75, 2017. [Article \(CrossRef Link\)](#)
- [15] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multitask learning for deep text recommendations," in *Proc. of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds. *ACM*, pp. 107-114, 2016. [Article \(CrossRef Link\)](#)
- [16] Shengxian Wan, YanyanLan, JiafengGuo, Jun Xu, Liang Pang and Xueqi Cheng, "Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN," in *Proc. of IJCAI-2016*, 2955-2928, 2016.
- [17] Pang, Liang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu and Xueqi Cheng, "DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval," *CIKM*, pp. 257-266, 2017. [Article \(CrossRef Link\)](#)
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. [Article \(CrossRef Link\)](#)
- [19] A. Graves, N. Jaitly and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic*, pp. 273-278, 2013. [Article \(CrossRef Link\)](#)
- [20] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. *ACL*, pp. 1724-1734, 2014. [Article \(CrossRef Link\)](#)

- [21] Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen and Zhaopeng Tu, "Modeling Past and Future for Neural Machine Translation," *Transactions of the Association for Computational Linguistics*, Vol. 6, pp.145-157, 2018. [Article \(CrossRef Link\)](#)
- [22] Yao K., Cohn T., Vylomova E., Duh K. and Dyer C., "Depth-Gated LSTM," *ArXiv, abs/1508.03790*, 2015.
- [23] Viswanathan S., Anand Kumar M. and Soman K.P., "A Sequence-Based Machine Comprehension Modeling Using LSTM and GRU," *Emerging Research in Electronics, Computer Science and Technology. Lecture Notes in Electrical Engineering*, vol 545, pp. 47-55, 2019. [Article \(CrossRef Link\)](#)
- [24] Jiao Z., Sun S., and Sun K., "Chinese Lexical Analysis with Deep Bi-GRU-CRF Network," *arXiv e-prints arXiv:1807.01882*, 2018.
- [25] Chen D., Wu Y., Le J., and Pan, Q., "Context-Aware End-To-End Relation Extracting from Clinical Texts with Attention-based Bi-Tree-GRU," *ILP Up-and-Coming / Short Papers.*, 2018.
- [26] L. Yao and Y. Guan, "An Improved LSTM Structure for Natural Language Processing," in *Proc. of IEEE International Conference of Safety Produce Informatization (IICSPI), Chongqing, China*, pp. 565-569, 2018. [Article \(CrossRef Link\)](#)
- [27] Nammous Mohammed. K. and Saeed, K. "Natural Language Processing: Speaker, Language, and Gender Identification with LSTM," *Advanced Computing and Systems for Security*, 143–156, 2019. [Article \(CrossRef Link\)](#)
- [28] García, J. C., and E. Serrano, "Automatic Music Generation by Deep Learning," in *Proc. of Distributed Computing and Artificial Intelligence, 15th International Conference. Advances in Intelligent Systems and Computing*, vol. 800, pp. 284-291, 2018. [Article \(CrossRef Link\)](#)
- [29] Madhok R., Goel S. and Garg, S., "SentiMozart: Music Generation based on Emotions," in *Proc. of ICAART*, pp. 501-506, 2018. [Article \(CrossRef Link\)](#)
- [30] Guangxiao Song, Zhijie Wang, Fang Han, Shenyi Ding and Muhammad Ather Iqbal, "Music auto-tagging using deep Recurrent Neural Networks," *Neurocomputing*, Vol. 292, pp. 104-110, 2018. [Article \(CrossRef Link\)](#)
- [31] Xie Y., Liang R., Liang Z., Huang C., Zou C. and Schuller B., "Speech Emotion Classification Using Attention-based LSTM," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1675-1685, 2019. [Article \(CrossRef Link\)](#)
- [32] Kang J., Zhang W.-Q., Liu W.-W., Liu J. and Johnson, M. T., "Advanced recurrent network-based hybrid acoustic models for low resource speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, 2018. [Article \(CrossRef Link\)](#)
- [33] Shota Nakayama and Shuichi Arai, "DNN-LSTM-CRF Model for Automatic Audio Chord Recognition," in *Proc. of the International Conference on Pattern Recognition and Artificial Intelligence (PRAI 2018), ACM, New York, NY, USA*, 82-88, 2018. [Article \(CrossRef Link\)](#)
- [34] Chen Z., Zhang X., Deng J., Li J., Jiang Y. and Li W., "A Practical Singing Voice Detection System Based on GRU-RNN," in *Proc. of the 6th Conference on Sound and Music Technology (CSMT). Lecture Notes in Electrical Engineering*, vol 568, pp. 15-25, 2019. [Article \(CrossRef Link\)](#)
- [35] Shu X., Tang J., Qi G., Liu W., and Yang, J.X., "Hierarchical Long Short-Term Concurrent Memory for Human Interaction Recognition," *ArXiv, abs/1811.00270*, 2018.
- [36] X. Shu, J. Tang, G. Qi, Y. Song, Z. Li and L. Zhang, "Concurrence-Aware Long Short-Term Sub-Memories for Person-Person Action Recognition," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI*, pp. 2176-2183, 2017. [Article \(CrossRef Link\)](#)
- [37] J. Tang, X. Shu, R. Yan and L. Zhang, "Coherence Constrained Graph LSTM for Group Activity Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1, 2019. [Article \(CrossRef Link\)](#)
- [38] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li and Qi Tian., "Participation-Contributed Temporal Dynamic Model for Group Activity Recognition," in *Proc. of the 26th ACM international conference on Multimedia (MM '18). ACM*, 1292-1300, 2018. [Article \(CrossRef Link\)](#)

- [39] Vahora, S. and Chauhan, N., “Deep neural network model for group activity recognition using contextual relationship,” *Eng. Sci. Technol. Int. J.* 22, 47–54, 2019. [Article \(CrossRef Link\)](#)
- [40] Toan H. Vu, An Dang, Le Dung and Jia-Ching Wang, “Self-Gated Recurrent Neural Networks for Human Activity Recognition on Wearable Devices,” in *Proc. of the on Thematic Workshops of ACM Multimedia 2017 (Thematic Workshops '17)*. ACM, New York, NY, USA, 179-185, 2017.
- [41] Yao-zhong, Zhang, Rui Yamaguchi, Seiya Imoto and Satoru Miyano, “Sequence-specific bias correction for RNA-seq data using recurrent neural networks,” *BMC Genomics* volume, 18, Article number. 1044, 2017. [Article \(CrossRef Link\)](#)
- [42] Vazhayil A., Vinaykumar R. and Soman K.P., “DeepProteomics: Protein family classification using Shallow and Deep Networks,” *ArXiv, abs/1809.04461*, 2018. [Article \(CrossRef Link\)](#)
- [43] Qiang Shi, Weiya Chen, Siqi Huang, Fanglin Jin, Yinghao Dong, Yan Wang and Zhidong Xue, “DNN-Dom: predicting protein domain boundary from sequence alone by deep neural network,” *Bioinformatics*, btz464.
- [44] Nguyen Quoc and Khanh Le, “Fertility-GRU: Identifying Fertility-Related Proteins by Incorporating Deep-Gated Recurrent Units and Original Position-Specific Scoring Matrix Profiles,” *Journal of Proteome Research*, 18 (9), 3503-3511, 2019. [Article \(CrossRef Link\)](#)
- [45] Hanson J, Yang Y, Paliwal K and Zhou Y, “Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks,” *Bioinformatics*, 33(5), 685-692, 2017.
- [46] Xueliang Leon Liu, “Deep Recurrent Neural Network for Protein Function Prediction from Sequence,” *BioRxiv*, jan 2017.
- [47] Hinkka M., Lehto T., Heljanko K. and Jung A. “Classifying Process Instances Using Recurrent Neural Networks,” *Lecture Notes in Business Information Processing*, 313–324, 2019. [Article \(CrossRef Link\)](#)
- [48] Hildebrandt T., van Dongen B. F., Röglinger M. and Mendling J., “Business Process Management,” *Lecture Notes in Computer Science*, vol. 11675, 2019. [Article \(CrossRef Link\)](#)
- [49] Nolle, T., Luetggen S., Seeliger A. and Mühlhäuser M., “BINet: Multi-perspective Business Process Anomaly Classification,” *ArXiv, abs/1902.03155*, 2019. [Article \(CrossRef Link\)](#)
- [50] C. Yan, X. Fu, W. Wu, S. Lu and J. Wu, “Neural Network Based Relation Extraction of Enterprises in Credit Risk Management,” in *Proc. of 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, pp. 1-6, 2019. [Article \(CrossRef Link\)](#)
- [51] Jagannatha, Abhyuday N. and Hong Yu., “Bidirectional RNN for Medical Event Detection in Electronic Health Records,” in *Proc. of the conference. Association for Computational Linguistics. North American Chapter*, 473-482, 2016. [Article \(CrossRef Link\)](#)
- [52] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou and Aidong Zhang, “Risk Prediction on Electronic Health Records with Prior Medical Knowledge,” in *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*, ACM, New York, USA, 1910-1919, 2018.
- [53] M. Pavithra, K. Saruladha and K. Sathyabama, “GRU Based Deep Learning Model for Prognosis Prediction of Disease Progression,” in *Proc. of 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, pp. 840-844, 2019. [Article \(CrossRef Link\)](#)
- [54] Maragatham G. and Devi Shobana, “LSTM Model for Prediction of Heart Failure in Big Data,” *Journal of Medical Systems*, 43(5), pp. 111, Mar 2019. [Article \(CrossRef Link\)](#)
- [55] Lee J. M. and Hauskrecht M., “Recent Context-Aware LSTM for Clinical Event Time-Series Prediction,” *Lecture Notes in Computer Science*, 13–23, 2019. [Article \(CrossRef Link\)](#)
- [56] Xu X., Wang Y., Jin T., and Wang J., “A Deep Predictive Model in Healthcare for Inpatients,” in *Proc. of 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2018 [Article \(CrossRef Link\)](#)
- [57] Cheng L., Ren Y., Zhang K., and Shi Y., “Medical Treatment Migration Prediction in Healthcare via Attention-Based Bidirectional GRU,” *Web and Big Data. APWeb-WAIM 2019. Lecture Notes in Computer Science*, vol. 11641, 2019. [Article \(CrossRef Link\)](#)

- [58] Che Z., Purushotham S., Cho K., Sontag D. and Liu, Y., "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Scientific Reports*, 8, pp. 1-12, 2018. [Article \(CrossRef Link\)](#)
- [59] Yu Zhu, Yu Gong, Qingwen Liu, Yingcai Ma, Wenwu Ou, Junxiong Zhu, Beidou Wang, Ziyu Guan, and Deng Cai, "Query-based Interactive Recommendation by Meta-Path and Adapted Attention-GRU," in *Proc. of CIKM '19: ACM International Conference on Information and Knowledge Management, Beijing, China. ACM, New York, NY, USA*, 9 pages, 2019.
- [60] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenberg and Jure Leskovec, "Hierarchical Temporal Convolutional Networks for Dynamic Recommender Systems," in *Proc. of the 2019 World Wide Web Conf. (WWW'19), San Francisco, CA, USA. ACM, NY, NY, USA*, 11 pages, 2019.
- [61] L. Huang, Y. Ma, S. Wang and Y. Liu, "An Attention-based Spatiotemporal LSTM Network for Next POI Recommendation," *IEEE Transactions on Services Computing*, pp.1-1, 2019. [Article \(CrossRef Link\)](#)
- [62] Douligieris C., Karagiannis D., and Apostolou D., "Knowledge Science, Engineering and Management," *Lecture Notes in Computer Science*, vol. 11776, 2019. [Article \(CrossRef Link\)](#)
- [63] Amit Livne, Moshe Unger, Bracha Shapira, and Lior Rokach, "Deep Context-Aware Recommender System Utilizing Sequential Latent Context," 2019.
- [64] Doan K.D., Yang G. and Reddy C.K., "An Attentive Spatio-Temporal Neural Model for Successive Point of Interest Recommendation," *Advances in Knowledge Discovery and Data Mining. PAKDD 2019. Lecture Notes in Computer Science*, vol. 11441, pp. 346-358, 2019. [Article \(CrossRef Link\)](#)
- [65] Hu F., Huang X., Gao X. and Chen G., "AGREE: Attention-Based Tour Group Recommendation with Multi-modal Data," *Database Systems for Advanced Applications. DASFAA 2019. Springer Lecture Notes in Computer Science*, vol. 11448, pp. 314-318, 2019. [Article \(CrossRef Link\)](#)
- [66] Huang R.N., McIntyre S., Song M., Haihong E., and Ou Z., "An Attention-Based Recommender System to Predict Contextual Intent Based on Choice Histories across and within Sessions," 8(12), 2426, 2018. [Article \(CrossRef Link\)](#)
- [67] Zheng H.T., Chen, J.Y., Liang N., Sangaiah A.K., Jiang Y., Zhao C.Z., "A Deep Temporal Neural Music Recommendation Model Utilizing Music and User Metadata," *Appl. Sci.*, 9(4), 703, 2019. [Article \(CrossRef Link\)](#)
- [68] S. Li, Z. Yan, X. Wu, A. Li and B. Zhou, "A Method of Emotional Analysis of Movie Based on Convolution Neural Network and Bi-directional LSTM RNN," in *Proc. of IEEE Second International Conference on Data Science in Cyberspace (DSC), Shenzhen*, pp. 156-161, 2017. [Article \(CrossRef Link\)](#)
- [69] Sebastian Heinz, Christian Bracher, Roland Vollgraf, "An LSTM Based Dynamic Customer Model for Fashion Recommendation," in *Proc. of Workshop on Temporal Reasoning in Recommender Systems, Como, Italy*, 5 pages, 31st August 2017.
- [70] Syed Tanveer Jishan and Yiji Wang, "Audience Activity Recommendation Using Stacked-LSTM Based Sequence Learning," in *Proc. of the 9th International Conference on Machine Learning and Computing (ICMLC 2017). ACM, New York, NY, USA*, 98-106, 2017. [Article \(CrossRef Link\)](#)
- [71] Hiroaki Sakoe and Seibi Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *Readings in speech recognition*, pp. 159-165, 1990. [Article \(CrossRef Link\)](#)
- [72] J. Aach and G.M. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, 17(6), 495-508, 2001. [Article \(CrossRef Link\)](#)
- [73] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria and Eamonn Keogh, "Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping," *ACM Trans. Knowl. Discov. Data*, 7(3), 2013. [Article \(CrossRef Link\)](#)
- [74] Y. Zhu and D. Shasha, "Warping indexes with envelope transforms," in *Proc. of SIGMOD*, pp. 181-192, 2003. [Article \(CrossRef Link\)](#)

- [75] Chotirat Ann Ratanamahatana and Eamonn Keogh, "Everything you know about dynamic time warping is wrong," in *Proc. of Third Workshop on Mining Temporal and Sequential Data*, 2004.
- [76] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning," *MIT Press*, 2016.
- [77] A. Greenstein-Messica, L. Rokach, and M. Friedman, "Session-based recommendations using item embedding," in *Proc. of IUI '17. ACM*, pp. 629–633, 2017. [Article \(CrossRef Link\)](#)
- [78] B. Hidasi, A. Karatzoglou, L. Baltrunas and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. of ICLR '16*, 2016.
- [79] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. of RecSys '16., ACM*, 241–248, 2016. [Article \(CrossRef Link\)](#)
- [80] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proc. of DLRS '16. ACM*, pp. 17–22, 2016. [Article \(CrossRef Link\)](#)
- [81] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. of KDD '15. ACM*, pp. 1235–1244, 2015. [Article \(CrossRef Link\)](#)
- [82] C. Wu, J. Wang, J. Liu and W. Liu, "Recurrent neural network based recommendation for time heterogenous feedback," *Knowl-Based System*, 109, 90–103, 2016. [Article \(CrossRef Link\)](#)
- [83] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola and H. Jing., "Recurrent recommender networks," in *Proc. of WSDM '17. ACM*, 495–503, 2017.
- [84] Donkers Tim, Loepp Benedikt and Ziegler Jürgen, "Sequential User-based Recurrent Neural Network Recommendations," in *Proc. of RecSys '17, ACM*, pp. 152–160, 2017. [Article \(CrossRef Link\)](#)
- [85] Sun Yu, Zhao Peize and Zhang Honggang, "TA4REC: Recurrent Neural Networks with Time Attention Factors for Session-based Recommendations," in *Proc. of IJCNN*, 1–7, 2018. [Article \(CrossRef Link\)](#)
- [86] Liu Q., Wu S., Wang D., Li Z. and Wang L., "Context-aware sequential recommendation," in *Proc. of ICDM 2016*, pp. 1053–1058, 2016. [Article \(CrossRef Link\)](#)
- [87] Elena Smirnova and Flavian Vasile, "Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks," in *Proc. of ACM Recommender Systems conference, Como, Italy, (RecSys '17), ACM*, pp. 2–9, 2017. [Article \(CrossRef Link\)](#)
- [88] Liu Q., Wu S., Wang L., Tan T., "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. of AAAI 2016*, pp. 194–200, 2016.
- [89] Liu J., Wang G., Hu P., Duan, L.-Y. and Kot A. C., "Global context-aware attention lstm networks for 3d action recognition," in *Proc. of CVPR*, 2017. [Article \(CrossRef Link\)](#)
- [90] S. Ilarri, R. Trillo-Lado and R. Hermoso, "Datasets for Context-Aware Recommender Systems: Current Context and Possible Directions," in *Proc. of IEEE 34th International Conference on Data Engineering Workshops (ICDEW), Paris*, pp. 25–28, 2018. [Article \(CrossRef Link\)](#)
- [91] Košir LDOS-CoMoDa dataset 2011.
- [92] Mingang Chen, Pan Liu, "Performance Evaluation of Recommender Systems," vol. 13, no. 8, pp. 1246–1256, 2017.
- [93] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto and H Chi, "Latent Cross: Making Use of Context in Recurrent Recommender Systems," in *Proc. of WSDM*, pp. 46–54, 2018. [Article \(CrossRef Link\)](#)
- [94] Lian J., Zhang F., Xie X. and Sun G., "CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems," in *Proc. of WWW*, pp. 817–818, 2017. [Article \(CrossRef Link\)](#)
- [95] M. Braunhofer, M. Elahi and F. Ricci, "STS: A context-aware mobile recommender system for places of interest," in *Proc. of 22nd International Conference on User Modeling Adaptation and Personalization (UMAP), CEUR Workshop Proceedings*, pp. 75–80, 2014.
- [96] D. J. Weller-Fahy, B. J. Borghetti, A. A. Sodemann, "A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2015. [Article \(CrossRef Link\)](#)



Mrs Kala K. U. pursued Bachelor of Science in Information Technology from the University of Calicut, Kerala, India in the year 2008 and Master of Computer Applications from University of Calicut in the year 2012. She has qualified UGC NET in Computer Science and Applications in July 2016 and currently pursuing Ph.D in Computer Science and Engineering in the Department of Computer Science, Pondicherry University, India since 2016. Her main research work focuses on Machine Learning, Deep Learning, Knowledge engineering and Recommender Systems. She has 3 years of teaching experience and 2 years research experience.



Dr M. Nandhini pursued Bachelor of Science in Mathematics, 1994 and Master of Computer Applications, 1997 from Bharathidasan University, Tiruchirappalli, Tamilnadu, India. She has qualified UGC NET in 1998 and pursued M.Phil in Computer Science, 2005 from Alagappa University and awarded Ph.D in Computer Science from Bharathiar University, Coimbatore in the year 2013. She is working as an Assistant Professor in the Department of Computer Science, Pondicherry University, India since 2009. She has published research papers in reputed international journals and conferences. Her main research work focuses on Soft Computing, Combinatorial Problem Optimization, Artificial Intelligence, and Data Mining. She has 21 years of collegiate teaching experience and 15 years of Research Experience.