International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Sep-2022)

https://dx.doi.org/10.12785/ijcds/120165

A Novel Approach towards Movie Recommender System using Deep Learning Techniques

Mugdha Sharma[1] and Akanksha Dhamija[2]

[1,2] Department of Computer Science, Bhagwan Parshuram Institute of Technology, Delhi, India

Received 26 Mar.2021, Revised 29 Mar. 2021, Accepted 20 May. 2021, Published 30 Sep. 2022

Abstract: Recommender systems have become a key technology to help the users in interacting with the increasingly larger data and information available online. The rapid advancements in Deep Learning techniques have been very useful in recommendation systems as it enhances the overall performance and accuracy of the recommendation systems. This paper attempts to work on a hybrid recommendation model by considering a weighted average of top N recommendations from both content based and collaborative based filtering methods and hence eliminating their individual shortcomings. A LightFM module has been also used to evaluate the loss functions on this hybrid model and to capture the latent features about attributes of users and items. Thereafter, a class of two-layer undirected graphical models, called Restricted Boltzmann Machine (RBM) and Auto-encoder is successfully applied to the Movielens data set to provide the accurate recommendations. This study shows that the proposed approach outperform the traditional recommender systems in terms of accuracy.

Keywords: Recommender System, Movies, Hybrid Algorithm, Deep Learning, RBM, Auto Encoders.

1. Introduction

In today's digital world, one of the challenging tasks is to find the appropriate information from the vast amount of data that is available online. There are various personalization techniques available in the market to overcome this problem; and recommendation system is one of the powerful tools in this era of information outburst which provides the relevant information according to user's interest [1]. The two main recommender algorithms are Content-based recommendation (CB) and collaborative filtering recommendation (CF). Based on the similarity score of the item description and profile of user's interest, CB provides the recommendations [2]. The CF method produces recommendations on the basis of users of similar taste [3]. But they suffer from two major problems a) cold start problem when there is no previous information available about the users and items b) the problem of sparsity. To overcome such problems there are various approaches such as matrix factorization using Singular Value Decomposition (SVD), SVD++, hybrid model and machine learning method.

A lot of research in the area of recommender system has been done through implementing deep learning concepts by deep neural networks and has resulted in some rapid advances in the field of Artificial Intelligence. Deep learning is a broader domain of machine learning aimed at recognizing patterns or feature extraction at an abstraction level running on huge neural networks allows modeling sparse data on a cluster efficiently. Research on new topologies for neural networks in order to deduce the fresh insights have outperformed the traditional approaches of recommender algorithm by providing better recommendations. One such approach for accomplishing this is through Restricted Boltzman Machine (RBM) and Auto-Encoders.

RBM is a generative approach of probabilistic distribution with a bi-layer structure which iteratively performs forward and backward passing during each epoch to reconstruct an efficient prediction. Integrating matrix factorization techniques with RBM have always been observed to deliver the better results. Auto-encoders being an extended version of RBM add a set of weights along with RBM functionality.

E-mail: mugdha.sharma145@gmail.com, akankshadhamija12@gmail.com

The collective hybrid model can also be trained to model based on ranking loss functions using a LightFM library. The results of the parameters used in LightFM are subjected to deep learning models.

Based on the concepts explained above, the overall delineated model for this proposed research work is as follows:

- To implement a recommendation approach for a real time movie dataset by organizing the data into some hyper-parameters in an attempt to make efficient and accurate predictions.
- To develop the accuracy matrix for comparing the existing approaches of recommender system
- To extract the latent features characterizing user and item attributes and regularizing in the form of matrix factorization techniques by evaluating user-item relationships to solve some problems like data sparsity and cold start problems.
- To Build a two-way hybrid recommender model covering important feature of various existing recommender system and analyzing their recommending capabilities and accuracy.
- To Integrating the basic hybrid model by leveraging concepts of deep learning and neural networks aimed at making the recommendations more accurate.

The remainder section of the paper is organized as follows. In section 2, an overview of the literature review has been explained in the field of recommender system and usage of deep learning with the RBM. Section 3 comprises the detailed description of the proposed hybrid model and its concepts. Section 4 and 5 summarizes the experimental description, explains the dataset and experimental results of the hybrid model with the comparison of existing recommender systems. Finally, section 6 outlines the conclusion from the proposed work and provides the direction and opportunity to the future research.

2. RELATED WORK

Deep learning architectures have drawn the interest of researchers to overcome the hand designed features of traditional algorithm. Deep Neural Network (DNN) have applied in various fields such as computer vision [4][5], speech recognition [6][7], and recommender system [8] [9][10]. Performance of the recommendation has influenced by input quality of DNN model. Ruiqin Wang et al introduced a two-stage deep learning recommender model. In first stage, marginalized stack de-noising auto encoder applied on the user and item features to learn the latent vector and in second step the resultant latent factor vector are used as the input of the DNN component and provide optimized result [11].

Kiran R. et al proposed a hybrid approach which integrates embedding, the side information of user and item with deep network, it consist of three hidden layers each layer calculates a linear function followed by a LeakyReLU followed by deep root which resolve the cold start problem and improved the values of Root mean Square Value (RMSE) and Mean Absolute Error (MAE) [12].

Defu Lian et al. used the LightFM module to implement ranking-based loss functions known as Personalized Ranking loss based on Importance Sampling (PRIS). The aim of the algorithm was to develop some series of negative samples to discover loss functions like Weighted Approximate Rank Pairwise Loss (WARP), Bayesian Personalized Ranking (BPR) and calculate the degree of their in-formativeness and approximation of accuracy [13]. These samples were subjected to five different datasets of varying size and difficulty. With the increase in adequacy of negative samples, not much negative samples of observations are required which improved the recommending performance.

Mehdi and Reza proposed an algorithm called RBMDeepNet using combination of deep convolutional deep neural network and RBM [14]. This induces a method to extract objects from airborne images of an area and recognize them to a vehicle based on geometrical or physical feature extraction (automatic car extraction or ACE). RBM helps in learning characteristics of visual objects and is integrated with the inputs of deep convolutional neural network to build a model whose ACE results are derived from some appropriate accuracy criteria. This paper has implemented an extended functionalities of SegNet and U-Net [15][16].

Wenming Cao et al. implemented a hybrid representation learning (HRL) model that works on cross modal retrieval tasks [17]. A fundamental multi-modal data is a data interconnected simultaneously to learn the unidentified relationships amongst the data attributes in a cognitive environment. Algorithms based on multimodal data can retrieve for instance, an image-text relationship or a text-audio relationship for an image representation. Another variation is a cross modal retrieval method which emphasizes on correlating amongst the modalities but tend to face some challenges in results. This paper has promised to solve this by using a stacked RBM (SRBM) and multimodal deep belief net (DBN) with a hybrid of auto-encoder and deep neural network [18]. Chih-Ming Chen et al. proposed the use of collaborative filtering technique using collaborative similarity embeddings (CSE) namely direct embedding and neighborhood embeddings to provide the better recommendations.

3. RESEARCH METHODOLOGY

The research has been conducted in a series of steps which have been explained in the below sub sections.

A. Pre-Processing

As practiced with traditional machine learning studies, the dataset on which the research is to be done is split into training set and test dataset where the analytical model of research study is fitted on the training data and the remaining test set is subjected to that model to calculate the predictions and their associated parameters. As a consequence, there are possibilities of under-fitting or over-fitting which may affect the reliability of the model. Over-fitting a model simply means any situation where the model or algorithm has been trained too well in accordance with the data that even random fluctuations in training data impacts the result but may be unnecessary in test data. On the other hand, under-fitting the model is incapable of deducing new trends in the data and is not suited for best predictions [19]. To avoid any of these cases, the machine learning concepts promise to develop an algorithm to maintain reliability as well as accuracy by implementing validations, regularization or ensemble features on the dataset.

The proposed research work implements this by using k-fold cross-validation and leave one out cross-validation (LOOCV). K Fold validation is a technique of dividing the dataset randomly into k groups of samples of equal sizes known as folds. The machine learning model is fitted onto each (k-1) sample of the fold and measured with their accuracy. Similarly, for every fold in this process, accuracy for that is evaluated resulting in an average weighted accuracy score which may be helpful in combating over-fitting problems. On the other hand, LOOCV is trained to build n models from n samples iterated k times, where n>k trained on n−1 samples (leaving one sample out) rather than (k−1) n/k proving more evaluation cost and overhead than k-fold cross-validation [20]. But in some cases where the bias-variance trade-off is related and k has to be kept small then LOOCV is preferred to k-fold. This paper proposes to use both the techniques individually and to compare accuracy results in both.

B. Recommender System

Now, focusing on the aspects of using a recommender system, one needs to be aware of how well the data for users, movies, and ratings be stored and represented while implementing an algorithm or a model. For a model-based recommender system, all the latent factors can have possible distinct values and can be applied to the model. But for a memory-based recommender system it becomes necessary to represent real-time data into a multi-dimensional matrix (2D) depicting entries of users in rows and movies in columns. This matrix representation using low-degree mathematical operations makes the analysis easier and more efficient to perform on real world data with its latent factors [21].

This means that predicting movies in a recommender system will be evaluated by performing matrix operations in the data and on further simplification of calculations, each of the user-item matrix is decomposed into the product of its constituent matrices with lower degree or dimensionality. This method is called matrix factorization. Also, there is a possibility in the matrix that not every entry of the user would have watched all movies listed in the column resulting in empty or null values. This paper deals with these issues in the user-item matrix where there is an effort to deduce those empty values by some matrix factorization techniques like SVD and SVD++.

For each matrix representation of $R_{U \times M}$, the rows of U are the left singular vectors (user vectors); $\sum$ is the diagonal matrix having the same dimensions as $R_{U \times M}$ having diagonally singular values and M_T has columns that are the right singular vectors (item/movie entry vectors). The SVD promises to signify an explanation of the original matrix where the covariance matrix is diagonal and same can be represented as shown in equation 1 below [22]:

$$R_{U \times M} = U_{U \times U} \cdot \sum\nolimits_{\boldsymbol{U \times M}} \cdot \; M^{T}_{M \times M} \tag{1}$$

The primary objective of a successful recommender system is to improve the prediction accuracy. Usually, users leave few implicit feedbacks like their browsing history, previous rating data, etc. So, the rating system reflects the actual preferences of the user for each latent vector till a certain extent. That's why, the SVD++ method introduces the implicit feedback information based on SVD. To obtain maximum accuracy, minimum squared error and avoiding over-fitting the training dataset, a regularization factor α is used along with some biases B, and the objective function of the SVD++ model is represented by the following equation:

$$\mathrm{Min}_{\,\mathrm{U, M, B}} \left(SUM_{(U, M)} \left(R_{U \times M} - U_{U \times U} . \sum\nolimits_{\boldsymbol{U \times M}} . M^{T}_{M \times M} - B\right)^2 + \boldsymbol{\alpha}[U_{U \times U}{}^{2} + M_{M \times M}{}^{2} + B^2] \right) \tag{2}$$

This leads to the analysis of SVD++ which hypothetically improved the overall performance of SVD [23].

C. Content and Collaborative based Filtering

In content-based filtering, recommendations generated on the basis of similar genres and similar year of the movies that user liked in the past. By similarity, the paper intends to find out how any two random selected movies are related to each other. This similarity has been best measured with correlation metric or similarity index. In the proposed work, content based similarity has been calculated on basis of genre and year with help of KNN algorithm, where similarity score has been calculated between the rated movie and the movie to be predicted, then select K-nearest neighbors' with highest content similarity followed by weighted average score to the movie whose rating to be predicted. The new user problem called cold start problem has been solved with the help of SVD and SVD++ module applied in previous section.

In the proposed system, collaborative filtering has been also used to generate top N recommendations by considering the ratings of similar taste user. This method is also called user-based CF. On the other hand, item-based CF is the one where ratings are being calculated on the basis of rating of similar items from the same user.

One crucial step in the CF algorithm is to calculate the similarity between items and users and finally to choose a group of nearest neighbours as recommendation partners for an active user [24]. And one of the most common similarity metrics that is used in CF is cosine similarity. But CF suffers from the problem of sparsity due lack of user's and item's rating. To avoid such problem modified cosine similarity called adjusted cosine similarity score is used for CF which can be represented as below:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}} \quad (3)$$

Here i and j are two users and $\bar{R}_u$ is the average of the *u*-th user's ratings. Cosine similarity is calculated on a scale between -1 and +1, where -1 implies the objects are completely dissimilar, +1 implies that they are totally similar and 0 implies that there is no relationship amongst the objects.

D. LightFM

The LightFM algorithm is a model for a hybrid recommender algorithm that uses both ratings of items, as well as item attributes for matrix factorization and for evaluating efficiency in the calculations [25]. This is being implemented in this paper as it outperforms collaborative and content-based models. Various explanations about ranking-based models have been already done with the evaluation of loss functions in the training model like BPR, WARP [25]. Before this, it is necessary to understand the concepts of "Area under Curve" (AUC) measuring the entire two-dimensional area underneath the entire ROC curve depicting precision and recall values in the prediction. AUC represents the probability that a random positive sample is positioned to the right of a random negative sample. It is directly proportional to the accuracy of predictions.

The probabilistic interpretation of this can be done by randomly choosing a positive case and a negative case, and the value of AUC in accordance to the given classifier determines the probability of outranking the positive case over a negative, necessarily normalized to a value of 1.

Since BPR optimized AUC, which treats all inconsistent pairs equally, BPR may not be best suitable for the top-k item recommendation. Therefore, the Weighted Approximate-Rank Pairwise (WARP) loss was proposed to optimize the precision [25, 26]. WARP also used uniform sampling with rejection to draw more informative negative samples, the score of which should be larger than that of the positive example minus one. WARP was also used for collaborative metric learning [27], leading to the state-of-the-art item recommendation method.

BPR [28] is a degree of measuring loss functions in a recommender system that randomly samples a negative item ensuring optimization by using a function like stochastic gradient descent (SGD) .WARP [26], uses the Weighted Approximate-Rank Pairwise loss function for collaborative filtering, which has better performance than BPR in some datasets. WARP focuses on an entity comprising three arguments: user, positive item and a negative item. The negative items in the entity instead of being selected randomly (as in BPR), are chosen among those which would violate the desired item ranking in the model. This approximates a form of active learning where the model selects those triplets that it cannot currently rank correctly and the learning rate is tuned in the validation set within {10^−1, 10^−2, 10^−3, 10^−4}, respectively.

E. RBM and Auto-Encoders

This paper proposes to provide a consistent performance in terms of accuracy of the recommender system. So, to ensure independence from the validations and factorization previously being done, deep learning concept is implemented using two methods used primarily for dimensionality reduction: Restricted Boltzmann constant and Auto-encoders. RBM proved to be a probabilistic generative model for presenting a Deep Learning (DL)-based hybrid model using a two-tier architectural layer with a *Relu activation* function.

On the other hand, auto-encoders use 3 layers: input layer on the bottom containing individual user input ratings, a hidden layer, and an output layer that gives us predictions by feature extraction. Additionally, prediction using auto- encoders is done using reconstructed outputs and their error reductions. A matrix of weights between the layers is maintained across every node in the neural network. The implementation of auto-encoder is a bit different but promises better results than RBM as no biases are used at each layer (unlike RBM), as hidden layer representations are much more dense and only the sigmoid activation function is required for the prediction analysis in this paper making it easier to be used in Tensorflow or Keras [29]. Learning the weights between input layer and the hidden layer is known as encoding and reconstructing predictions with the weights between the hidden layer and the output layer is decoding. The *RMSProp optimizer* is enabled to help build up the predicted ratings for every item for a given user in this research study.

F. Optimization Function

In an attempt to predict the optimum results in the recommendations, the network model is trained by choosing the weights in a way to minimize the difference between the predictions and the factors to be predicted. This can be measured by accuracy or more precisely by a loss function gradient calculation (mostly done by stochastic gradient descent) where the weights of each neuron in neural network is updated in the direction of negative descent eventually leading to the minimum weights or set of parameters.

To implement these loss functions, an optimization algorithm is used which updates the weighted parameters at each epoch. There are various optimization algorithms, but we used three different algorithms in this paper and those are: Stochastic gradient descent, RMSProp, and Adaptive Moment Estimation (Adam) optimizer. SGD is one of the earliest and most commonly used optimization algorithms, but it might take significantly longer than with some of the optimizers. In contrast, RMSprop can deal with its radically diminishing learning rates. That's why it is also used in the proposed study. Adam adds bias-correction and momentum to RMSprop. RMSprop and Adam are very similar algorithms that do well in similar circumstances [30]. Bias-correction helps Adam slightly outperform RMSprop towards the end of optimization as gradients become sparser.

Their complexity differs in terms of their efficiency to reach the global optimum in the graph and calculating the sum of squared descents. The value of hyper-parameters are set to β_1=0.9, β_2= 0.999 and learning rate= 0.001- 0.0001 [21].

4. Experimental Setup

This section covers the brief description of the dataset chosen for the experiment followed by the algorithm for the proposed study and evaluation criteria for the proposed system. The results from this experimental setup have been discussed and compared with the two existing recommender systems in the next section.

A. Dataset

Group lens research group has developed the Movielens dataset for the researchers working in the area of recommendation and predicting rating for given users [31]. In the proposed system, Movielens 100k data set consisting of 943 users, 1,682 movies and 100,000 ratings is used to estimate the performance. Basic description of the dataset components is mentioned in table 1 below:

TABLE I. DESCRIPTION OF THE DATASET

Notation Symbol	Description
U	Users
I/M	Movies
R	Ratings
Genre	G
Year	Year
P_R	Predicted rating
U_R	User rating
n_U	Number of users
H	Number of hits
$Rank_I$	Rank of each item
S	Average similarity
D	Diversity
I	Number of genres
cosSim(m,n)	Cosine similarity of movies m and n based on genre i
m	Depicting User 1
n	Depicting User 2
m_{avg}, , n_{avg}	Average of user m's and n's rating respectively
R_{avg}	Average of movie's rating
A	Regularization factor
B	Bias for SVD

B. Algorithm for the Proposed Approach

1. Start
2. Identify the list of users (U), items or movies (I/M) with their genres (g) and the ratings (R) of the existing viewers along with their IDs.
3. Pre-process the data using K-Fold Cross Validation and LOOCV.
   a. K-fold Cross validation is responsible for splitting training data in number of K-folds and test and validate the proposed model. In our recommendation system, we use each fold to train the recommendation system independently and then measure the accuracy against the test set.
   b. With the help of LOOCV, we test our recommendation system for its ability to test the top-N list for the user that is left out from our training data.
4. Fit a model on the training set and evaluate it on the test set
5. Categorize the data into a two dimensional matrix Mx(R) containing the attribute users ID (U) in the rows and attribute Movie ID/name (M) in the column and the entries comprises of the ratings for each of the corresponding and column.
6. For any user ID (U) in the matrix Mx(R):

a. There may exist entries where the values of Mx[Uy][Mz] may be null signifying that User 'y' has not recommended any movie Mz

b. Apply the PCA algorithms on the new matrix Mx(U) containing entries of users as rows and genres (g) as columns

c. Apply matrix factorization techniques like SVD and SVD++.Then evaluate and compare all their accuracy metric.

$$R = U \sum I^T$$

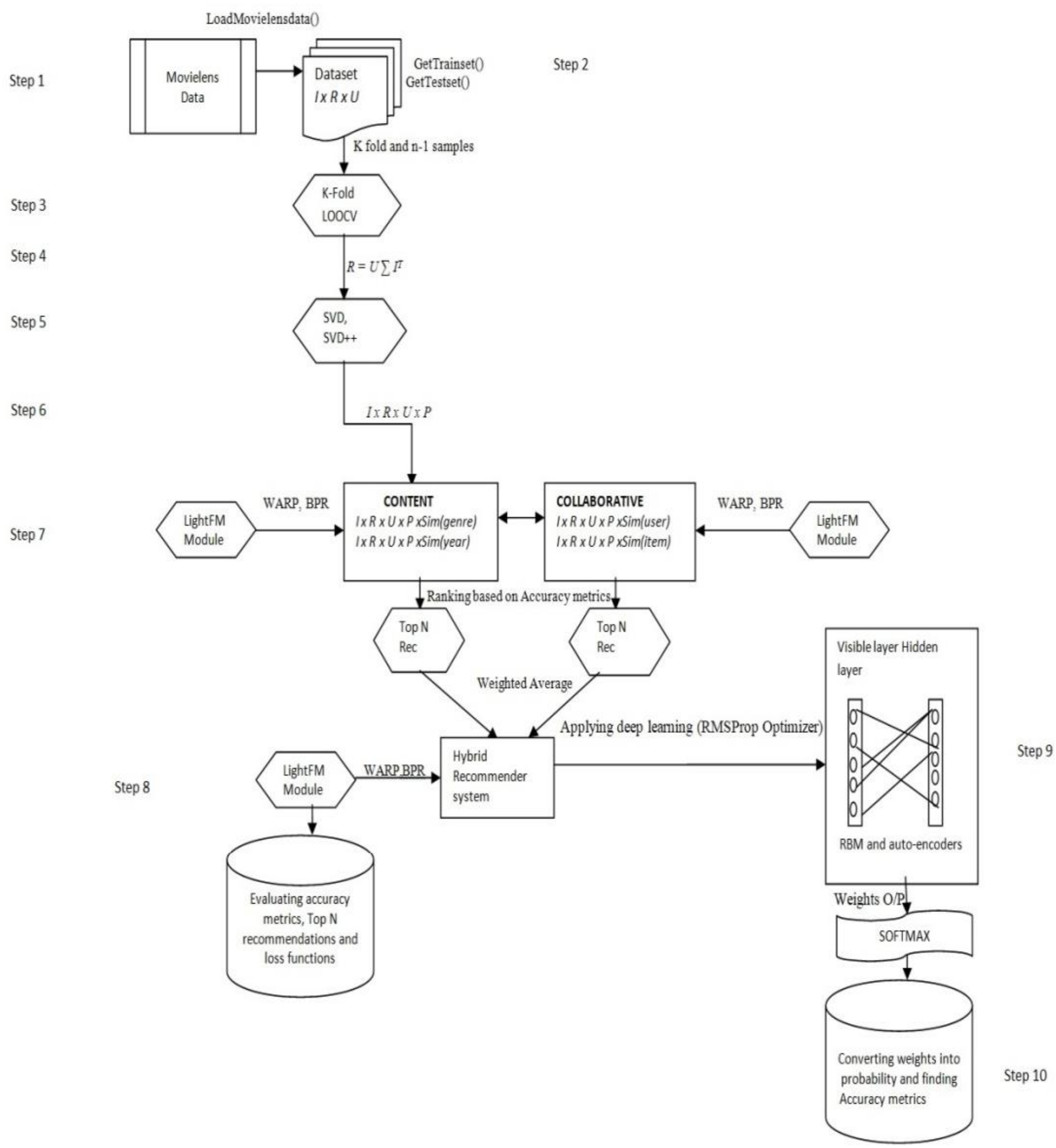

Figure. 1. Flow Chart of the Proposed Approach.

7. Develop an estimate function for implementing the hybrid recommender system constituting the following steps:
   a. Implementing Content based Recommender system using KNN Baseline Algorithm and calculating similarities on the basis of Genre(g) and Year(year)

KNN SampleTopNRecs() U x I x R----------→Sim(g) x Sim(year) x R--------→Top N Recommendation

Where Sim(g) and Sim(year) represents similarity scores based on genres and year respectively

   b. Implementing Collaborative based Recommender system using KNN Baseline Algorithm and calculating similarities on the basis of User based and Item based Fitering

KNN SampleTopNRecs()U x I x R---------→ CosSim(U) x CosSim(I) x R -------→ Top N Recommendation

Where CosSim(U) and CosSim(I) represents adjusted similarity scores based on user and item preferences respectively.

   c. Add both the algorithms in a hybrid form of recommendation with the AddAlgorithm() under the AlgoBase Class of Surprise library and evaluate the results.

8. Assessing loss function using LightFM and calculating AUC values for all the function curves.
9. Implementing layer-based RBM and Auto-encoder algorithms for analysis of effect of accuracy using deep learning and thus summarizing all the results in a consolidated table.
10. End

The whole proposed approach has been conducted in a series of steps which have been explained in the flow chart depicted in Fig. 1.

C. *Performance Metric*

Following performance metrics are used to evaluate the performance of the proposed approach:

1) Hit Rate

Hits are considered to be the rating provided by the user to one of the movies from top n –recommendation. The first step in hit rate computation for any user is to obtain all the items from the training data set of that user's history. Then one of item is removed using Leave-One-Out cross-validation. Then all the other items are used by the recommender system to provide top N recommendations. If the removed item is recommended by the system, then it's a hit and if not, then it's not a a hit. And it is represented as:

$$Hit\ Rate = \frac{\sum Hits\ from\ top\ N\ Recommendation}{Total\ number\ of\ Users} \qquad (4)$$

2) Cumulative Hit Rate

It is similar to the hit rate but here we need to add a threshold value of the rating to consider the movies that user actually likes. For example, we can ignore the predicted ratings lower than 4, to compute the cumulative hit rate for the ratings greater than or equal to 4.

3) Average Reciprocal of Hit Rate

It calculates the sum of reciprocal rank of each hit provided by the user divided by number of users [32]. It depends on the rank of the recommendations are being displayed to the user. It is represented as:

$$\text{ARHR} = 1/n \sum_{\text{i}=1}^{h} 1/p_i \qquad (5)$$

Where h is the number of hits, p_i is the position of the item in the ranked recommendation list for the i-th hit, and n is the number of users.

4) Coverage

It is the percentage of possible recommendations a proposed system is able to provide to the individual users above the threshold value and summing them divided by the number of users [33]. It is represented as:

$$\text{Coverage} = |Ip|/|I| \qquad (6)$$

Where I is the set of available items and Ip is the set of items for which a prediction can be made.

5) Diversity

It describes how broad variety of item our proposed system is suggesting to the user [34].. It is represented as:

$$Diversity = \frac{1}{2}\sum_{i_j \in u}\sum_{i_k \in u} \text{sim}(i_j, i_k) \qquad (7)$$

sim(i_j, i_k) is the similarity measure between two item i_j and i_k commonly rated by the user u.

6) Novelty

It refers to how different item has been recommended to the each user with respect to the previously seen or known item [34]. It is represented as:

$$\text{Novelty} = U_x/U_i \qquad (8)$$

5. EXPERIMENTAL RESULTS

Both SVD and SVD++ techniques are experimented on the dataset characterized by latent factors illustrated in the previous section and comparison between their performance metric have been depicted in Table 2 below. It can be observed that SVD provides a lower-dimensional reduction of the rating matrix and identifies the relationship between their latent factors in any user-item matrix. SVD++ proves to be better than SVD in terms of additional bias being used to improve the performance in a given dataset. With the additional bias, the stochastic descent is calculated for every item and user with the aim to get the optimized desirable accuracy.

TABLE II. PERFORMANCE METRIC

	SVD	SVD++
RMSE	0.9034	0.8943
MAE	0.6978	0.6887
HR	0.0298	0.0345
CHR	0.029	0.0345
ARHR	0.0112	0.0115
Coverage	0.9553	0.9768
Diversity	0.0445	0.0719
Novelty	491.5768	557.8365

Table 3 shows the RMSE, MAE values for the models applied with the content-based algorithm, a collaborative based algorithm and deep learning-based techniques to measure the values when RBM and Auto-encoder are used in the system. The values are evaluated on the basis of the type of class taken with their similarity scores amongst the movies which define whether it's user-based or item-based, whether genre-based or year-based or whether a hybrid model of the recommender system would be used.

TABLE III. PERFORMANCE COMPARISON

Metrics		RMSE	MAE
Content Based on	**Genre**	0.9552	0.9375
	Year	0.9626	0.7263
Collaborative Based on	**Item**	0.9995	0.7798
	User	0.9961	0.7711
Hybrid		0.9953	0.7124
RBM		1.1897	0.9935
AutoEncoder		1.8253	1.4222

RBM and Auto-encoder identifies the values based on an optimization function which will be discussed and analyzed in subsequent results. The results in Table 4 are calculated by using optimizers and evaluating loss function values ideally analyzed by ROC Curves. The comparison of content, collaborative, and a hybrid model of recommender system with these loss functions signifies that a hybrid one has the maximum value for each loss function. WARP shows uniform sampling and produces efficient and optimized results than BPR. The area under the curve is also shown.

TABLE IV. FINAL RESULTS

	Content	Collaborative	Hybrid
BPR	0.713	0.519	0.8766
WARP	0.673	0.4962	0.8604
AUC	0.66	0.43	0.71

The subsequent graphs have been designed to study the effect of a few parameters on the dataset based on batch size and then on epochs.

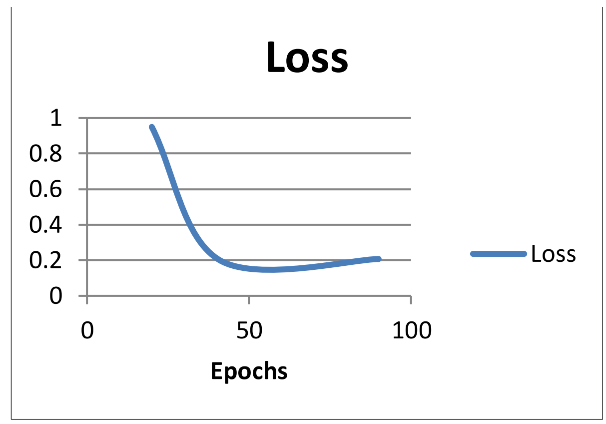

Figure 2. Loss

Fig 2 depicts the relationship between loss and epochs with the given dataset. It was observed that with the increasing number of epochs, the loss value is decreased.

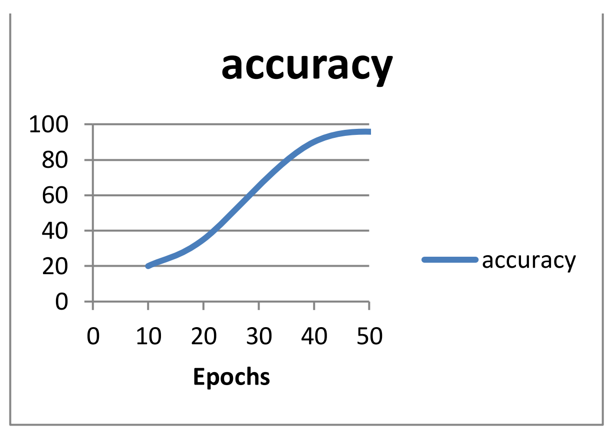

Figure 3. Accuracy

Fig 3 shows the direct proportionality of accuracy with epochs at the x-axis.

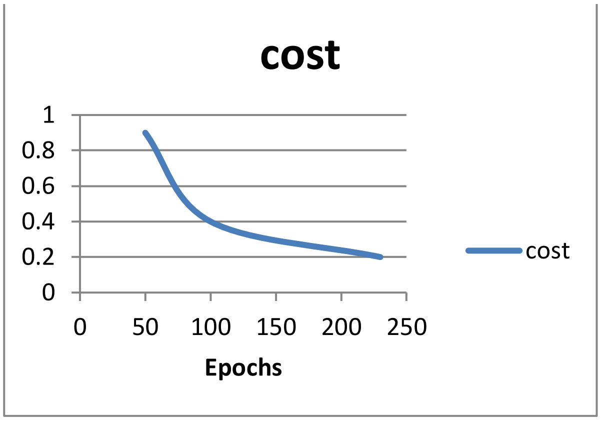

Figure 4. Cost

Fig 4 represents a decreasing parabolic relationship between the cost of implementing a hybrid model with respect to the epochs count.

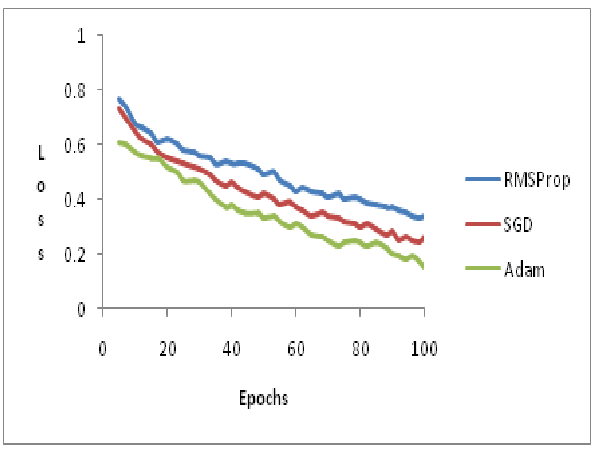

Figure 5. Comparative analysis of losses

Finally, fig 5 illustrates the comparative analysis of losses using optimization functions used in this paper (RMSProp, Adam and Stochastic Gradient Descent). It can be observed that the performance of SGD is better than RMSProp and Adam proves to be the best among these.

6. Conclusion

This research study focuses on building a hybrid movie recommender system by implementing deep learning techniques. From analysis of data and evaluation of the proposed model, the study constitute of various sections like literature review, research methodology, experimental setup, and evaluation of the proposed model. The proposed hybrid movie recommendation model is implemented with the help of LightFM module for an approximation of loss functions. The metrics evaluated under this are Area Under curve, BPR, WARP. Movielens dataset has been used to implement the movie recommendation model and to deal with this type of huge dataset; a Deep learning technique of multi-layer RBM and auto-encoders is used in the proposed study. And it is observed from the experimental results that by using deep learning concepts, the focus shifted from acquiring optimum accuracy to having good recommendations as RBM and auto-encoder produced better recommendations when compared with already accomplished approaches. Thus, this proposed study has explained the basic concepts of recommender systems and provided a path towards development of an efficient hybrid movie recommender system.

Considering the real-world challenges for the recommendation systems, like data sparsity, cold start problem, Random Exploration, etc, implementation of different recommendation machine learning models could be possible future improvement in this recommendation model. It might improve the RMSE and MAE values for better accuracy results and final recommendation.

Acknowledgment

We would like to thank Movielens for providing free database to build the efficient model. We thank Management, Principal (Prof.) Payal Pahwa and HOD(CSE) Dr. Achal Kaushik of Bhagwan Parshuram Institute of Technology for promoting research and providing valuable support in the Institute which led us to achieve great heights.

References

[1] M. Sharma, L. Ahuja and V. Kumar, “Study and Classification of Recommender Systems: A Survey.” in: Gupta D., Khanna A., Bhattacharyya S., Hassanien A.E., Anand S., Jaiswal A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, 2020, vol 1165. Springer, Singapore.

[2] N. Koubai, “MyRestaurant: A Smart Restaurant with a Recommendation System.” International Journal of Computing and Digital Systems, 2019, Vol. 8, pp. 143-156.

[3] I. Burman, S. Som, and Hossain Syed Akhter, “Model for Analyzing Psychological Parameters Recommending Student Learning Behavior.” International Journal of Computing and Digital Systems, 2020, Vol. 10, pp. 2-17.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge” International Journal of Computer Vision, 2015, Vol. 3, pp. 211–252.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[6] L. Lu, L. Kong, C. Dyer, N. Smith and S. Renals, “Segmental Recurrent Neural Networks for End-to-end Speech Recognition”, in Interspeech, 2016, pp. 385-389.

[7] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio and A. Courville, “Towards end-to-end speech recognition with deep convolutional neural networks”, in Interspeech, 2016, pp. 410-414.

[8] H. Wang, N. Wang and D. Yeung, D., “Collaborative Deep Learning for Recommender Systems.”, KDD, 2015.

[9] L. Sheng, K. Jaya and F. Yun Fu, "Deep Collaborative Filtering via Marginalized Denoising Auto-encoder.", in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), Association for Computing Machinery, New York, NY, USA, 2015, pp. 811–820.

[10] S. Zhang, L. Yao and X. Xu, "AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders.", in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17), Association for Computing Machinery, New York, NY, USA, 2017, pp. 957–960.

[11] R. Wang, H. Cheng, Y. Jiang and J. Lou, "TDCF: A Two-stage Deep Learning based Recommendation Model.", Expert Systems with Applications, Vol. 145, 2020.

[12] R. Kiran, P. Kumar and B. Bhasker, "DNNRec: A Novel Deep Learning based Hybrid Recommender System.", Expert Systems with Applications, Vol. 144, 2019.

[13] D. Lian, Q. Liu and E. Chen, "Personalized Ranking with Importance Sampling.", in Proceedings of The Web Conference 2020 (WWW '20), Association for Computing Machinery, New York, NY, USA, 2020, pp. 1093–1103.

[14] M. Khoshboresh Masouleh and R. Shah-Hosseini, "A hybrid deep learning–based model for automatic car extraction from high-resolution airborne imagery.", Appl Geomat, 2020, Vol. 12, pp. 107–119.

[15] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation.", IEEE Trans Pattern Anal Mach Intell, 2017, Vol. 39, pp. 2481–2495.

[16] O. Ronneberger, P. Fischer and T. Brox, "U-net: convolutional networks for biomedical image segmentation.", Lect Notes Comput Sci Subser Lect Notes, Artif Intell Lect Notes Bioinforma, 2015, pp. 234–241.

[17] C. Wenming, Q. Lin, Z. He, and Z. He, "Hybrid Representation Learning for Cross-modal Retrieval.", Neurocomputing, 2019.

[18] N. Zhang, S. Ding, H. Liao and W. Jia, "Multimodal correlation deep belief networks for multi-view classification.", Applied Intelligence, 2018.

[19] B. Ghojogh and M. Crowley, "The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial.", 2019.

[20] J. Yuan, Y. M. Li, C. L. Liu and X. F. Zha, "Leave-One-Out Cross-Validation Based Model Selection for Manifold Regularization." In: Zhang L., Lu BL., Kwok J. (eds) Advances in Neural Networks - ISNN 2010. ISNN 2010. Lecture Notes in Computer Science, Vol. 6063, Springer, Berlin, Heidelberg.

[21] F.O. Isinkaye, Y. Folajimi and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation." Egyptian Informatics Journal, 2015, Vol.16, pp. 261-273.

[22] M. G. Kim and K. J. Kim, "Recommender Systems using SVD with Social Network Information.", Journal of Intelligence and Information Systems, 2016, Vol. 22, pp. 1-18.

[23] X. Zhou, J. He, G. Huang and Y. Zhang, "SVD-based incremental approaches for recommender systems.", Journal of Computer and System Sciences. 2014.

[24] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Know.-Based Syst, 2014, Vol. 56, pp. 156–166.

[25] M. Kula, "Metadata Embeddings for User and Item Cold Start Recommendations.", In Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, 2015, pp. 16-20.

[26] J. Weston, S. Bengio and N. Usunier, "Large scale image annotation: learning to rank with jointword-image embeddings.", Machine learning, 2010, Vol. 81 issue 1, pp. 21–35.

[27] C. K. Hsieh, L. Yang, Y. Cui, T. Y. Lin, S. Belongie and D. Estrin, "Collaborative metric learning.", In Proceedings of WWW'17 International World Wide Web Conferences Steering Committee, 2017, pp. 193–201.

[28] H. F. Yu, M. Bilenko and C. J. Lin, "Selection of negative samples for one-class matrix factorization.", In Proceedings of the 2017 SIAM International Conference on Data Mining. SIAM, 2017, pp. 363–371.

[29] A. De. Giorgio, "A study on the similarities of Deep Belief Networks and Stacked Autoencoders.", 2015.

[30] D. P. Kingma, and J. L. Ba, "Adam: a Method for Stochastic Optimization.", International Conference on Learning Representations, 2015, pp. 1–13.

[31] N. A. ALRossais and D. Kudenko, "ISynchronizer: A Tool for Extracting, Integration and Analysis of MovieLens and IMDb Datasets," in Proc. Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization' 18, 2018, pp. 103-107.

[32] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms.", ACM Transactions on Information Systems, 2004, Vol. 22, pp. 143-177.

[33] J. Herlocker, J. Konstan, L. Terveen and J. Riedl, "Evaluating collaborative filtering recommender systems.", ACM Transactions on Information Systems, 2004, Vol. 22 issue 1, pp. 5–53.

[34] G. Asela and G. Shani. "A survey of accuracy evaluation metrics of recommendation tasks." Journal of Machine Learning Research Dec 2009, Vol. 10, pp. 2935-2962.

Dr. Mugdha Sharma is currently working as Assistant Professor in Department of Computer Science Engineering at Bhagwan Parshuram Institute of Technology, Guru Gobind Singh Indraprastha University of New Delhi, India since 2014. She has more than 7 years of teaching experience. She has published several research papers in reputed international journals and conferences including IGI Global, IEEE, ACM, SPRINGER, ELSEVIER during the tenure of her research experience. Her primary research domain is Data Mining and Recommender Systems.

Akanksha Dhamija is designated as an assistant professor in Computer Science Engineering at Bhagwan Parshuram institute of Technology, affiliated to Guru Gobind Singh Indraprastha University, New Delhi. Holding above 2.5 years of experience, she has published several research papers primarily in the field of IoT, Machine learning, Artificial Intelligence, Security and Forensics.