Received: March 22, 2022. Revised: July 5, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Optimized Feature Selection Method for E-Learning Recommender System Using Deep Neural Network based on Multilayer Perceptron

Poornima Krishnamoorthy Balasamy[1]* **Kangaiammal Athiyappagounder[2]**

[1]*Department of Computer Science,*
Government Arts College (Autonomous), Salem, Tamilnadu, India
[2]*Department of Computer Applications,*
Government Arts College (Autonomous), Salem, Tamilnadu, India
* Corresponding author's Email: bkpoornimoorthy23@gmail.com

Abstract: An effective E-Learning system must identify learning content appropriate for the needs of the specific learner from among the many sources of learning content available. The learning content can consist of different types of media content namely Text, Video and Image and to determine a rating. Recommendation systems become increasingly used in information systems and especially in e-learning platform. These systems are used to propose and recommend content of these platforms to users according to needs of the later in order to allow them to have the maximum information for learning. In this paper, a recommendation system based on data mining and deep learning has been proposed to help the learner by recommending the appropriate learning content. To improve the recommendations a content matrix is created and based on this, the logistic regression and deep learning methods could classify recommendations. Experimental results show that the precision, recall and f measure for Top 5 & 10 Recommendations using Deep Neural Network performsby 1.439% better than Top 5 & 10 Recommendations using Logistic Regression among Chi Squared, SA, PSO and ICA methods.

Keywords: Recommender system, E-learning, Feature selection, Deep learning, Multilayer perceptron (MLP).

1. Introduction

Nowadays, new technologies and the fast growth of the Internet have made access to information easier for all kind of people, raising new challenges to education when using internet as a medium. One of the best examples is how to guide students in their learning processes. The need to look for guidance from their teachers or other companions that many Internet users experience when endeavour to choose their reading, exercises for practises is a very common reality. In addition, distance learning through e-Learning platform is now being increasingly used. In order to cater for this need many different information and recommendation strategies have been developed. Recommendation systems is one of these. Recommendation systems try to help the user, presenting him those objects the learner could be more interested in, based on his known preferences or on those of other users with similar characteristics [1].

Recently, different recommendation techniques in e-learning have been designed that are helpful to both the learners and the educators in a wide variety of e-learning systems. Customized learning, which requires e-learning systems designed based on educational experience that suit the interests, goals, abilities, and willingness of both the learners and the educators, is required in some situations. One common approach to building accurate recommender models is collaborative filtering (CF). CF is a method of making predictions about an individual's preferences based on the preference information from other users. CF has been shown to work well across various domains, and many successful web-services such as Netflix, Amazon and YouTube use CF to deliver highly personalized recommendations to their

users. Recommender systems suggest items by exploiting the interactions of the users with the system (e.g., the choice of the movies to recommend to a user is based on those s/he previously evaluated). In particular, content-based systems suggest items whose content is similar to that of the items evaluated by a user [2].

In general, recommendation lists are generated based on user preferences, item features, user-item past interactions and some other additional information. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system and hybrid recommender system based on the types of input data [3]. Collaborative filtering makes recommendations by learning from user-items like historical interactions, either explicit (e.g. user's previous ratings) or implicit feedback (e.g. browsing history). Content-based recommendation is based primarily on comparisons across items and users auxiliary information. A diverse range of auxiliary information such as texts, images and videos can be taken into account. Hybrid model refers to recommender system that integrates two or more types of recommendation strategies [4].

Deep learning can be generally considered to be sub-field of machine learning. The typical defining essence of deep learning is that it learns deep representations, i.e., learning multiple levels of representations and abstractions from data. Neural architectures have demonstrated tremendous success in both supervised and unsupervised learning tasks [5]. Multilayer perceptron (MLP) is a feed-forward neural network with multiple (one or more) hidden layers between the input layer and output layer. Here, the perceptron can employ arbitrary activation function and does not necessarily represent strictly binary classifier. MLPs can be interpreted as stacked layers of nonlinear transformations, learning hierarchical feature representations. MLPs are also known to be universal approximations.

Recently, deep learning has been revolutionizing the recommendation architectures dramatically and brings more opportunities to improve the performance of recommender. Recent advances in deep learning based recommender systems have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality. Deep learning is able to effectively capture the non-linear and non-trivial user-item relationships. Thus, in this work, deep learning method, an MLP-feed forward 4 layer, is proposed for finding the best recommendation. The rest of the paper is organized into four section, related works, methodology, results & discussion and conclusion.

2. Related works

Shishehchi et al [6] proposed a semantic recommender system for e-learning by means of which, learners will be able to find and choose the right learning materials suitable to their field of interest. The proposed web-based recommendation system comprises ontology and web ontology language (OWL) rules. Rule filtering will be used as a recommendation technique. The proposed recommendation system architecture consists of two subsystems; semantic based system and rule based system. Modules for either subsystems are: observer, learner profile, recommendation storage and user interface.

Tan et al., [7] the user-based collaborative filtering method is chosen as the primary recommendation algorithm, combined with online education. We analyze the requirement of a web-based e-learning recommendation system, and divide the system workflow into five sections: data collection, data etl, model generation, strategy configuration, and service supply. Moreover, the architecture is proposed based on which further development can be accomplished. In this architecture, there are seven modules, and four of them are core modules: recommendation models database, recommendation system database, recommendation management, Data/Model management.

Bhaskaran et al., [8] developed an intelligent recommender using split and conquer strategy-based clustering that can adapt automatically to the requirements, interests, and levels of knowledge of the learners. The recommender analyzes and learns the styles and characteristics of learners automatically. The different styles of learning are processed through the split and conquer strategy-based clustering. The proposed cluster-based linear pattern mining algorithm is applied to extract the functional patterns of the learners. Then, the system provides intelligent recommendations by evaluating the ratings of frequent sequences. Experiments were conducted on different groups of learners and datasets, and the proposed model suggested essential learning activities to learners based on their style of learning, interest classification, and talent features. It was experimentally found that the proposed cluster-based recommender improves the recommendation performance by resulting in more lessons completed when compared to learners present in the no-

Table 1. List of attributes

User_id
Gender
Exp
Page content
page_id
time_on_page_second
Content_on_page
No. of time page_id_visited
Page Rating

recommender cluster category. It was found that more than 65% of the learners considered all criteria to evaluate the proposed recommender.

Kulkarni et al., [9] reviewed the main paradigms of recommender systems using explicit and implicit feedback and also the various methodologies that have been implemented to design recommender systems to enhance learning. The authors summarized the concepts of eLearning, recommendation systems and deep learning. E-learning recommendation systems will be useful to enhance learning.

A recommendation framework for E-learning based on deep learning is proposed by Wang et al., [10]. The proposed model is based on deep learning, which has strong capability to learn from train set. It has some improvements than previous methods. First, it is based on the conventional K-nearest neighbor (KNN) method to train a model, thus its accuracy is guaranteed. Second, it can recommend the new item whose similarity cannot be calculated. Third, it greatly reduces the heavy burden for a running system, which is useful in real practice of recommendation systems. In conclusion, the framework can offer a new recommendation method for more personalized learning in the future.

Zarzour et al., [11] proposed a novel approach called RecDNNing with a combination of embedded users and items combined with deep neural network. The proposed recommendation approach consists of two phases. In the first phase, a dense numeric representation for each user and item is created, called user embedding and item embedding, respectively. Following that, the items and users embedding are averaged and then concatenated before being fed into the deep neural network. In the second phase, the model of the deep neural network is used to take the concatenated users and items embedding as the inputs in order to predict the scores of rating by applying the forward propagation algorithm. The experimental results on MovieLens show that the proposed RecDNNing outperforms state-of-the-art algorithms.

As observed from the reviews of the works available in the literature, content matrix, deep learning methods are used in recommender systems. It is also seen that the features of the content is considered as whole. To improve the recommendations, feature selection is proposed in this work.

3. Methodology

The dataset is obtained from an organization which has got registered with all India council of Technical Education for research purposes. The data collected is compiled with attributes given in Table 1. With this data, the recommender system is framed as trying to predict one value of the event given the others. For example, for a tuple $e = (i, j, R)$, use (i, j) predict R. From a machine learning perspective, split the tuple e into features x and label y such that $x = (i, j)$ and label $y = R$. It can be re-framed the recommendation problem as predicting which content a user will need to learn at a given time by defining $x = (i,t)$ and $y = j$. Depending on if the label is categorical, rating, the machine learning problemis either a classification or regression problem, respectively.

Fig. 1 shows the overall process of recommender system. It assists in build the center of interest for users by facilitating access to content. The system is divided into four main parts: The first part for data collection where learners's data is collected and existing page ranking is collected. The second part for processing of information already collected in the previous section forming a content matrix. The third part deals with feature selection from content matrix. In the fourth part, the similarity between learners and content is found and make the recommendation.

In this section, the various techniques used in this investigation are presented in this section. The Chi-squared, simulated annealing (SA), particle swarm optimization (PSO) and imperialist competitive algorithm (ICA) are used for feature selection. The classifiers used are logistic regression and deep learning network based on MLP feed forward with4-layer for finding recommendations.

3.1 Content matrix

A subset of dataset is used for evaluation. It is preprocessed as follows:

- A table of all words found in the page is prepared which is indexed and a matrix of frequency of words is obtained.
- Stop word list and stemming of words is done.

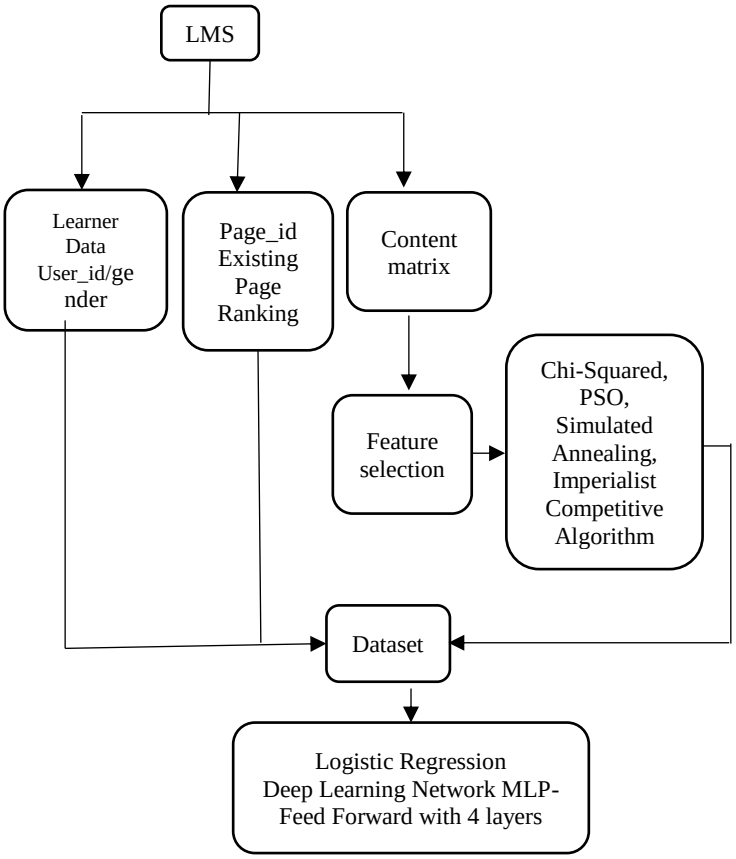

Figure. 1 Overall architecture of recommender system

Based on the above rule the word frequency is created using the exclude list.

The terms document frequency is computed. For a set of given review documents x and a set of terms$a$, a vector v in the a dimensional spaceR^a is modeled for each document can be modeled. The term frequency is denoted by$freq(x,a)$, this represents the number of occurrence of term a in the document$x$. The term-frequency matrix $TF(x,a)$ measures the association of a term $a$with respect to the given document$x$and has a value of zero on non-occurrence of term in the document or a number otherwise. The number could be set as $TF(x,a)=$ 1when term a occurs in the document x or uses the relative term frequency [12]. The **relative term frequency** is the term frequency versus the total number of occurrences of all the terms in the document. The term frequency is generally normalized by Eq. (1):

$$TF(x,a) = \begin{cases} 0 \\ 1 + log\big(1 + log\big(freq(x,a)\big)\big) \end{cases} \begin{matrix} freq(x,a)=0 \\ otherwise \end{matrix} \quad (1)$$

Inverse document frequency (IDF) represents the scaling factor for the terms importance. If term a occurs frequently in many documents, then its importance is scaled down due to its reduced discriminative power [12]. The $IDF(a)$is defined as in Eq. (2):

$$IDF(a) = log\frac{1+x}{x_a} \quad (2)$$

$x_a$is the set of documents containing term$a$.

The numerical data so obtained from the above process for each page is represented as the content matrix and is used to train the classification algorithm. Classification based on logistic regression and deep learning method.

3.2 Chi-squared

Chi-squares is generally used to measure the lack of independence between term (t) and class or category (c) and compared to the X^2 distribution with one degree of freedom [13]. The chi-squared method is for measuring any form of non-availability of independence between a new term and its category. This is a very common statistical test to measure any form of divergence from such distribution in case one assumes such feature occurrence to be independent of a class value. It has an erratic behavior for expected counts and these can be common among text classification as there are some features that occur

Table 2. Calculate chi squared statistics

Pseudocode of Chi squared
Define Hypothesis.
Build a Contingency table (A table showing the distribution of one variable in rows and another in columns. It is used to study the relation between two variables)
Find the expected values.
Calculate the Chi-Square statistic.
Accept or Reject the Null Hypothesis.

rarely and at times owing to a few positive training examples [14].

The chi-squared attribute reduction signifies the relationship amongst the attributes and the comparing categorical output. The disparity from the normal dispersion is estimated through the measurable assessment dependent on presumption that the attribute existence is autonomous of the last categorical output [15]. It is characterized as Eqs. (3) and (4),

$$CHI(t, c_i) = \frac{Nx(AD-BE)^2}{(A+E)x(B+D)x(A+B)x(E+D)} \quad (3)$$

$$CHI\ max_i\ C_{i_{max}} \quad (4)$$

Here A is the rate of recurrence as soon as t&i happens together; B signifies the count of occurrences once t happens in the absence of c_i. E signifies count of occurrences once c_i happens in the absence of t; D is the rate of recurrence once neither c_i nor t happens &N is the entire list of samples in the page collection. The chi-squared measurement will be nil uncertainty t and c_i are free.

3.3 Simulated annealing (SA) based feature selection

The SA heuristic is modeled after the physical annealing process, in which a solid is slowly cooled until its atoms are repositioned to form a crystal with a low-energy state [16]. The algorithm begins with a random solution. It then generates a neighboring solution by slightly modifying the current solution through the use of a perturbation method (which simulates atoms moving randomly within the solid). The algorithm accepts the new solution over the prior if the quality (energy) of the new solution, as determined by the cost function, is lower. Otherwise, the solution is accepted with a probability of $e\ \Delta E/T$, where ΔE is the change in energy between the solutions and T is the solutions temperature. Temperature acts as a control parameter in the algorithm. Once enough perturbations to the solutions are made at a given temperature to allow the energy to settle to a thermal equilibrium, the temperature is decreased until a minimum temperature is reached. Reaching a thermal equilibrium at each temperature allows for a minimum near the global minimum to be found regardless 105 of the initial solution. When implementing the SA algorithm, there are two categories of considerations: problem-specific choices and generic choices. Problem-specific choices involve implementing the various simulated annealing functions, namely the cost function, perturbation method, and the structure of the solution, based upon the problem of interest. Generic choices, such as setting the parameters for cooling, are not directly related to the specific problem a SA algorithm aims to solve.

3.4 Particle swarm optimization (PSO) based feature selection

PSO [17] is a meta-heuristic search technique that simulates the flock of bird's movements to find the food. Each particle in the swarm represents a candidate solution that flies through the multi-dimensional search space. A particle uses the best position explored by itself and its neighbors to move towards an optimum solution. The performance of each particle (i.e. the "closeness" of a particle to the global minimum) is measured according to a predefined fitness function. Suppose that the search space is D-dimensional and there are *m* particles in the swarm. Each particle is located at position $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ with velocity $V_i = [v_{i1}, \mathrm{v}_{i2}, \dots, \mathrm{v}_{iD}]$, where *i=1, 2, …, m*. In the PSO algorithm, each particle moves towards its own best position (p_{best}) denoted as $P_{best_i} = \left[p_{best_{i1}}, p_{best_{i2}}, \dots, p_{best_{iD}}\right]$ and the best position of the whole swarm (*gbest*) denoted as $G_{best} = \left[g_{best_1}, \mathrm{g}_{best_2}, \dots, \mathrm{g}_{best_D}\right]$. And each particle changes its position according to its velocity, which is randomly generated towards the p_{best} and gbest positions. For each particle, *i* and dimension *s*, the new velocity v_{is} and position x_{is} can be calculated as in Eq. (5):

$$v_{is}^t = wv_i^{t-1} + c_1 b_1\left(pbest_{is}^{t-1} - x_{is}^{t-1}\right) + c_1 b_1 \\ (gbest_s^{t-1} - x_{is}^{t-1} x_{is}^t = x_{is}^{t-1} + v_{is}^t \quad (5)$$

Where *t* is the iteration number. The inertial weight *w* is used to control the velocity and balance of the exploration and exploitation abilities of algorithm. A large value of w keeps particles at high velocity and prevents them from becoming trapped in the local optima. A small value of *w* maintains

Table 3. Imperialist competitive algorithm (ICA)

Step 1. First, the population number (Npop), the number of iterations, number of empires, assimilation factor, and the algorithm's revolution rate are selected.
Step 2. Select a few random points on the search space and form the initial empires.
Step 3. The policy of assimilation: move the colonies toward the imperialist countries.
Step 4. Revolution: apply the revolution operator.
Step 5. If a colony in an empire costs less than the imperialist, we will replace the colony and the imperialist.
Step 6. The total cost of an empire is calculated.
Step 7. Select one (or more) colonies from the weakest empire and give it to the empire most likely to conquer.
Step 8. Remove the weak empires.
Step 9. If there is only one empire left, stop; otherwise, return to Step 2.

particles at low velocity and encourages them to exploit the same search area. The constants c_1 and c_2 are called acceleration coefficients that determine whether particles prefer to move closer to the p_{best}or g_{best} positions. The b_1 and b_2 are independent random numbers uniformly distributed between 0 and 1. The termination criterion of the PSO algorithm includes the maximum number of generations, the designated value of p_{best}, or no further improvement in p_{best}. PSO is easily implemented, tuned by few parameters, and escaped from local minima.

3.5 Imperialist competitive algorithm (ICA) based feature selection

Natural and biological phenomena have inspired optimization algorithms, but the imperialist competitive algorithm was created by simulating competition among imperialist countries. Colonization has been an inevitable historical phenomenon, and it is a process by which a central system of power dominates the surrounding land and its components. This algorithm firstly was introduced by the Atashpaz-Gargari and Lucas [18] in 2007, in which they mathematically modeled the social, cultural, and economic evolution of countries. The algorithm initially starts from several countries; in fact, these countries are candidate solutions to the problem. If we compare ICA with previous algorithms, countries are like chromosomes in the genetic algorithm. They are divided into two groups, colonial countries and colonized countries. The algorithm iteratively improves the states by solving specific processes and finally reports the best solution it has been found. The politics of assimilation, colonial competition, and revolution are essential pillars of this algorithm [19]. The main steps of the algorithm are summarized in Table 3:

3.6 Logistic regression

Generally, a logistic regression model calculates theclass membership probability for one of the two categories in the data set as in Eq. (6):

$$P(1|x,a) = \frac{1}{1+e^{-(\alpha-x)}} \quad (6)$$

And $P(0|x,\alpha) = 1\text{-}P(1|x,\alpha)$. Here, $P(1|x,\alpha)$ is written to make the dependence of the posterior distribution on the parameters α explicit. It can be shown that this model is correct when both the class-conditional densities $p(x|1)$ and $p(x|0)$ are multinormal with equal covariance matrices [20].

The hyperplane of all points x satisfying the equation $\alpha.x=0$ forms the decision boundary between the two classes; these are the points for which $P(1|x,\alpha) = P(0|x,\alpha) = 0.5$. A logistic regression model that includes only the original covariates is called a main effects model; including interaction terms such as products makes the model nonlinear in the covariates, and therefore more flexible. Although higher flexibility may be desirable in general, it carries with it a higher risk for model overfitting, which can potentially reduce a model's accuracy on previously unseen cases. In predictive modeling, fitting the training cases is just part of the task: correctly classifying new cases is the most important goal.

3.7 Proposed deep neural network based on multilayer perceptron (MLP)

Deep learning uses artificial neural networks to perform sophisticated computations on large amounts of data. It is a type of machine learning that works based on the structure and function of the human brain. Deep learning algorithms train machines by learning from examples. MLP is a deep learning algorithm. An MLP is a feed-forward artificial neural network (ANN) framework, an adaptive but complex system that is capable of converting its internal structure based on the information that has been passed through. This may also be achieved by adjusting its weight and its connection. The weight refers to the actual number controlling a signal existing between various neurons. Such weights are then adjusted to improve results [21]. MLP is an ideal example of a condition of supervised learning. It is capable of acquiring knowledge in the form of a connected unit of a network. It can be a challenge to extract this kind of knowledge, and such a factor has been motivated for the extraction of data mining rules and their classification. The process further begins using a dataset that may be grouped into two. They

are the training and testing sample. The former is for learning the network, and the latter is for measuring the accuracy of the classifier. The division of the dataset is normally done using various methods like cross-validation, random sampling, and the hold-out method. The steps involved in the neural network are:

- A structure is defined using nodes in hidden, input, and output layers.
- An algorithm is employed to enable the learning process.

The ability of the neural network was to ensure adjustments are made to the structure, and the weight is changed often to make it helpful in artificial intelligence. The MLPs had been trained by using a dataset consisting of different inputs with certain other corresponding results generated based on logical rules. Once this is done, it was applied for the verification of case studies. In the case of the training phase, each record of its training samples needs to be applied to bring about accuracy in the test oracle. In order to increase the quality of the network, the results of the MLP were compared to the accurate results. It was done by computing the distance between the results that were expected and the actual results generated. In order to achieve an adequate error rate, the network parameters (neurons and the biases weights) had been adjusted by using back-propagating error data. This whole process had been repeated and then continued until there was an adequate error rate obtained. The new network adjustment cycles had been completed, and the network is now ready for being employed.

Back propagation (BP) had been observed to be a technique that was extremely popular and used for training the MLPs during the optimization purposes like the gradient descent [22]. The method computed all gradients of loss functions that were connected to the weights identified within the network. Using the right quantity of such input nodes, the hidden layers, including the output layers *n, k,* and *m* were taken as the overall quantity of input instances. These were x_{pi} that implied the instance of P'sith input value by v_{ki} indicating the i-th node for the hidden input layer for the kth node weight. ω_{jk} Was the actual node weight from k to j and also its output layer. To ensure this is easier, the threshold becomes the connection weights along with the hidden layer node k as in Eq. (7):

$$z_{pk} = f(net_{pk}) = f\left(\sum_{i=0}^{n} v_{ki} x_{pi}\right) \quad (7)$$

The output layer nodes for node j as in Eq. (8):

$$y_{pj} = f(net_{pj}) = f\left(\sum_{i=0}^{n} w_{jk} z_{pk}\right) \quad (8)$$

Wherein, the standard sigmoid function is the incentive function as in Eq. (9):

$$f(x) = \frac{1}{1+e^{-x}} \quad (9)$$

A global error function is given as per Eq. (10):

$$E = \sum_{p=1}^{P} E_P = \frac{1}{2}\sum_{p=1}^{p}\sum_{j=1}^{m}(t_{pj} - y_{pj})^2 \quad (10)$$

Wherein E_P is the error of a sample p, and t_{pj} the ideal result. There are adjustment formulae of these weights, which are thus.

The weight adjustment equation for the output layer neurons are in Eq. (11):

$$\Delta\omega_{jk} = \eta \sum_{p=1}^{p}\left(\sum_{j=1}^{m} \delta_{pi}\omega_{jk}\right) z_{pk}(1 - z_{pk}) x_{pi} \quad (11)$$

Wherein, η indicates the rate of learning that has a range between 0.1 and 0.3.

The weight adjustment equation for the hidden layer neurons is as in Eq. (12):

$$\Delta v_{ki} = \eta \sum_{p=1}^{p}\left(\sum_{j=1}^{m} \delta_{pi}\omega_{jk}\right) z_{pk}\left(1 - z_{pk}\right) x_{pi} \quad (12)$$

The notion of the BP had been to make sure the process of learning was split to form two phases; the first would be the forward propagation process in which an input data had been given by means of processing which was done layer by layer for each of the hidden layers along with the actual output value for every unit of y_{pj} that had been computed. The following phase was in a reverse process that had an anticipated output that had not been obtained using an output layer that includes a recursive calculation made layer by layer for error difference between its actual and its anticipated output. This gradient descent technique was employed to alter the weights of the $\Delta v_{ki}, \Delta\omega_{jk}$, to keep the overall error function minimal. In this work, a total of four layers are used; that is, one input layer, two hidden layer and one output layer.

4. Results and discussion

The experiments were conducted on MATLAB platform and WEKA tool. The precision, recall and F-measure for logistic regression and deep neural network are evaluated. The content matrix is created based on the features extracted by TF-IDF. Features are selected using chi squared, SA, PSO and ICA

Table 4. Precision for top 5 &10 recommendations -deep neural network

Precision	Top 5 Recommendations- Logistic Regression	Top 5 Recommendations – Deep Neural Network	Top 10 Recommendations- Logistic Regression	Top 10 Recommendations - Deep Neural Network
Chi Squared	0.61	0.69	0.64	0.71
SA	0.62	0.69	0.64	0.71
PSO	0.63	0.7	0.65	0.72
ICA	0.63	0.7	0.65	0.73

Figure. 2 Precision for top 5 recommendations - deep neural network

algorithms to improve the performance of the classifiers. Based on the features selected the top 5 and 10 recommendations are predicted by the logistic regression and deep neural network. Tables 4 to 6 and Figs. 2 to 7 show the results for Top 5 & 10 recommendations -deep neural network for precision, recall and F-measure respectively.

Table 4 and Fig. 2 shows that the precision for top 5 recommendations using deep neural network performs better than top 5 Recommendations using logistic regression among chi squared, SA, PSO and ICA methods. The precision of ICA performs better by 3.23%, by 1.6% and no change thanChi squared, SA and PSO for top 5 recommendations – logistic Regression. The precision of ICA performs better by 1.439%, by 1.439% and no change thanChi squared,

Figure. 3 Precision for top10 recommendations -deep neural network

SA and PSO for top 5 recommendations – deep neural network.

Learner's always preferred to view the content in fewer recommendations that means learners are interested in smaller recommendations list having the learning content of their interest. This approach generates smaller top 5 recommendations list by placing learner's unseen content in recommendation list and thus attaining high precision value.

Table 4 and Fig. 3 shows that the precision for top 10 recommendations -deep neural network performs better than top 10 recommendations – logistic regression among chi squared, SA, PSO and ICA. The precision of ICA performs better by 1.55%, by 1.55% and no change thanChi squared, SA and PSO for top 10 recommendations – logistic regression. The precision of ICA performs better by 2.78%, by 2.78% and by 1.38% thanChi squared, SA and PSO for top 10 recommendations – deep neural network. It is observed that the precision of the top 10 recommendations is more accurate than the top 5 recommendations. As the number of recommendations increases, the requirement of the learner is better covered.

Table 5 and Fig. 4 shows that the recall for top 5 recommendations -deep neural network performs better than top 5 recommendations – logistic regression among chi squared, SA, PSO and ICA. The recall of ICA performs better by 2.41%, by 2.41% and by 2.41% than chi squared, SA and PSO for top 5 recommendations – logistic regression. The recall of ICA performs better by 4.26%, by 2.11% and by 2.11% than chi squared, SA and PSO for top 5 recommendations – deep neural network.

The feature selection helped to improve the performance of the recommender system. The feature selection refines the search and achieves better performance for recommendations compared to using the full features. As the number of instances increase,

Table 5. Recall for top 5 & 10 recommendations -deep neural network

Recall	Top 5 Recommendations- Logistic Regression	Top 5 Recommendations - Deep Neural Network	Top 10 Recommendations- Logistic Regression	Top 10 Recommendations - Deep Neural Network
Chi Squared	0.41	0.46	0.42	0.48
SA	0.41	0.47	0.43	0.49
PSO	0.41	0.47	0.43	0.49
ICA	0.42	0.48	0.44	0.49

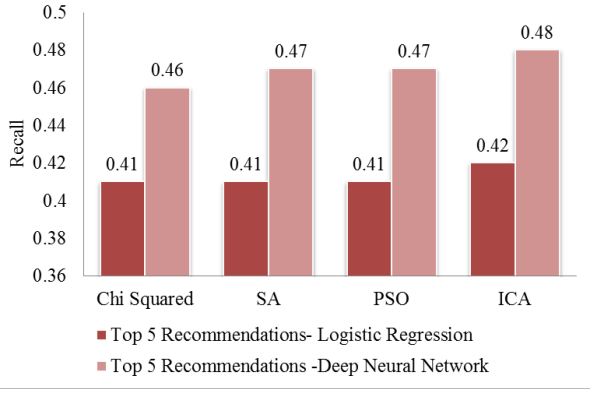

Figure. 4 Recall for top 5 recommendations -deep neural network

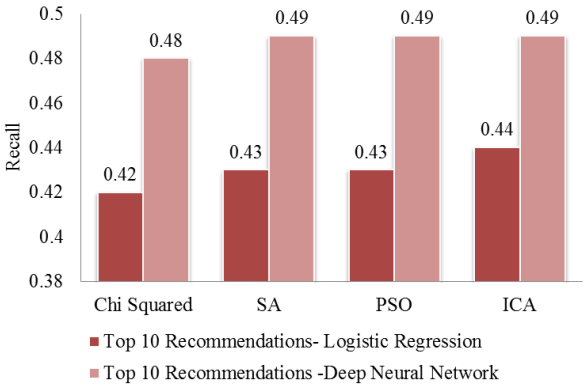

Figure. 5 Recall for top 10 recommendations -deep neural network

Table 6. F-measure for top 5 & 10 recommendations - deep neural network

F-Measure	Top 5 Recommendations- Logistic Regression	Top 5 Recommendations - Deep Neural Network	Top 10 Recommendations- Logistic Regression	Top 10 Recommendations - Deep Neural Network
Chi Squared	0.49	0.55	0.51	0.57
SA	0.49	0.56	0.51	0.58
PSO	0.5	0.56	0.52	0.58
ICA	0.5	0.57	0.52	0.59

the use of metaheuristic methods for feature selection will reduce the execution time.

Table 5 and Fig. 5 shows that the recall for top 10 recommendations -deep neural network performs better than top 10 recommendations – logistic regression among chi squared, SA, PSO and ICA. The recallof ICA performs better by 4.7%, by 2.29% and by 2.29% than chi squared, SA and PSO for top 10 recommendations – logistic regression. The recallof ICA performs better by 2.1%, no change and no change than chi squared, SA and PSO for top 10 recommendations – deep neural network. The feature selection is a NP-hard problem, thus, using metaheuristic methods are feasible to improve the performance of the classifiers.

Table 6 and Fig. 6 shows that the F measurefor top 5 recommendations -deep neural network performs better than top 5 recommendations – logistic regression among chi squared, SA, PSO and ICA. The F-measure of ICA performs better by 2.02%, by 2.02% and no change overChi squared, SA and PSO for top 5 recommendations – logistic regression. The F measure of ICA performs better by 3.57%, by 1.77% and by 1.77% than chi squared, SA and PSO for top 5 recommendations – deep neural network. Deep neural network helps to predict the learner preference and learner type better than the logistic regression.

Table 6 and Fig. 7 show that the F-measure for top 10 recommendations -deep neural network performs better than top 10 recommendations – logistic regression among chi squared, SA, PSO and ICA. The F-measure of ICA performs better by 1.94%, by 1.94% and no change than chi squared, SA and PSO for top 10 recommendations – logistic regression. The F measure of ICA performs better by

Figure. 6 F-measure for top 5 recommendations -deep neural network

Figure. 7 F-measure for top 10 recommendations -deep neural network

3.45%, by 1.71% and by 1.71% than chi squared, SA and PSO for top 10 recommendations – deep neural network. The hypothesis on using feature selection techniques for effective page recommendation is a possible solution for faster training of machine learning algorithms. Deep learning techniques, though requiring more training time, are more efficient machine learning approaches. Though, the deep learning techniques are effective, the selection of hyperparameters of the methods is challenging and significantly impacts the performance. Further work to optimize the selection of hyperparameter needs to be explored.

5. Conclusion

Considering the importance of the e-learning in the current situation with regards to learning among all the world, recommendation systems must be present in the e-Learning platforms to facilitate, propose and recommend articles, books, presentations and courses that will be consulted and followed, as well as valuing the content in the platform. Several works have been made to implement the recommendation systems in eLearning platforms by using the data mining approach. The proposed recommendation system is based on data mining and deep learning techniques. The system is divided into four main parts: The first part for data collection where learners's data is collected and existing page ranking is collected. The second part a content matrix is created by extracting features from the pages. Feature selection is applied to improve the performance and finally the similarity between learners and content is found and the recommendations given using deep neural network.

Experimental results show that that the precision for top 5 recommendations using deep neural network performs better than top 5 recommendations using logistic regression among chi squared, SA, PSO and ICA methods. The precision of ICA performs better by 3.23%, by 1.6% and no change thanChi squared, SA and PSO for top 5 recommendations – logistic regression. The precision of ICA performs better by 1.439%, by 1.439% and no change thanChi squared, SA and PSO for top 5 recommendations – deep neural network. Similarly the results show that the recall, precision and f measure (both top 10 Recommendations using deep neural network).

Future work must concentrate on optimizing the deep neural network.

Conflicts of interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, 1 and 2; methodology, 1; software, 1; validation, 1 and 2; formal analysis, 1; investigation, 1; resources, 1; writing—original draft preparation, 1; writing—review and editing, 1; supervision, 2; project administration, 2;

References

[1] Ó. SanjuánMartínez, G. Pelayo, C. Bustelo, G. R. Crespo, and E. T. Franco, "Using Recommendation System for E-learning Environments at degree level", *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 1, Issue 2, 2009.

[2] M. E. Mabrouk, S. Gaou, and M. K. Rtili, "Towards an Intelligent Hybrid Recommendation System for E-Learning Platforms Using Data Mining. *International Journal of Emerging Technologies in Learning*", Vol. 12, Issue 6, 2017.

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey

and new perspectives", *ACM Computing Surveys (CSUR)*, Vol. 52, Issue 1, pp. 1-38, 2019.

[4] O. Bourkoukou, and E. E. Bachari, "Toward a hybrid recommender system for e-learning personalization based on data mining techniques. JOIV", *International Journal on Informatics Visualization*, Vol. 2, Issue 4, pp. 271-278, 2018.

[5] L. Deng and D. Yu, "Deep learning: methods and applications", *Foundations and Trends in Signal Processing*, Vol. 3, Issue 4, pp. 197–387, 2014.

[6] S. Shishehchi, S. Y. Banihashem, and N. A. M. Zin, "A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system", In: *Proc. of International Symposium on Information Technology, IEEE*, Vol. 1, pp. 1-5, 2010.

[7] H. Tan, J. Guo, and Y. Li, "E-learning Recommendation System", In: *Proc. of International Conference on Computer Science and Software Engineering*, Vol. 5, pp. 430-433, 2008.

[8] S. Bhaskaran, R. Marappan, and B. Santhi, "Design and Analysis of a Cluster-based Intelligent Hybrid Recommendation System for E-learning Applications", Vol. 9, Issue 2, 2021.

[9] P. V. Kulkarni, S. Rai, and R. Kale, "Recommender system in eLearning: a survey", In: *Proc. of International Conference on Computational Science and Applications*, Springer, Singapore, pp. 119-126, 2020.

[10] X. Wang, Y. Zhang, S. Yu, X. Liu, Y. Yuan, and F. Y. Wang, "E-learning recommendation framework based on deep learning", In: *Proc. of IEEE International Conference on Systems, Man, and Cybernetics* (SMC), pp. 455-460, 2017.

[11] H. Zarzour, Z. A. A. Sharif, and Y. Jararweh, "RecDNNing: a recommender system using deep neural network with user and item embeddings", In: *Proc. of International Conference on Information and Communication Systems (ICICS)*, pp. 99-103, 2019.

[12] L. Havrlant, and V. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic", *International Journal of General Systems*, Vol. 46, No. 1, pp. 27-36, 2017.

[13] A. W. Haryanto, and E. K. Mawardi, "Influence of word normalization and chi-squared feature selection on support vector machine (SVM) text classification", In: *Proc. of International Seminar on Application for Technology of Information and Communication*, pp. 229-233, 2018.

[14] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. A. Rahman, and S. Fong, "Feature Selection Methods: Case of Filter and Wrapper Approaches for Maximising Classification Accuracy", *Pertanika Journal of Science & Technology*, Vol. 26, Issue 1, 2018.

[15] H. Saxena, and V. Richariya, "Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain", *International Journal of Computer Applications*, Vol. 98, Issue 6, 2014.

[16] J. Lee, and D. Perkins, "A simulated annealing algorithm with a dual perturbation method for clustering", *Pattern Recognition*, Vol. 112, 2021.

[17] M. Amoozegar, and B. M. Bidgoli, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism", *Expert Systems with Applications*, Issue 113, pp. 499-514, 2018.

[18] E. A. Gargari, and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition", In: *Proc. of IEEE Congress on Evolutionary Computation*, pp. 4661–4667, 2007.

[19] A. Kaveh, P. Rahmani, and A. D. Eslamlou, "An efficient hybrid approach based on Harris Hawks's optimization and imperialist competitive algorithm for structural optimization", *Engineering with Computers*, pp. 1-29, 2021.

[20] S. Dreiseitl, and L. O. Machado, "Logistic regression and artificial neural network classification models: a methodology review", *Journal of Biomedical Informatics*, Vol. 35, Nos. 5-6, pp. 352-359, 2002.

[21] Z. Duan, Y. Wang, and Y. Xing, "Sound Quality Prediction of Vehicle Interior Noise under Multiple Working Conditions Using Back-Propagation Neural Network Model", *Journal of Transportation Technologies*', Vol. 5, No. 02, p. 134, 2015.

[22] A. K. Dwivedi, "Artificial neural network model for effective cancer classification using microarray gene expression data", *Neural Computing and Applications*, Vol. 29, No. 12, pp. 1545-1554, 2018.

[23] B. K. Poornima, D. Deenadayalan, and A. Kangaiammal, "Text preprocessing on extracted text from audio/video using R", *International Journal of Computational Intelligence and Informatics* Vol. 6, No. 4, pp. 267–278, 2017.

[24] D. Deenadayalan, A. Kangaiammal, B. K. Poornima, "Learner Level and Preference Prediction of E-learners for E-learning

Recommender Systems", *Studies in Computational Intelligence*, Vol 771, Springer-2019,

Appendix 1

Table. Notations

Notation	Description
$TF(x,a)$	Term Frequency
x	document
a	Term
$IDF(a)$	Inverse document frequency
$CHI(t,c_i)$	Chi-Squared
t	Term
c_i	Class
A	Rate of recurrence
B	Count of occurrences
D	Rate of recurrence
N	Entire list of samples
ΔE	Change in energy between the solutions
T	Temperature
x_{i1}	Position of particle
v_{i1}	Velocity of particle
$gbest$	Best position of particle in whole swarm
p_{best}	Best position of particle
w	Inertial weight
c_1 and c_2	Acceleration coefficients
b_1 and b_2	Independent random numbers uniformly distributed between 0 and 1
$P(1\|x, \alpha)$	Dependence of the posterior distribution on the parameters α
x_{pi}	Input Value
v_{ki}	i-th node for the hidden input layer
ω_{jk}	Node weight
E_P	Error of a sample p
t_{pj}	Ideal result.
η	Rate of learning
y_{pj}	Output