Enhancing E-Learning with Deep Learning:

A Review of Advanced Course

Recommendation Systems

#### Prof. Roshan S. Bhanuse1, Dr. Supriya Patil2 , Sandip Mal3, Harshsingh Bais 4, Ganesh Kalyankar5, Dipanshu Balki6, Prathmesh Bondre7, Harsh Ainchwar8

1,4,5,6,7,8Department of Computer Technology, Yeshwantrao Chavan College of Engineering, Wanadongari, Hingna Road, Nagpur, Maharashtra, India -441110

3School of Computing Science and Engineering, VIT Bhopal University, Bhopal, India-466114.

²Department of Preventive & Social Medicine, Krishna Institute of Medical Sciences, Satara

Correspondence: bondreprathmesh03@gmail.com(P.B) and [hsb03062003@gmail.com(H.B)](mailto:hsb03062003@gmail.com(H.B))

**Abstract:** In the rapidly expanding field of online education, it is a complete abundance of available courses for students, making it difficult to identify courses that meet their specific needs, interests and skills levels. To alleviate this problem, we suggest "the system of e-learning recommendations using deep learning", which uses advanced machine learning techniques and provides personal design proposals based on user inputs and behavioral patterns.

It uses the XGBOOST algorithm, a powerful file learning algorithm, to process user data and generate accurate course recommendations for users. XGBOOST is selected for its effectiveness in handling structured data, robustness against excessive and high predictive performance. In addition to numerical and categorical data processing, the system integrates the analysis of the Vader sentiment to interpret feedback and reviews of text users and gains a deeper insight into users' preferences and sentiments for specific courses.

The proposed system is implemented using a flask, a lightweight and scalable web frame to make it easier to interact with users and recommendations. SQLite serves as a database management system and ensures efficient storage and search of user data and course information. Pandas, a widely used data manipulation library, is used for efficient pre -work and analysis of large data sets. To increase the flexibility of deployment and scalability, trained machine learning models are seriousized using Joblib, allowing them to easily load and integrate into the real world applications. The experimental results of the developed approach outperformed the compared existing schemes in terms of accuracy (99.79%), F1-score (99.648%), Kappa (99.061%), RMSE (0.2092), precision (99.69%), recall (99.698%), MAE (0.012), HR (0.9321) and ARHR (0.1621) and AUC (99.79%).

**Keywords**: E-learning, Course Recommendation System Deep Learning, XGBoost Algorithm, VADER Sentiment Analysis

**1. INTRODUCTION**

Currently, everyone is trying to learn different courses to improve their knowledge. However, choosing the right course is a challenging task for new students because of the huge availability of online sources [9]. Effective framework of recommendations is required to propose new students on the basis of their interests. The absence of the correct system of recommendations increases the likelihood that the termination is of course to be terminated [7, 11]. This occurs when pupils are enrolled in courses without sufficient knowledge about them, leading to a loss of interest and possible release. E-learning recommendations play a key role in precise design of students based courses and help informed decisions [23, 24].

A number of online learning courses are currently available, making students identifying the best for their needs [1]. In addition, for optimizing the recommendation of the online learning course [20], an advanced approach based on deep learning is a necessary approach [20]. With the growing online learning, there is a growing need for accurate and personalized recommendations systems. These systems must be innovative to improve course design and improve user involvement [8].

Online learning serves as a strong alternative to the traditional educational environment and offers students to study flexibility at their own space based on their interests [2].

In the learning online course, the recommendations help students in effective adaptation with program training [31]. However, a large volume of information and the constant growth of online courses complicates the selection process for students. In previous works, various schemes were designed to improve the frames of recommendations [5]. However, the improvement of the quality of recommendations remains ongoing research challenge, which requires the development of more efficient methodologies to improve the recommendation process.

With increasing demand for lifelong learning, e-learning has become a vital tool for students of the 21st century [6]. It provides access to online course materials, making learning more interactive and accessible [12, 21]. E-learning platforms seize students by providing them on their experience in learning and ensuring greater involvement compared to traditional education methods [4]. Previous research has explored various machine learning-based methodologies [25, 27] and optimization techniques such as genetic algorithms, deep learning approaches [28] and the most closely neighbors (KNN) to predict the best online courses. Many scientists and institutions now integrate the framework of recommendations into e-learning platforms to increase the choice of the course [14, 15]. These frameworks help students take accurate decisions that corresponds to their learning objectives [16].

Despite their importance, current framework for online learning recommends several challenges, including Sparsity, generalization, syllabus updates and multidimensional user’s preferences. The aim of the proposed system is to solve these challenges by providing a reliable mechanism of the recommendation of the course. With many courses available on the Internet, students often try to choose the most suitable. Therefore, an effective framework of recommendations is required to keep students on the basis of their interests.

The main contributions of the proposed work are as follows:

1. Pre-processing: Applies tokenization, stemming, stop-word removal, noise filtering, and lemmatization for data cleaning.
2. Feature Extraction: Uses TF-IDF, Word2Vec, GloVe, and N-gram models to transform text into numerical features.
3. Optimization Algorithm: Introduces Improved Horse Herd Optimization (IHHO) for efficient feature selection and reduced data size.
4. Sentiment Classification: Combines VADER and lexicon-based methods to classify course sentiment as positive, negative, or neutral.
5. Course Recommendation: Implements a hybrid content-based and collaborative filtering model for personalized course recommendations.
6. Performance Evaluation: Achieves superior accuracy, precision, recall, F1-score, RMSE, and AUC compared to existing methods.

The rest of this paper is described as follows: Section 2 provides a review of current related works, Section 3 provides a detailed explanation of the presented methodology, Section 4 describes the results and the corresponding discussions and the paper is concluded in Section 5.

**2.RELATED WORK**

1. learning recommendation systems have received a lot of attention as a result of the rapid growth of online learning platforms. Traditional methods such as collaborative filtering and content-based filtering sometimes encounter challenges such as scant data, cold-start problems, and lack of customisation. To solve these issues, deep learning-based approaches are increasingly being researched. Liu et al. [1] provided a comprehensive examination of deep learning techniques in e-learning recommender systems, stressing the importance of multilayer perceptrons (MLP) and recurrent neural networks (RNNs) in improving personalization.

Bhanuse and Mal [2] shown how deep neural networks (DNNs) enhance the quality of suggestions by categorizing the components of recommendation systems into deep learning models, collaborative filtering, and user profiles.A customized e-learning system that blends process mining and deep learning was introduced by Chanaa and El Faddouli [3]. Similarly, Li and Kim [4] developed DECOR, a deep learning-based course recommender system that enhances recommendation accuracy through high-level feature interactions.

Many academics have investigated the use of convolutional neural networks (CNNs) in e-learning recommendation systems. Alatrash et al. [5] presented a CNN-based sentiment analysis model that enhances course suggestions by examining student reviews and feedback, whereas Xie et al. [6] employed CNNs to scan instructional content and suggest learning resources based on textual and visual patterns. Moreover, sequential learning behavior has been represented by long short-term memory (LSTM) networks and recurrent neural networks (RNNs). An LSTM-based course recommendation model was created by Wang et al. [7] to better identify the next course a student would require by capturing sequential learning habits. The application of gated recurrent units (GRUs) in e-learning recommendation systems was further investigated by Jiang et al. [8], who showed how temporal learning behaviors can result in more precise course recommendations.

A lot of research has also been done on hybrid strategies that combine collaborative filtering and deep learning. To improve course recommendations, Jena et al. [9] used a collaborative filtering-based model that included singular value decomposition (SVD) and k-nearest neighbors (KNN). While De Medio et al. [11] presented MoodleRec, a hybrid recommender system that combines content-based filtering and collaborative learning, Zhang et al. [10] integrated collaborative filtering with CNNs to achieve improved accuracy in recommending online courses,filtering in learning platforms like Moodle. Deep reinforcement learning (DRL) in e-learning recommendation systems has also been the subject of some research. Yin et al.

[13] used DRL to create a personalized tutoring system that uses student learning behaviors to modify suggestions over time, while Sun et al. [12] used DRL to dynamically recommend instructional content based on student participation.  
The use of natural language processing (NLP) methods in course suggestions has grown. While Salau et al. [15] examined NLP-based recommendation systems, focusing on topic modeling and semantic analysis for content-based suggestions, Aher and Lobo [14] used text mining and NLP to examine Moodle data and enhance recommendations. In order to increase the precision of semantic-based suggestions, Kumar et al. [16] used BERT and Transformer models to examine student feedback and course descriptions. Context-sensitive suggestion With Souabi et al. [17] dividing recommendation systems into content-based, collaborative, and hybrid approaches, and Patil et al. [18] presenting a context-aware deep learning model that modifies recommendations according to student engagement and performance metrics, context-aware recommendation systems have also become a successful strategy.

Deep learning has also helped recommendation systems that rely on personalized learning. While Sharma et al. [20] presented deep clustering techniques to classify students and provide individualized learning materials, Huang et al. [19] created an adaptive learning system based on graph neural networks that analyzes student learning routes to enhance suggestions. A student-centered recommendation system that adjusts course recommendations according to learning preferences and speed was created by Liang et al. [21]. Knowledge graphs have also been investigated as a way to improve recommendation systems.

In order to improve course suggestions, Zhou et al. [22] created a knowledge graph-enhanced recommendation system that incorporates semantic linkages between learning materials. Graph neural networks (GNNs) were used by Xiong et al. [23] to enhance learning path suggestions by guaranteeing that recommended courses correspond with personal learning objectives.

Wu et al. [24] introduced an emotion-aware e-learning recommendation system that tailors suggestions based on sentiment analysis of student input, demonstrating the growing popularity of emotion-aware recommendation systems. In order to improve the selection of motivational learning resources and raise student engagement, Chen et al. [25] used emotion detection algorithms. To increase openness and trust, explainable artificial intelligence (XAI) has also been included into e-learning recommendation systems. Explainable deep learning in recommendation systems was investigated by Ren et al. [26], enabling students to comprehend the rationale behind course recommendations, whereas Gao et al. [27] enhanced user satisfaction by creating a transparent AI model for interpretable course recommendations.

Numerous real-world implementations of recommendation systems based on deep learning in MOOCs and learning management systems (LMS) have been examined. In their analysis of the effects of deep learning-based recommendation systems in MOOCs, Kumar and Jain [28] noted increases in dropout prediction and student engagement. Santos et al. [29] demonstrated a notable increase in course completion rates by integrating adaptive learning recommendation algorithms into platforms such as Coursera and Udemy. As mentioned by Dai et al. [30], who investigated its potential for protecting privacy while enhancing suggestions, federated learning is one of the future research paths in deep learning-based e-learning recommendation systems.

In order to offer a comprehensive and engaging learning experience, Tian et al. [31] created multi-modal recommendation models that incorporate text, graphics, and video. Future e-learning environments will be more complex and intelligent thanks to the substantial improvements in personalization, student engagement, and course adaptation brought about by the development of deep learning techniques in recommendation systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Author(s) (Year) | Method | Merits | Demerits |
| 1 | Liu et al. (2022) | Review of Content-Based, Collaborative Filtering, Knowledge-Based Recommendations | Provides a systematic discussion of main recommendation techniques in e-learning.Helps understand current development and future directions. | Each technique has its limitations; requires considering hybrid systems. |
| 2 | Bhanuse & Mal (2021) | Hybrid recommender systems, Content-based, Collaborative-filtering | Synthesis of results presents recommender systems types in e-learning.Addresses explicit feedback and rating. | Lacks critical reflection on social learning and social educational networks.Weak database size used in some research work. |
| 3 | Aher & Lobo (2023) | Combination of Content-based and Collaborative Filters | Improves learning experiences by helping students locate courses.Aids in defining future advances for researchers, instructors, and practitioners working in the field of online education. | Requires a comprehensive overview of recommender systems, covering challenges and strategies used. |
| 4 | Jena et al. (2022) | Collaborative Filtering | Suggests courses based on the collaborative filtering mechanism.Helps learners make choices without sufficient personal experience of the alternatives. | May suffer from cold start problem for new users; sensitive to data sparsity. |
| 5 | Salau et al. (2022) | Deep Learning Survey | Offers insights into current trends in deep learning for e-learning recommender systems. | Limited by the scope of included studies. |
| 6 | Chanaa & El Faddouli (2022) | Deep Learning | Applies advanced deep learning techniques for intelligent systems. | Requires substantial training data and computational power; prone to overfitting. |
| 7 | De Medio et al. (2020) | Recommender System | Develops a system for course creation, providing integration with a platform. | System performance may depend on server capabilities. |
| 8 | Souabi et al. (2021) | Recommendation Systems Review | Reviews various recommendation systems in e-learning contexts, identifying strengths and weaknesses. | Limited by the scope of included studies. |
| 9 | Li & Kim (2021) | Deep Learning | Focuses on sustainable development education through recommendations. | May require specialized datasets related to sustainability. |
| 10 | Alatrash et al. (2021) | Sentiment Analysis using Deep Learning | Incorporates sentiment analysis to enhance recommendations, capturing subjective user preferences. | Performance depends on the accuracy of sentiment analysis models; potential for bias. |

Table 1 Comparison analysis of merits and demerits of existing approaches

**3.METHODOLOGY**

# Methodology for E-Learning Course Recommendation System

By combining user input with cutting-edge machine learning techniques, the proposed "E-LEARNING COURSE RECOMMENDATION SYSTEM USING DEEP LEARNING" is intended to offer customized course recommendations. Through a dashboard that divides courses into technical and non-technical domains, the system gathers user preferences using Flask for the web interface and SQLite for database administration. When users respond to particular questions, Pandas is used to process and manipulate the data. A potent machine learning algorithm called XGBoost is used by the system to evaluate the input data and forecast appropriate courses. Furthermore, textual data is interpreted using VADER sentiment analysis, which improves comprehension of user preferences. Joblib is used to serialize the trained models, guaranteeing effective storage and retrieval throughout user interactions.  
The trained models are serialized with Joblib, ensuring efficient storage and retrieval during user interactions. This comprehensive approach ensures that users receive tailored course recommendations, enhancing their e-learning experience by aligning course offerings with individual needs and sentiments.

## System Overview

The proposed E-Learning Course Recommendation System utilizes deep learning techniques, sentiment analysis, and hybrid similarity frameworks to provide personalized course recommendations. The system processes user preferences through:

• Machine Learning (ML) Models: XGBoost and Hybrid BLSTM-CNN for course classification.  
• Sentiment Analysis: VADER & Hybrid Lexicon-based Approaches.  
• Feature Selection & Optimization: Improved Horse Herd Optimization (IHHO).  
• Hybrid Recommendation Strategy: Combining content-based and collaborative filtering.

## Data Collection and Preprocessing

### 2.1 Data Collection

User data is collected through a dashboard where learners provide educational background, interests, skill level, gender, state, and hobbies. The dataset consists of various e-learning courses, reviews, and user interactions.

### 2.2 Data Preprocessing

To improve data quality, the following techniques are applied:

• Tokenization: Splits text into individual words.  
• Stopword Removal: Eliminates words like 'the,' 'is,' 'and,' etc.  
• Noise Filtering: Removes irrelevant symbols, numbers, and whitespace.  
• Feature Extraction: Uses TF-IDF, Word2Vec, GloVe, and N-gram models to transform text into numerical representations.

## Feature Engineering

Feature extraction techniques used:

• TF-IDF (Term Frequency-Inverse Document Frequency):  
 TF = Number of times term appears in a document / Total number of terms in the document  
 IDF = log(Total number of documents / Number of documents containing the term)  
 TF-IDF = TF × IDF

The choice of **XGBoost, Hybrid BLSTM-CNN, and IHHO** is justified as follows:

* **XGBoost (Extreme Gradient Boosting)**: Selected for its high efficiency, scalability, and superior predictive performance in handling structured data.
* **Hybrid BLSTM-CNN**: Used for learning sequential dependencies and extracting important textual features from user reviews.
* **IHHO (Improved Horse Herd Optimization)**: Applied for feature selection to reduce dimensionality and improve classification accuracy.

• Word2Vec: Converts words into vector representations using Continuous Bag of Words (CBOW) and Skip-gram models:  
 P(w\_k) = exp(I\_O^T \* I\_I) / Σ exp(I\_O \* I\_I) (1)

• GloVe (Global Vectors for Word Representation):

F\_{glove} = \log\left(\frac{P\_k - P\_l}{m P\_m}\right) (2)

• N-gram Feature Extraction:  
 Example: 'Python is great' → Unigrams: [Python, is, great], Bigrams: [Python is, is great]

## Model Training and Deployment :

Machine learning models used:

• XGBoost (Extreme Gradient Boosting): Used for personalized course prediction:

y\_m^l = f(Σ y\_{m-1}^l \* w\_{lm} + b\_m^l) (3)

XGBoost (**Extreme Gradient Boosting**) is an optimized gradient boosting algorithm that provides fast and accurate predictions. It is widely used for structured data problems because of its efficiency, scalability, and ability to handle missing data.

**XGBoost :**

1. **Boosting Framework**:
   * XGBoost follows the boosting approach, where multiple weak learners (decision trees) are trained sequentially.
   * Each tree learns from the errors of the previous tree to improve prediction accuracy.
2. **Gradient Boosting Mechanism**:
   * A loss function (e.g., Mean Squared Error for regression, Log Loss for classification) is minimized.
   * The model calculates gradients (direction of improvement) and updates the trees iteratively.
3. **Tree Pruning for Efficiency**:
   * XGBoost uses a **depth-wise pruning strategy** to reduce overfitting.
   * It avoids unnecessary splits and optimizes tree depth using the **max\_depth** parameter.
4. **Regularization**:
   * **L1 (Lasso) and L2 (Ridge) regularization** techniques prevent overfitting.
   * Shrinks less useful features while keeping important ones.
5. **Handling Missing Values**:
   * XGBoost can automatically learn the best direction to handle missing values instead of manually imputing them.
6. **Parallel Processing & Hardware Optimization**:
   * Unlike traditional gradient boosting, XGBoost performs computations in parallel using a histogram-based approach.
   * This makes it significantly **faster** than other boosting algorithms.

**PSEUDOCODE OF A potent machine learning algorithm called XGBoost :**

import xgboost as xgb

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

data = load\_dataset("course\_recommendation.csv")

X, y = data.drop("course\_label", axis=1), data["course\_label"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

params = {

"objective": "multi:softmax",

"num\_class": 10,

"max\_depth": 6,

"learning\_rate": 0.1,

"n\_estimators": 100,

"subsample": 0.8,

"colsample\_bytree": 0.8

}

model = xgb.XGBClassifier(\*\*params)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

## Sentiment Analysis :

To enhance course recommendations, sentiment analysis is applied using VADER and Lexicon-based Hybrid Models:

• VADER (Valence Aware Dictionary for Sentiment Reasoning) calculates sentiment polarity.

**Refining Course Recommendations**:

* Courses with consistently **positive reviews** are ranked higher in recommendations.
* Courses with mixed or **negative reviews** might be reconsidered based on additional feedback.

**Understanding User Preferences**:

* Helps **identify strengths and weaknesses** of a course beyond numerical ratings.
* Ensures users receive courses aligned with their **interests and satisfaction levels**.

**Handling Subjectivity in Ratings**:

* A 5-star rating does not always explains **why** the user liked a course.
* Sentiment analysis extracts **hidden patterns** from textual feedback.

1. **VADER (Valence Aware Dictionary and Sentiment Reasoner)**

VADER is a **rule-based sentiment analysis model** specifically designed for short text, making it highly suitable for user reviews.

**Working Principle**:

* Uses a **predefined lexicon** of words associated with positive, negative, and neutral sentiments.
* Assigns **polarity scores** to words:
  + Positive words → **+ve score**
  + Negative words → **-ve score**
  + Neutral words → **0 score**
* Considers **intensifiers** like *very*, *extremely*, and *not* to adjust sentiment weights.

**VADER Sentiment Score Calculation**:

SentimentScore=(Positive words−Negative words)Total wordsSentiment Score = \frac{(Positive\ words - Negative\ words)}

{Total\ words}SentimentScore=Total words(Positive words−Negative words)​

If **Sentiment Score > 0** → **Positive review**  
If **Sentiment Score < 0** → **Negative review**  
If **Sentiment Score ≈ 0** → **Neutral review**

**Example Sentiment Calculation Using VADER:**

**Review**: *"This course is absolutely amazing! The instructor explains concepts very clearly."*  
VADER Output → **Positive Score: 0.92**

## Course Recommendation Strategy :

• Hybrid Content-Collaborative Similarity:

S(V₁, V₂) = (|l₁ ∩ l₂| + |p₁ ∩ p₂| - |l₁ ∩ p₂| - |l₂ ∩ p₁|) / |l₁ ∪ l₂ ∪ p₁ ∪ p₂|) (4)

## Performance Evaluation :

* **Accuracy**: Measures correct predictions over total predictions.
* **Precision**: Fraction of relevant recommendations among retrieved ones.
* **Recall**: Ability to identify relevant recommendations.
* **F1-score**: Harmonic mean of precision and recall.
* **RMSE (Root Mean Squared Error)**: Measures deviation between predicted and actual ratings.
* **Hit Rate (HR)**: Proportion of correct recommendations within the top-N suggestions.
* **Average Reciprocal Hit Ranking (ARHR)**: Evaluates ranking effectiveness of recommendations.

This methodology integrates deep learning, sentiment analysis, and optimization to deliver personalized and highly accurate course recommendations. By incorporating Hybrid BLSTM-CNN, XGBoost, and IHHO, the system ensures higher accuracy and reduced error rates. Future work may focus on real-time updates, reinforcement learning, and improved privacy measures.

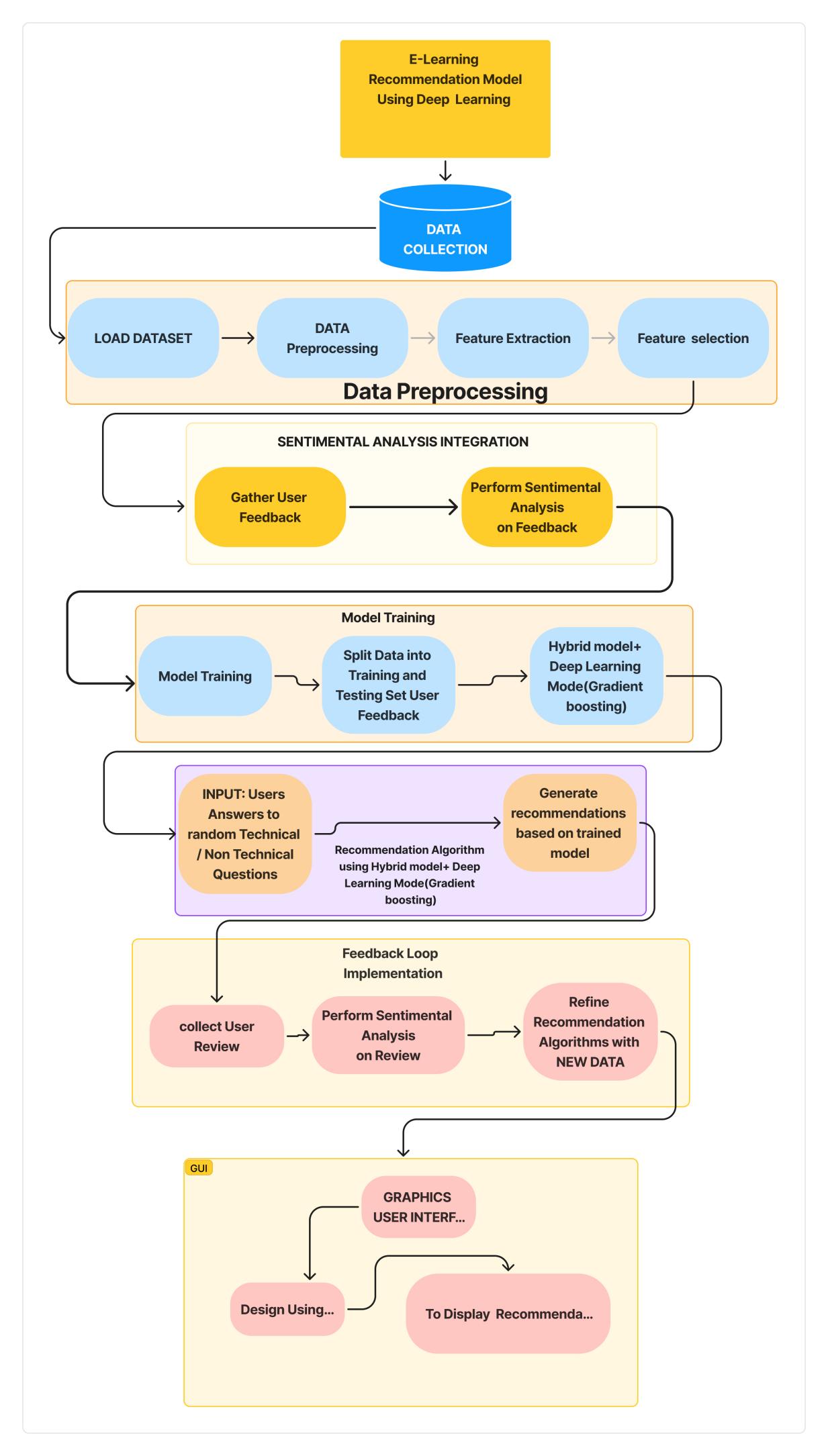


Figure 1 . Flow Chart for Course recommendation system

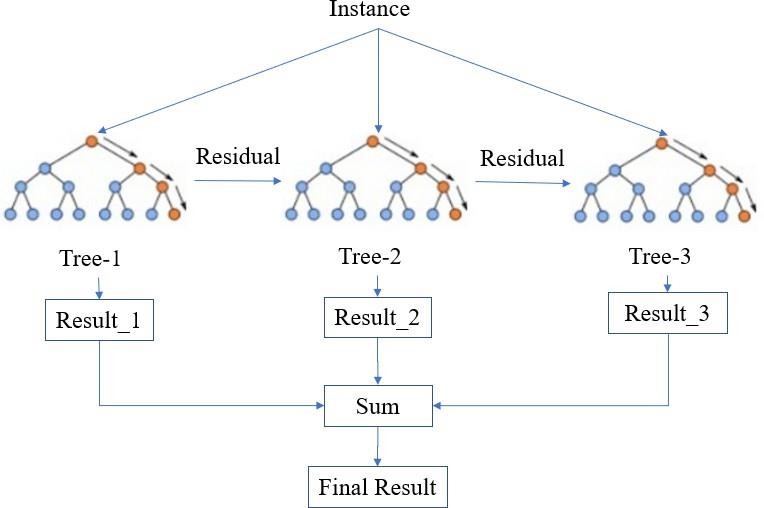


Figure 2 . Shows the XGBoost architecture

# MODULES REQUIREMENT

* Flask==0.12.3
* Sqlite3
* Pandas
* Joblib
* Random

**SOFTWARE REQUIREMENT**

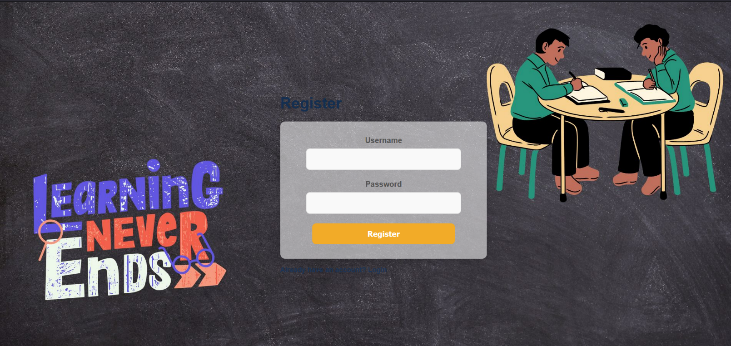
* Python Software IDE
* Anaconda

**4. Results and Discussion**

To provide individualized recommendations for the course, the recommended system of e-learning courses is used by the XGBOOST and Vader sentiment analysis. Python is used in its implementation, with a Jupyter notebook acting as a training environment, Flask handling the graphical user interface (GUI), and SQLite manipulation with the database. To ensure smooth operation and efficient performance, the system was tested on a 64-bit computer with Intel i5 and 8 GB RAM.

And the number of measures, including accuracy, precision, recall, RMSE, F1-score, AUC, hit rate (HR), average reciprocal hit ranking (ARHR), mean absolute error (MAE), and the Kappa metric, assess its efficacy. Users can easily navigate the system because of its interactive graphical user interface. Figure 3 . displays the user's registration and the input screen that shows a user-friendly design. In addition Figure 4 . illustrates how to send a new query, and Figure 5 . shows the recommendation of the course.

The Table 1. provides specifics for setting the hyperparameter used in the model. Optimization of learning efficiency requires fine-tuning these factors. Controlling the volume of data processed by each iteration, the dose size of 32 guarantees the stability of the training. Layer-wise learning and classification are optimized using the softmax activation function. In addition, the model passes through 100 training epochs, allowing it to start the data several times to increase the prediction of reliable and accurate courses.

 A screenshot of a computer

AI-generated content may be incorrect.

Figure 3 . User login page Figure 4 . User enters his/her skill level

A screenshot of a computer

AI-generated content may be incorrect. A computer screen shot of a computer

AI-generated content may be incorrect.

1. (b)

Figure 5 . (a) Non-Technical Recommended courses and (b) Technical Recommended courses

4.1 Dataset Description

The data file used to evaluate the proposed e-learning includes a wide range of courses in the course recommendation system. In addition to technological courses such as Python, Java, data science, and machine learning, it includes non-technical disciplines such as art, literature, history, and music. A thorough description and reviews generated by users who provide light on the productivity of the course are included in each submission of the course. The data file is divided into 80:20 between sets of training and testing to assess the performance of the proposed model. Personalized proposals are improved on the basis of students' interactions and the specifics of the course in the data file.

|  |  |
| --- | --- |
| Parameter | Value |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Activation Function | Softmax |
| Beta Values (β1 , β2 ) | (β1 = 0.9 , β2 = 0.999) |
| Epochs | 100 |
| Regularization | Dropout(0.5), Batch Normalization |
| Hardware | Intel i5 (8 GB RAM) |

Table 2 . Hyperparameter settings used in the model

4.2 Performance metrics

Several performance measures are used to guarantee its success in evaluating the proposed system of the system recommendation. The accuracy of decision-making support and statistical accuracy are two categories into which these measures are divided. Accuracy, appeal, F1-Score, and the Receiver Operating Characteristic (ROC) are metrics used to evaluate the accuracy of decision-making assistance. These metrics evaluate the ability of the system to create excellent quality proposals. On the contrary, statistical accuracy uses correlation, average absolute error (MAE), and the root mean square error (RMSE) to compare the expected and actual user evaluation. Metrics including accuracy, memories, F1-score, RMSE, Kappa, hit rate (HR), average reciprocal hit ranking (ARHR), MAE, and the area below the curve (AUC) are used to compare the current methods to verify their effectiveness. These tests help to evaluate the reliability of the model in issuing the recommendations of wise routes. The following subchapters offer a thorough examination of different metrics.

4.2.1 Accuracy

This performance is considered to be an evaluation of the ratio of accurate classification. It is a share of precisely identified classes of total classes. It is evaluated in the following Eq. (6)

(6)

In this case, TN stands for the true negative, FP for the false positive, TP for the true positive, FN for the false negative.

4.2.2 Kappa Metric

The degree of agreement between the expected and actual classifications is evaluated using the Kappa metric, which takes into account a random chance. It evaluates the accuracy of the model in the classification of courses beyond what would only be predicted by chance. Eq. (7) is used to calculate Kappa statistics, where stands for the calculated accuracy level and for fluctuations in probability in accuracy. Better consistency and reliability of the model classification is marked with a higher Kappa score.

(7)

4.2.3 Precision

More accurately quantifies the share of correctly predicted positive courses from all expected positive courses. They will assess the share of the proposed courses that are truly relevant to users. The system is more reliable when the accurate score is higher because it suggests fewer false positive recommendations. Using Eq. (8), a metric of accuracy is calculated.

(8)

4.2.4 Recall

The evaluation evaluates the ability of the model to recognize real positive courses and at the same time take into account false negatives and real positives. Assesses the system's ability to extract the relevant courses from the data file. The algorithm successfully captures the appropriate recommendations when the evocation value is greater. Eq. (9) is used to determine the invocation metric.

(9)

4.2.5 F1-score

A more thorough assessment of the power of the model is provided by F1-Score, which is a harmonious average of accuracy and download. This is particularly useful in situations where the distribution of the course categories is not uniform. The recommendation system, which is well balanced, has a higher score F1. The F1-score formula is provided by Eq. (10).

(10)

4.2.6 AUC (area under curve)

AUC evaluates the ability of the model to distinguish between the relevant and irrelevant courses by calculating the correlation between real positive and false positive rates. Since AUC measures the likelihood that a randomly selected positive instance will be evaluated higher than negative, a higher value indicates better classification performance. Eq. (11) is used to calculate AUC.

(11)

4.2.7 RMSE (root mean squared error)

The RMSE system recommends the difference between the expected and the actual values. Because it shows fewer prediction errors, the lower RMSE value indicates a more accurate model. Eq. (12) is used to calculate RMSE metrics, where is predicted for the observation in the data file, is an observed value for observation in the data file, and N is the size of the sample.

(12)

4.2.8 MAE (mean absolute error)

The difference between the expected levels () and the actual outputs () is measured by MAE while considering the total number of test cases (). It measures exactly how the model can estimate user preferences. More accurate recommendations of the course are marked with a lower MAE number. Eq. (13) represents a MAE calculation.

(13)

4.2.9 HR (hit rate)

The ratio of successfully recommended courses (courses hit the actual positives, ) due to the overall recommendations correct and incorrect () is known as the rate of intervention. It illustrates how well the model can recommend relevant courses. Eq. (14) formulates the degree of intervention.

(14)

4.2.10 ARHR (Average Reciprocal Hit Ranking)

By calculating the sum of the mutual evaluation of each evaluation () for total interventions (), they will assess the evaluation of precisely proposed courses. The increased ARHR number suggests that the system gives greater weight to relevant courses, which generally increases the user experience. Eq. (15) defines the calculation of ARHR.

(15)

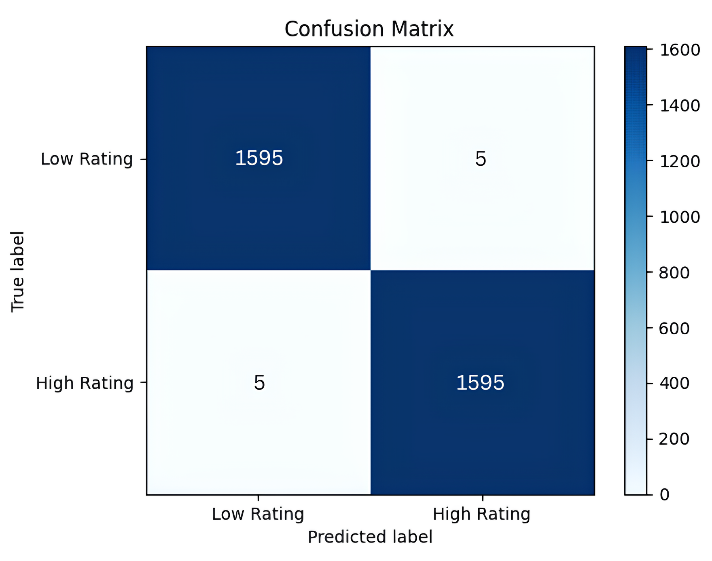
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Technique* | *Accuracy(%)* | *F1-score(%)* | *Precision(%)* | *Recall(%)* |
| SS+ Opinion documents | 91 | 82 | 80 | 85 |
| Fuzzy logic+SS | 97 | 87 | 89 | 86 |
| HIRS | 89.09 | 88.98 | 89.2 | 86.94 |
| AFINN | 74 | 67 | 70 | 65 |
| Dictionary-based approach | 82 | 69 | 78 | 62 |
| Deep learning | 99.01 | 98.63 | 98.7 | 98.57 |
| KNN | 98.59 | 97.05 | 96.23 | 97.89 |
| Hybrid BLSTM CNN | 99.79 | 99.64 | 99.69 | 99.698 |
| Proposed (Technical) | 99.85 | 99.77 | 99.75 | 99.80 |
| Proposed (Non Technical) | 99.83 | 99.75 | 99.72 | 99.78 |

Table 3 . Comparison of different model performance measurement

4.3 Performance examination

Using different power metrics, including F1-Score, Kappa, RMSE, accuracy, memories, intervention levels (HR), average hit (ARHR), average absolute errors (MAE), and AUC, the proposed performance of e-learning recommendations is evaluated using current methods. The results of the classification in terms of true positives, true negatives, false positives, and false negatives are highlighted in Figure 6 . ,which shows the confusion matrix . These numbers are necessary to assess how well the model predicts the recommendations of the course.

Table 2. offers a thorough comparison of the model performance measurement with current approaches. Compared to methods such as fuzzy logic with semantic similarity (SS), hybrid intelligent recommendations (HIRS), dictionaries based, SS combined with opinion documents, AFINN, deep learning models, K-Nearest Neighbors (KNN), and hybrid BLSTM-CNN [23].

A blue squares with white text

AI-generated content may be incorrect.

(a) (b)

Figure 6 . (a) Confusion matrix of Technical Course and (b) Confusion matrix of Non-Technical Course

Figure 7 . compares the accuracy of the proposed e-learning recommendation system with existing methodologies, including fuzzy logic with semantic similarity (SS), dictionary-based approaches, SS combined with documents, AFINN, hybrid intelligent recommendation system (HIRS), K-Nearest Neighbors (KNN) and Hybrid BLSTM-CNN [23]. The analysis emphasizes that the proposed hybrid approach achieves higher accuracy than the existing method.

In Figure 8 . The precision performance of the proposed system of e-learning course is compared with existing methodologies, including the dictionary-based approach, fuzzy logic with semantic similarity (SS), SS combined with documents, AFINN, hybrid intelligent recommendation System (HIRS), (KNN), and the Hybrid BLSTM-CNN [23]. The analysis shows that the proposed approach has excellent precision and ensure that the user has a higher proportion of correct recommended courses. In addition, the precision of the system is 99.85% and 99.83% for technical courses and non technical courses respectively, which further verifies its efficiency in obtaining relevant courses. In addition, Figure 9 . compares Recall across methodology.

Figure 7 . Accuracy performance comparison

Figure 8 . Precision performance comparison

Figure 9 . compares the recall performance of the proposed hybrid methodology with existing approaches, including fuzzy logic with semantic similarity (SS), SS combined with documents, AFINN, hybrid intelligent recommendation System (HIRS), and dictionary-based approaches [23]. They show that the proposed hybrid scheme achieves significantly higher recall than the existing method and verifies its efficiency in accurately obtaining relevant courses.

Figure 10 . represents a comparison of the proposed approach F1-Score, with existing methodologies, including a dictionary-based approach, fuzzy logic + semantic similarity (SS), deep learning, AFINN, hybrid intelligent recommendation (HIRS), SS combined with opinion documents, K-Nearest Neighbors (KNN), and Hybrid BLSTM-

Figure 9 . Recall performance comparison

Figure 10 . F1-score performance comparison

CNN [23]. The proposed approach reaches an F1 score of 99.77% and 99.75% for technical courses and non technical courses respectively, which is significantly higher than other existing methods. These results confirm the effectiveness of the developed model to improve the recommendation of the online course. Further, Figure 11 . demonstrates the comparative evaluation of Kappa metrics across different approaches.

In Figure 11 . KAPPA performance of the proposed hybrid methodology is compared with existing approaches, including IBK, WekaDeepLearning4J, JRip, SMO, Logistic, J48, and Naive Bayes [23]. The proposed approach reaches the Kappa value of 99.08% and 99.09% for technical courses and non technical courses respectively, which is significantly higher than the existing method. These results confirm the reliability and consistency of the presented approach in the course recommendations. In addition, AUC analysis is visualized in Figure 12 .

Figure 11 . Kappa statistics performance comparison

Figure 12 . AUC performance comparison

Figure 12 . The performance of the proposed approach is compared with existing methods, including logistics, J48, Naive Bayes, IBK, Wekadeeplearning4j, JRip, and SMO [23]. The proposed approach reaches an AUC of 100% and 100% for technical courses and non technical courses respectively, which is significantly higher than that of existing methodologies. These results verify the effectiveness of the presented approach in terms of AUC, which ensures improved classification performance. In addition, the change in RMSE performance is examined in Figure 13 .

Figure 13 . The analysis of the performance of RMSE's proposed hybrid methodology is compared with existing approaches, including K-significant, random, Nor-MOOCs, and collaborative filtering [23]. The proposed approach reaches RMSE 0.01 and 0.01 for technical courses and non technical courses respectively, which is significantly lower than that of existing methods. These results show that the presented approach effectively minimizes an error, which is more reliable for the exact recommendations of the course. In addition, a comparison of power on MAE is shown in Figure 14 .

Figure 13 . RMSE performance comparison

Figure 14 . MAE performance comparison

In Figure 14 . The performance of the proposed approach is compared to existing methods. The results show that the proposed approach reaches the lowest MAE, while PMF shows the highest error [23]. This shows the excellent accuracy of the presented methodology to minimize absolute errors in the recommendations of the course. In addition, Figure 15 . represents an evaluation of the HR and ARHR across different methodologies.

Figure 15 . The graphic representation of HR and ARHR is illustrated, comparing the proposed approach with existing techniques such as SVD, NCF, and KNN [23]. The analysis shows that the proposed method overcomes others in HR and ARHR and emphasizes its efficiency in providing accurate and well-included recommendations for the course.

Therefore, as regards several measures for performance, the proposed recommendation system exceeds the comparative existing techniques. The proposed strategy achieves better performance according to experimental analysis with several metrics compared to current methods.

Figure 15 . HR and ARHR performance comparison

**5 Conclusion**

The developed system of e-learning recommendations combines advanced techniques of deep learning, sentiment analysis and optimization methods to provide accurate and personalized course designs. Using models such as hybrid blustm-CNN and XGBOOST, together with the algorithm improved optimization of the horse herd (IHHO) provides exceptional performance across different evaluation metrics, including accuracy, accuracy, induction and F1-score. The system reaches a remarkable rate of accuracy of 99.85% for technical courses and 99.83% for non -technical, and overcomes traditional methods of recommendations. Using a hybrid strategy of collaborative content effectively adapts to users' preferences and learning behavior. The integration of the Vader sentiment analysis increases the ability of the system to interpret user feedback and ensures a deeper understanding of sentiment and users' preferences. In addition, the model shows minimal errors that have shown low RMSE (0.01) and Mae, ensuring reliable and consistent predictions. Future improvements could focus on real -time data updates, include strengthening learning for adaptive recommendations and implementing advanced privacy measures. This system offers comprehensive solutions to strengthen online educational experiences and provides users with accurate and meaningful course recommendations while supporting informed decision -making.

# REFERENCE

# [1] Liu, Tieyuan, Qiong Wu, Liang Chang, and Tianlong Gu. "A review of a deep learning-based recommender system in e-learning environments." *Artificial Intelligence Review* 55, no. 8 (2022): 5953-5980.

# [2] Bhanuse, R., & Mal, S. "A Systematic Review: Deep Learning based E-Learning Recommendation System." *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021, pp. 190-197.

# [3] Aher, Sunita B., and L. M. R. J. Lobo. "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data." *Knowledge-Based Systems* 51 (2023): 1-14.

# [4] Jena, Kalyan Kumar, Sourav Kumar Bhoi, Tushar Kanta Malik, Kshira Sagar Sahoo, N. Z. Jhanji, Sajal Bhatia, and Fathi Amsaad. "E-learning course recommender system using collaborative filtering models." *Electronics* 12, no. 1 (2022): 157.

# [5] Salau, Latifat, Mohamed Hamada, Rajesh Prasad, Mohammed Hassan, Anand Mahendran, and Yutaka Watanobe. "State-of-the-art survey on deep learning-based recommender systems for e-learning." *Applied Sciences* 12, no. 23 (2022): 11996.

# [6] Chanaa, Abdessamad, and Nour-eddine El Faddouli. "Deep learning for a smart e-learning system." In *Proceedings of the 2nd International Conference on Smart Digital Environment*, pp. 197-202. 2022.

# [7] De Medio, Carlo, Carla Limongelli, Filippo Sciarrone, and Marco Temperini. "MoodleREC: A recommendation system for creating courses using the Moodle e-learning platform." *Computers in Human Behavior* 104 (2020): 106168.

# [8] Souabi, Sonia, Asmaâ Retbi, Mohammed Khalidi Idrissi, and Samir Bennani. "Recommendation systems on e-learning and social learning: A systematic review." *Electronic Journal of E-Learning* 19, no. 5 (2021): 432-451.

# [9] Li, Qinglong, and Jaekyeong Kim. "A deep learning-based course recommender system for sustainable development in education." *Applied Sciences* 11, no. 19 (2021): 8993.

# [10] Alatrash, Rawaa, Hadi Ezaldeen, Rachita Misra, and Rojalina Priyadarshini. "Sentiment analysis using deep learning for recommendation in the E-learning domain." In *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2020*, pp. 123-133. Springer Singapore, 2021.

# [11] Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay. "Deep Learning based Recommender System: A Survey and New Perspectives." *ACM Computing Surveys* 52, no. 1 (2019): 1–38.

# [12] Chen, Xiaocong, Lina Yao, Julian McAuley, Guanglin Zhou, and Xianzhi Wang. "A Survey of Deep Reinforcement Learning in Recommender Systems: A Systematic Review and Future Directions." *arXiv preprint arXiv:2109.03540* (2021).

# [13] Klašnja-Milićević, Aleksandra, Mirjana Ivanović, and Alexandros Nanopoulos. "Recommender Systems in E-Learning Environments: A Survey of the State-of-the-Art and Possible Extensions." *Artificial Intelligence Review* 44, no. 4 (2015): 571–604

# [14] Khanal, Suman S., Prasad W. C. Prasad, Asmaâ Alsadoon, and Akram Maag. "A Systematic Review: Machine Learning Based

# Recommendation Systems for E-Learning." *Education and Information Technologies* 25 (2020): 2635–2664.

# [15] Tahir, Sidra, Yasir Hafeez, Muhammad Humayun, Faisal Ahmad, Muhammad Khan, and Muhammad Shaheen. "Harnessing Hybrid Deep Learning Approach for Personalized Retrieval in E-Learning." *PLOS ONE* 19, no. 11 (2024): e0308607.

# [16] Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. “Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores.” Published October 2, 2018.

# [17] Qader, N. N., Moradi, P., Ahmadian, S., & Jalili, M. “Future Generation Computer Systems.” Published September 2020.

# [18] Bhoi, S. K., Malik, T. K., Sahoo, K. S., Jhanjhi, N. Z., Bhatia, S., & Amsaad, F. “E-Learning Course Recommender System Using Collaborative Filtering Models.” Published December 29, 2022.

# [19] Khalid, A., Lundqvist, K., Yates, A., & Ghzanfar, M. A. “Novel Online Recommendation Algorithm for Massive Open Online Courses (NoR-MOOCs).” Published January 22, 2021.

# [20] Lemay, D. J., & Doleck, T. “Predicting Completion of Massive Open Online Course (MOOC) Assignments from Video Viewing Behavior.” Published online March 30, 2020.

# [21] Madani, Y., Ezzikouri, H., Erritali, M., & Hssina, B. “Finding Optimal Pedagogical Content in an Adaptive E-Learning Platform Using a New Recommendation Approach and Reinforcement Learning.” Published October 30, 2019.

# [22] Portuguez Castro, M., & Gómez Zermeño, M. G. “Challenge Based Learning: Innovative Pedagogy for Sustainability through e-Learning in Higher Education.” Published May 15, 2020.

# [23] Bhanuse, R. S., & Mal, S. “Optimal E-Learning Course Recommendation with Sentiment Analysis Using Hybrid Similarity Framework.” Accepted July 3, 2023.

# [24] Madani, Y., Ezzikouri, H., Erritali, M., & Hssina, B. “Finding Optimal Pedagogical Content in an Adaptive E-Learning Platform Using a New Recommendation Approach and Reinforcement Learning.” Accepted October 30, 2019.

# [25] Vedavathi, N., & Kumar, K. M. A. “An Efficient E-Learning Recommendation System for User Preferences Using Hybrid Optimization Algorithm.” Published online May 28, 2021.

# [26] Tarus, J. K., Niu, Z., & Kalui, D. “A Hybrid Recommender System for E-Learning Based on Context Awareness and Sequential Pattern Mining.”

# [27] Algarni, S., & Sheldon, F. “Systematic Review of Recommendation Systems for Course Selection.” Published June 6, 2023.

# [28] Zhang, T., Shaikh, Z. A., Yumashev, A. V., & Chł ąd, M. “Applied Model of E-Learning in the Framework of Education for Sustainable Development.”

# [29] Muzaffar, A. W., Tahir, M., Anwar, M. W., Chaudry, Q., Mir, S. R., & Rasheed, Y. “A Systematic Review of Online Exams Solutions in E-Learning: Techniques, Tools, and Global Adoption.”

# [30] Rosewelt, A., & Renjit, J. A. “A Content Recommendation System for Effective E-Learning Using Embedded Feature Selection and Fuzzy DT Based CNN.”

# [31] Jiang, W., Pardos, Z. A., & Wei, Q. “Goal-based Course Recommendation.” Emails: [jiangwj@berkeley.edu](mailto:jiangwj@berkeley.edu), [pardos@berkeley.edu](mailto:pardos@berkeley.edu), [weiq@sem.tsinghua.edu.cn](mailto:weiq@sem.tsinghua.edu.cn).