[1]Yongjun Zhang[1*]

A Recommender System for Personalized Reading Recommendations and Literature Discovery utilizing the HGRNN-EOO technique

Journal of Electrical Systems

Abstract: - Recommender systems are used to address information overload, enhance personalization, and improve user experience by providing tailored suggestions based on individual preferences, thereby increasing engagement and facilitating content discovery. This paper proposes a hybrid approach for recommender system in personalized reading recommendation and literature discovery. The proposed hybrid approach is the combined performance of both the Hierarchal Gated Recurrent Neural Network (HGRNN) and Eurasian Oystercatcher Optimizer (EOO).Commonly it is named as HGRNN-EOO technique. The major objective of the proposed approach is to provide a recommender system for personalized reading recommendation and literature discovery. HGRNN is designed to provide personalized recommendations based on their preferences, behaviour, and interactions to enhance user experience and engagement. The personalized recommendations from the HGRNN are optimized by using the EOO. By then, the MATLAB working platform has been proposed and implemented, and the present processes are used to calculate the execution. Using performance metrics like accuracy, error rate, F-score, precision, recall, computation time, ROC, sensitivity, and specificity, the proposed method's effectiveness is evaluated. From the result, the proposed approach based error is less compared to existing techniques. The result shows that the accuracy level of proposed Recommender System in Personalized Reading Recommendation using Hierarchal Gated Recurrent Neural Network and Eurasian Oystercatcher Optimizer (RSPRR-HGRNN-EOO) approach is 98% that is higher than the other existing methods. The specificity and the F-score of the proposed RSPRR-HGRNN-EOO approach is 99% and 97%. The error rate of the proposed RSPRR-HGRNN-EOO approach is 2.5%, which is very less compared to other existing techniques. The proposed method shows better results in all existing methods like Recommender System in Personalized Reading Recommendation Convolutional Neural Network (RSPRR-CNN), Recommender System in Personalized Reading Recommendation Deep Neural Network (RSPRR-DNN) and Recommender System in Personalized Reading Recommendation Feed-Forward Neural Network (RSPRR-FNN). Based on the outcome, it can be concluded that the proposed strategy has a lower error rate than existing methods.

Keywords: Reading Recommendation, Literature Discovery, Recommender System, Hierarchal Gated Recurrent Neural Network, Eurasian Oystercatcher Optimizer, Deep learning, Suggestions.

I. INTRODUCTION

A subset of information filtering systems, recommender systems (also called recommender systems or recommendation systems) make suggestions for items based on user relevancy. Deciding which product to choose from a service's possibly bewildering array of options is made much easier with the aid of recommender systems [1]. An artificial intelligence program known as a recommender system makes predictions and offers information or products to users based on their browsing habits and interests. It makes individualized recommendations by using algorithms to examine user data, including demographics, ratings, and previous interactions [2]. Collaborative filtering, which finds trends by comparing users' preferences, and content-based filtering. A collection of distinct, labeled attributes of a thing is used by content-based filtering techniques to recommend additional items with similar properties [3]. This model is then used to anticipate items that the user would be interested in. These are the two primary categories of recommender systems [4]. Hybrid systems that use both of these strategies are also typical [5]. By providing personalized recommendations, recommender systems are widely employed in a variety of businesses, such as social networking, streaming services, and e-commerce, to improve user experience and engagement [6].

Several problems affect the recommender system, among them are data scarcity, data fluctuation, user preferences, unexpected objects, scalability, and privacy protection [7]. Among them are freshness, scalability, sparsity, over-specialization, cold start, and privacy. New product cold start and new user cold start are the two categories of cold start problems [8]. An item may have a cold start issue if there are insufficient prior ratings available for it. Additionally, it can be challenging to suggest products to first-time users because the system lacks knowledge about his previous purchases or because it's probable that he hasn't rated any products yet, making the system inexperienced with his tastes [9]. To deliver the most accurate and optimal recommendations to users, the system requires more resources as the number of items and users increases. Most resources are used to match similar products and consumers based on shared attributes and preferences. A huge amount of users

[1] *Research and External Relations Office, Henan College of Transportation, Zhengzhou, Henan, 451460, China
[1] *Corresponding author Email: zhangyj030505@163.com

and items are in an online shop. If a customer has rated any of the few things they have bought from the store. It will then result in the issue of Lack of abundance [10].

It is also possible to define sparsity as the result of ignorance. In the context of demographic recommender systems, privacy is a significant concern. The system must collect the most pertinent user data, such as location and demographic data, which, to provide the user the most precise recommendation possible, might compromise their privacy [11]. This is one of the most common problems the content-based recommendation system runs into. Something that content-based systems do not offer is diversity in the goods that they propose. Nothing about "surprised" is given. It keeps users from picking up new and interesting information. Products that users are already familiar with are recommended to them. Another problem that recommender systems these days usually address is predictability [12]. Despite the fact that the recommendations sent to the user are varied, it may be well-known to the user. One such approach suggests only the best sellers. In this instance, there are a variety of recommendations. The user could already be aware of or comfortable with the suggested items [13].
These cutting-edge techniques improve the system's capacity to anticipate user preferences and provide more precisely tailored content [14]. To address the cold-start issue, demographic information from social networking sites or the website sign-up page can be utilized [15]. In order to recommend products to a new user, a hybrid strategy also employed that combines collaborative filtering with demographic recommendation [16]. The hybrid recommendation technique can help solve the sparsity issue. Instead of depending only on content-based methods, sparsity can be resolved by fusing collaborative and content-based strategies. Sharing of information could be enhanced by focusing on an item's attributes rather than its physical attributes [17]. In order to provide incredibly precise and customized content or product recommendations, the AI-based recommender can evaluate large and complicated data sets. It does this by learning from user behavior, preferences, and interactions [18]. Recommendation systems have significantly advanced from conventional recommendation approaches with the incorporation of AI. Conventional techniques sometimes depended on rigid algorithms that may recommend products based on overall user patterns or apparent content similarities [19]. By contrast, AI-driven systems can identify small differences and trends that conventional approaches can miss. These systems have the ability to adjust to particular user preferences, which allows them to provide recommendations that are more suited to the needs of each unique user [20].
Major contribution of this paper as follows;

- Hierarchal Gated Recurrent Neural Network optimized with Eurasian Oystercatcher Optimizer for Recommender System in Personalized Reading Recommendation and Literature Discovery (RSPRR-HGRNN-EOO) is proposed.
- Here, the data is collected from the Criteo 1TB Click Logs dataset.
- The dataset sourced from Criteo 1TB Click Logs dataset was employed in conducting the pre-processing process using the Collaborative Filtering Technique. This involved a meticulous analysis of the dataset to identify and choose the most relevant data for subsequent analysis.
- In personalized recommendation, the Hierarchal Gated Recurrent Neural Network (HGRNN) emerges as a prominent technique. HGRNN’s primary function effectively lies in personalized recommendation. It accomplishes this by leveraging the selected attributes derived from theCollaborative Filtering Technique.
- To further enhance the recommendations, HGRNN utilizes theEurasian Oystercatcher Optimizer (EOO) to fine tune and optimize its personalized recommendations accuracy.

The document's remaining sections are organized as follows: The background and current research projects are covered in Segment 2. Segment 3 describes the proposed methodology of Recommender System for Personalized Reading Recommendation and Literature Discovery and provides an illustration for the proposed technique HGRNN-EOO to recommend personalized reading recommendation and literature discovery using hybrid HGRNN-EOO method. In segment 4, the discussion and results are explained. Finally, Segment 5 presents the conclusion.

II. METHODOLOGY

Several works have presented earlier in literatures were relying on recommendation and personalization systems that used the Deep Learning Recommendation Model. Few of them were mentioned here,
Sharma et al. [21] have suggested the semantic personalized recommendation system (SPRS) signifies a noteworthy progression in tackling the difficulties associated with customized recommendations on the internet.

In contrast to traditional systems relying on metadata, the purpose of SPRS aimed to close the semantic gap between low-level content and high-level features by incorporating domain ontology. By recommending personalized sets of videos based on user past activities and utilizing ontology to understand domain concepts, the system aims to improve recommendation accuracy and relevance. Predictive Accuracy Metrics, precision, and recall were employed to assess the system's performance in predicting and aligning with user preferences. Nair et al. [22] have investigated the rise of big data has made it easier for scientific users to access academic articles efficiently, but the increasing volume poses a challenge of information overload for scholars. Article recommendation systems, employing CBF and CF, alleviate this issue by offering personalized suggestions. While the field has progressed, this work introduces a novel approach, the C-SAR model, leveraging GRU in deep learning and the Apriori algorithm in association rule mining. This combined deep learning and classical algorithm approach was to enhance the recommendation accuracy for scholarly articles by focusing on content similarity.

Katzman et al. [23] have suggested the DeepSurv as a Cox proportional hazards deep neural network, offering a cutting-edge method for simulating intricate relationships between patient variables and therapeutic efficacy. Unlike traditional survival models requiring extensive feature engineering, DeepSurv demonstrates the capability to inherently capture intricate relationships, making it a potential tool for personalized treatment recommendations. The study includes experiments on simulated and real survival data, showcasing DeepSurv's performance compared to other models and its ability to model increasingly complex relationships, ultimately enabling medical researchers to leverage deep neural networks for exploring, understanding, and predicting the impact of patient characteristics on the risk of failure.

Guan et al. [24] have suggested the Deep Multi-view Information integration (Deep-MINE), a multi-view recommendation model that incorporates diverse content sources, In response to the growing abundance of pictures on online shopping sites. The model incorporates an integration module to capture interactions among multi-view latent representations, stacked auto-encoder networks that map multiple views of data into a single latent space, as well as a layer of cognition that represents the heterogeneous cognitive styles of the customers. Extensive experiments demonstrate Deep-MINE's effectiveness in accurate product ranking, particularly in cold-start scenarios, and its ability to enhance overall model performance compared to single-view models, highlighting its efficacy in information integration.

Li and J. Kim [25] have investigated the DECOR, due to the rise in demand for online education. Acknowledging challenges in selecting relevant course content due to diverse user knowledge structures. DECOR was designed to alleviate information overload and address high-dimensional data sparsity issues. By capturing intricate user behaviours and course attribute features, the model aims to outperform traditional recommendation approaches, offering a more robust solution for personalized course recommendations on online education platforms.

Jeong et al. [26] have suggested an adaptive recommendation system utilizing deep learning for efficient big data processing to deliver personalized tourism recommendations based on user information and usage patterns. The three-layer architecture of the system consists of the Adaptive Definition Layer, which uses deep learning to identify tourism types that are in line with different personality traits, the Recommendation Service Layer, which provides personalized suggestions, and the Service Provisioning Layer, which facilitates actual user interaction. This design enhances scalability, allowing flexible and user-centric data delivery.

Nitu et al. [27] have suggested the travel recommendation system using users' Twitter data and social connections to tailor personalized suggestions for places of interest. Using a machine learning classifier to detect tweets about travel, the model sets itself apart by adding a time-sensitive recency weight in an effort to identify users' most recent travel preferences. Unlike conventional personalized recommendation systems, this approach focuses on adapting to changing user preferences over time, providing a dynamic and relevant travel recommendation experience.

A. Background of Recent Research Work

A new study reveals the Deep Learning Recommendation Model for Recommendation and Personalization Systems using multiple methods and aspects. The Semantic Personalized Recommendation System (SPRS) relies on domain ontology, introducing potential issues related to ontology accuracy and computational complexity. Using the Gated Recurrent Unit and the Apriori algorithm, the C-SAR model, may encounter challenges in mitigating the cold start problem, impacting its ability to accurately recommend products or users to new users. Deep Surv's data intensity and interpretability issues can hinder its application in the medical domain. Deep Multi-view Information Integration (Deep-MINE) was computationally expensive, dependent on diverse content sources, and sensitive to source reliability. DECOR may grapple with data sparsity and sensitivity to shifts in user behavior. The adaptive tourism recommendation system's efficacy relies on accurate mapping of personality traits to tourism types, potentially limiting exploration. Lastly, the travel recommendation system using Twitter data raises privacy concerns and may be constrained by the reliance on social connections, impacting the model's ability to capture individual preferences effectively. The above mentioned advantages and disadvantages motivated me to do this work.

III. PROPOSED RSPRR-HGRNN-EOO BASED RECOMMENDER SYSTEM FOR PERSONALIZED READING RECOMMENDATION AND LITERATURE DISCOVERY

The recommender system for personalized reading recommendation and literature discovery based on the proposed (RSPRR-HGRNN-EOO) approach is described here. In the propose system the methods used for pre-processing, neural network and optimization are described. Block Diagram for RSPRR-HGRNN-EOO is given bellow in fig 1,

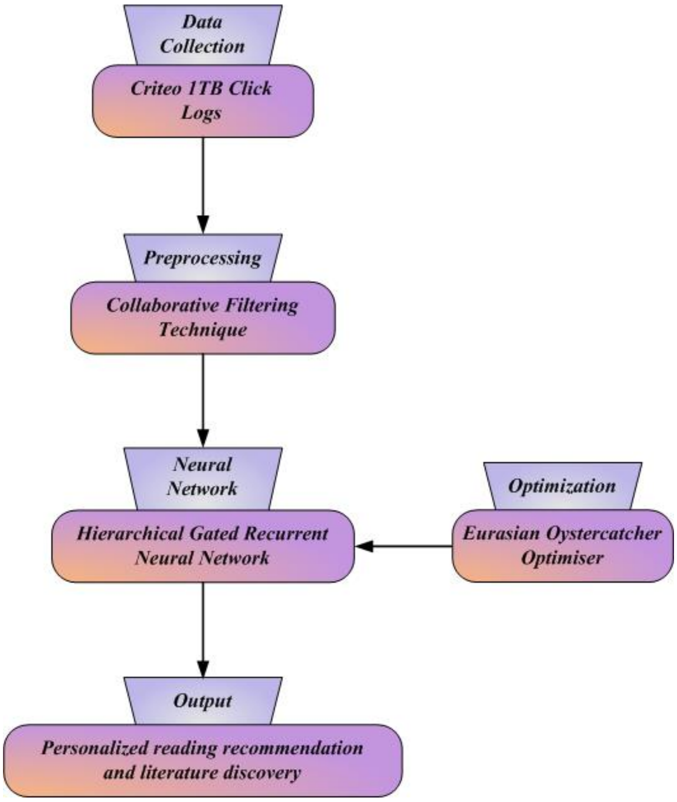

Fig 1: Flowchart of proposed method

A. Data set

In this section, information is gathered from the Criteo 1TB Click Logs dataset [28]. For millions of display advertising, feature values and click feedback are included in this dataset. Its goal is to assess algorithms for the prediction of click through rate (CTR). It is comparable to the Display Advertising Challenge dataset, albeit bigger. A fraction of Criteo's traffic over a 24-day period makes up the training dataset. Every row is a display ad that Criteo has delivered, and the first column shows whether or not the ad has been clicked. To minimize the size of the dataset, the positive (clicked) and negative (non-clicked) samples have both been subsampled, but at varying rates. There are 26 categories features and 13 features that accept integer values (mainly count features). To ensure anonymity, the values of the categorical traits have been hashed onto 32 bits. These features semantics are unknown. There could be missing values in several features. The rows are arranged in chronological order.

B. Pre-processing using Collaborative Filtering Technique

In this section, CF Technique is used [29]. Filtering is used to describe the process of narrowing down and choosing products or information that a user is likely to find interesting. Using the CF method, suggestions are made based on the tastes and actions of users who are similar to you. In the system of recommendations, there is X user in a set $u' = \{u'_1, u'_2, \ldots, u'_x\}$ and a set δ with $o' = \{o'_1, o'_2, \ldots, o'_\delta\}$ big data. The word frequency is given by $TF_{X,\delta}$ is given in equation (1)

$$TF_{X,\delta} = \frac{R_{i',j'}}{\sum_{k\in 0} R_{i',j'}} \tag{1}$$

$$IDF_{oj} = \lg \frac{x}{x_{oj}} \tag{2}$$

Where, $\sum_{k\in 0} R_{i',j'}$ denote the total quantity of times the user has viewed every piece of information in the collection. X_{oj} denote the total number of viewers of the data $o'_{j'}$, IDF_{oj} denote the reverse document frequency, $R_{i',j'}$ denote the quantity of times user $u'_{i'}$ views the data $o'_{j'}$.The user's interest in data content is obtained as equation (3)

$$C_{X,\delta} = \frac{R_{i',j'}}{\sum_{k\in 0} R_{i',j'}} * \lg \frac{x}{x_{oj}} \tag{3}$$

There exist notable variations in the quantity of people perusing distinct categories of data. The quantity of views might be perpetually elevated. In addition, Users may not have any prior experience with the data, but they are also not inherently disinterested in data with no opinions. The purpose Sigmoid function was to normalize the quantity of browsing sessions required by the data function to pique a user's interest. The typical S -type function, Equation (4) provides the sigmoid function.

$$S(x') = \frac{1}{1 + e^{-x}} \tag{4}$$

$$F_{i'j'}{}' = \frac{1}{1 + e^{-\left(R_{i'j'}{}' - R'_{i'}\right)}} \tag{5}$$

Where $R'_{i'}$ indicates the typical frequency with which the user $u'_{i'}$ interface views all of the data viewed, $F_{i'j'}{}'$ indicates the user's functional interest $u'_{i'}$ in the data $o'_{j'}$. Equation (6) predicts the target user's interest in the target data after obtaining the user content and functional interest.

$$pred(u'_{a'}, o'_{q'}) = \overline{F}'_{a'} + \frac{\sum_{u_{b' \in N'_{a'}}{}'} S_{a'b'}{}' \tilde{A} - (F_{b'a'}{}' - \overline{F}'_{a'})}{\sum_{u_{b' \in N'_{a'}}{}'} S_{a'b'}{}'} \tag{6}$$

Where $\overline{F}'_{a'}$ stands for the target user's average functional interest when perusing all of the data. A collaborative filter based on the expected value can be applied to the final big data items after the infinite depth neural network predicted the widespread Web interest in big data.

Thus this method selects the content to be of interest to the user from the input data. Finally, the pre-processed output is given towards neural network segment.

C. Personalized Recommendation using Hierarchical Gated Recurrent Neural Network (HGRNN)

In this section, Hierarchical Gated Recurrent Neural Network (HGRNN) is discussed. HGRNN is made to offer users tailored suggestions according to their likes, actions, and interactions on a platform. This personalization enhances user experience and engagement. HGRNN is specifically designed to handle categorical features efficiently [30]. Many recommendation scenarios involve categorical data, such as user IDs, item IDs, or other discrete attributes. HGRNN's architecture is tailored to address the challenges associated with modelling and learning from categorical features. The implementation of a hierarchical attention strategy results in training parameter redundancy, which increases the risk of over fitting. In order to address the issue of over fitting for the HAN, adversarial and virtual adversarial training are employed to strengthen the HAN classifier's resistance to perturbations in the embedding. Hierarchical Gated Recurrent Neural Network (HAN) variation by the given name is referred. The system dynamics can be calculated in equation (7)

$$E(x_{kt}) = \sum_{t=1}^{T} f_t (x_{kt} - E(x_{kt}))^2 \tag{7}$$

Where, f_t denote the frequency of the t^{th} word, x denote the input, θ denote the parameter of the classifier y. The embedding x_{kt} is replaced by $\tilde{x}_{kt}$ and given in equation (8)

$$\tilde{x}_{kt} = \frac{x_{kt} - E(x_{kt})}{\sqrt{Var(x_{kt})}} \tag{8}$$

Each category feature will be processed by means of an embedding vector of the same dimension, which generalizes the idea of latent components from matrix factorization. A Multi-Layer Perception (MLP) will be used to handle the continuous features. This transformation will produce a dense representation with the same length as the embedding vectors. The loss function L is given in equation (9)

$$L = -\log p(y \mid x + r_{AT}; \theta) \tag{9}$$

$$r_{AT} = \arg\min \log p(y \mid x + r; \tilde{\theta}) \tag{10}$$

Here $\tilde{\theta}$ stands for the classifiers' current constant set of parameters and r_{AT} stands for the adversarial perturbation. Adversarial perturbation of words r_{w-AT} is not applied directly to input, but rather to word embedding. Word sequences make up the hidden representation to which adversarial sentence perturbation r_{s-AT} is applied. A combination of a series of sentence representation vectors (S=S1, S2) and a series of word embedding vectors that are normalized $X = [\tilde{x}_k^1, \tilde{x}_k^2, ..., \tilde{x}_k^T]$. In addition, the classifier y's model conditional probability given S and factor θ as $p(y \mid S; \theta)$ and the classifier y's model conditional probability given X and factor θ as $p(y \mid X; \theta)$. Equation (11), which defines the loss of adversarial with respect to word level,

$$L_{w-AT}(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n \mid (X_n + r_{w-AT,n}); \theta) \tag{11}$$

In terms of sentence level, the adversarial loss is defined by equation (12).

$$L_{s-AT}(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n \mid (X_n + r_{w-AT,n}, S_n + r_{s-AT,n}); \theta) \tag{12}$$

VAT is a novel concept of regularization strategies that are based on local distributional smoothness. The virtual adversarial loss with respect to word level is defined in equation (13)

$$L_{w-VAT}(\theta) = \frac{1}{N} \sum_{n=1}^{N} KL[p(. \mid X_n, S_n; \theta) \,||\, p(. \mid X_n + r_{w-VAT,n}); \theta] \tag{13}$$

Here, N denote the number of examples, d denote a small random vector with TD dimensions, θ denote the parameter of the classifier y. Equation (14) defines the virtual adversarial loss with respect to sentence level.

$$L_{s-VAT}(\theta) = \frac{1}{N} \sum_{n=1}^{N} KL[p(. \mid X_n + r_{w-VAT,n}, S_n + r_{s-VAT,n}; \theta) \,||\, p(. \mid X_n + r_{w-VAT,n}); \theta] \tag{14}$$

Here, N denote the quantity of examples, d denote a small random vector with TD dimensions, θ denote the parameter of the classifier y. Examined is the capacity of virtual and adversarial training at the word and sentence levels.

D. Optimization of HGRNN using Eurasian Oystercatcher Optimizer (EOO)

In this section, the Eurasian Oystercatcher Optimiser (EOO) [31] is used for optimize the gain parameter of HGRNN, which precisely recommends the Personalized recommendation. Initially, EOO creates the uniformly dispersed populace for optimizing the initialization parameters of HGRNN parameters. EOO is used to reduce inference time. By simulating the vulture behaviour, it explores the search space for optimal solutions in recommending the personalized recommendations.

Step 1: *Initialization*

Initialize the input parameter, here the input parameter are the gain parameters of HGRNN which was denoted as r_{AT} .

Step 2: *Random Generation*

After generalization, the FHO generates the mechanism of moving towards the Fire Hawk Optimizer using input parameters chosen at random. The system dynamics can be calculated in equation (15)

$$H = \begin{bmatrix} G_{11} & G_{12} & \ldots\ldots & G_{1n} \\ G_{21} & G_{22} & \ldots\ldots & G_{2n} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ C_1 & C_2 & \ldots\ldots & C_n \end{bmatrix} \tag{15}$$

Step 3: *Fitness Function*

The initialized parameters are determined by the best position that is currently available. Determine the individual's fitness value.

$$fitness\ function = F = Optimizing[r_{AT}] \tag{16}$$

Where, r_{AT} denote the adversarial perturbation.

Step 4: *Exploration Phase*

The main objective of the Eurasian oystercatcher is to maintain a balance between their energy and the mussels' calories. The size of mussels directly affects their energy and calorie content. The length of the mussels, the calories, and the time necessary for openings also rise. As a result, high energy from EO waste is needed. The way that EO acts during the search process is represented in equation (17) and the position of candidate mussel is given in equation (18)

$$Y = T + E + L * r * (X_{best} - X_{i-1}) \tag{17}$$

$$X_i = X_{i-1} * C \tag{18}$$

Here, Y denote the last energy of each cycle, X_i denote the potential mussel's position, L denote the mussel's length, which represents the range of the ideal length and is a random number between 3 and 5, T denote the amount of time needed to open the present mussel, E denote the energy of the EO at the time, r denote the random number range [0, 1] to increase unpredictability and find new locations within the search area, C denote the caloric value that depend on the length of the mussel.

Step 5: *Exploitation Phase*

Equation (19) provides the time needed to open the current mussel.

$$T = \left(\left(\frac{L-3}{5-3} \right) * 10 \right) - 5 \tag{19}$$

Where, T denote the duration needed to open the current mussel, L denote the mussel's length, which represents the range of the ideal length and is a random number between 3 and 5.

$$E = \left(\frac{i-1}{n-1} \right) - 0.5 \quad Where\ i > 1 \tag{20}$$

Where, E denote the energy of the EO at the time, i represents the iteration value,

$$C = \left(\left(\frac{L-3}{5-3} \right) * 2 \right) + 0.6 \tag{21}$$

Where, C denote the caloric value depend on the length of the mussel, L denote the mussel's length, which is a arbitrary number between 3 and 5 that indicates the range of ideal length. The value from equation 21 is displayed in the range of 0.6 to 0.8, while the value from equation 19 is displayed in the range of 5 to -5. Trial and error was used to determine these values. It's crucial to note that if the time is negative, this indicates that the mussels may take longer to break than the bird can, and vice versa. This is thought to be a limitation. The obtained E value, where I is the iteration's value, dropped linearly from 0.5 to -0.5, which begins with the number of iterations and ends with one. The E value will be fixed in the final two iterations. As a result, T and E , the amount of time and energy needed to open the candidate mussel, may be negative values. Equation (17) and equation (18) depend on L , a random variable that fluctuates randomly, for the T and C values, respectively. Because it prevents a local minimum problem and enables EO to arrive at any location within the search area, this condition emphasizes exploration every time. Using the mussel's opening time, which is a measurement based on the bird's energy and the mussel's size one can improve the quality of selection by determining the expected location of the target food. At every iteration, the random values entered during optimization aid in the exploration of new territory. Avoid a local minimum issue as a result. Fig 2 shows that Flow chart of Eurasian Oystercatcher Optimizer

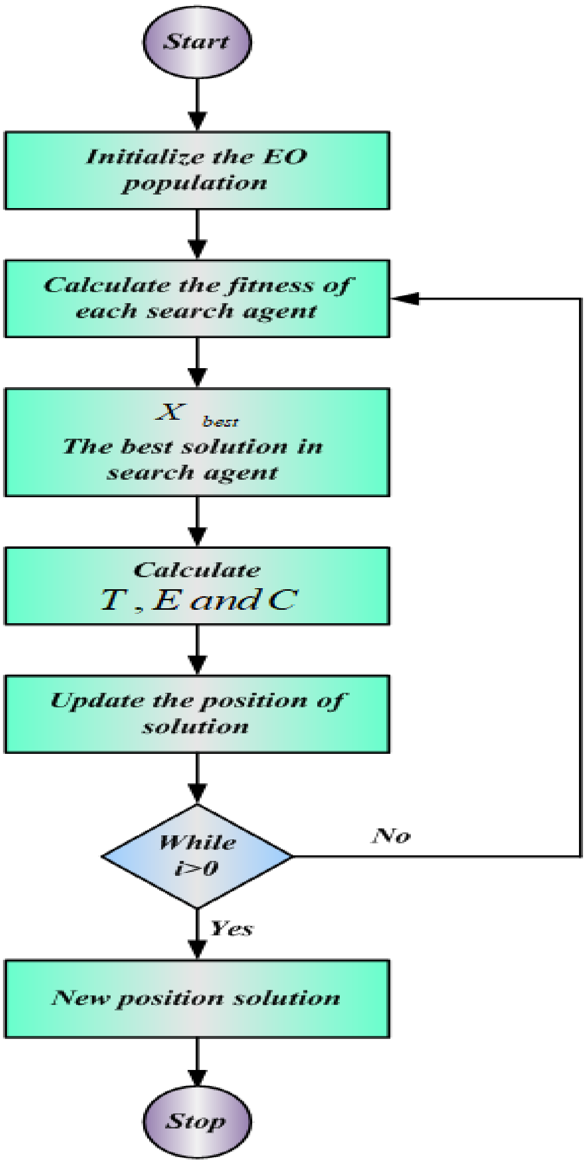

Fig 2: Flow chart of Eurasian Oystercatcher Optimizer

Step 6: *Termination Criteria*

Finally, EOO optimizes the factor iteratively repeat till it reaches halting criteria $x = x + 1$. CF effectively recommends the personalized reading recommendation and literature discovery. Finally, the proposed HGRNN-EOO found out the reading recommendation and literature discovery model gets successful results. Verify the end criteria and if the best result is achieved then process is end or else go to step 3.

IV. RESULTS AND DISCUSSIONS

In this section a hybrid RSPRR-HGRNN-EOO technique for personalized reading recommendation and literature discovery is presented in this research. The hybrid technique that has been suggested combines Recommender System for Personalized Reading Recommendation (RSPRR) the capabilities of the Hierarchical Gated Recurrent Neural Network (HGRNN) and Eurasian Oystercatcher Optimizer (EOO). It is commonly referred to as the RSPRR-HGRNN-EOO approach. The proposed method's main goal is to provide recommender system for personalized reading recommendation and literature discovery.

A. Performance measures

This section describes the proposed approach's performance in light of the simulation's results. To estimate the recommender system for personalized reading recommendation and literature discovery, in this paper utilize the hybrid RSPRR-HGRNN-EOO approach. The objective of the proposed method helps in recommending personalized reading recommendation and literature discovery. Performance metrics including accuracy, error rate, F-score, precision, recall, computation time, ROC, sensitivity, and specificity are analyzed to assess the performance.

1) Accuracy

For a given dataset, it is the proportion of accurately predicted entries to all of the predictions. Equation (22) is used to measure it.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{22}$$

2) Computational time

The execution time of an algorithm can only be determined by its time complexity, which is totally dependent on the algorithm and its inputs. The computational complexity indicates how long an algorithm takes to run. This is scaled by equation (23)

$$CPU\ Time = {IC * CPI}/{Clockrate} \tag{23}$$

3) ROC

Equation (24) provides the ratio of the false negative to the true positive area.

$$ROC = 0.5 \times \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right) \tag{24}$$

4) Precision

The accuracy of a machine learning model's positive prediction is one metric used to assess the performance of the model, along with accuracy. Precision is defined by equation (25), which calculates the ratio of true positives to the total amount of positive predictions.

$$\mathrm{Pr}ecision = \frac{TP}{TP+FP} \tag{25}$$

5) Recall

A machine learning model's recall, the proportion of data samples that the model accurately categorizes as being in a class of interest, also known as the TPR. It is measured by following equation (26).

$$\mathrm{Re}call = \frac{TP}{TP+FN} \tag{26}$$

6) F-Score

One metric used to assess a machine learning model's performance is the F-score. Equation (27), which combines recall and precision into a single score, describes it.

$$F - Score = 2 * \frac{precision * recall}{precision + recall} \tag{27}$$

7) Error Rate

One less accuracy equals the error rate. A model with 90% accuracy would have a 10% error rate. Equation (28) is used to calculate it.

$$Error\,Rate = 1 - \frac{FP+FN}{TP+TN+FP+FN} \tag{28}$$

8) Sensitivity

The metric used to assess a model's predictive power for true positives in each available category is called sensitivity. You can use these metrics with any categorical model. You can find it in equation (29).

$$Sensitivity = \frac{TP}{TP+FN} \tag{29}$$

9) Specificity

The metric used to assess a model's capacity to forecast true negatives for every category that is available is called specificity. You can use these metrics with any categorical model. It is given in equation (30).

$$Specificity = \frac{TN}{TN + FP} \tag{30}$$

B. Performance Analysis

Figure 3 to 11 depicts simulation results of proposed RSPRR-HGRNN-EOO method. Then the proposed RSPRR-HGRNN-EOO method is likened to existing RSPRR-CNN, RSPRR-DNN and Feed-Forward Neural Network (RSPRR-FNN) methods.

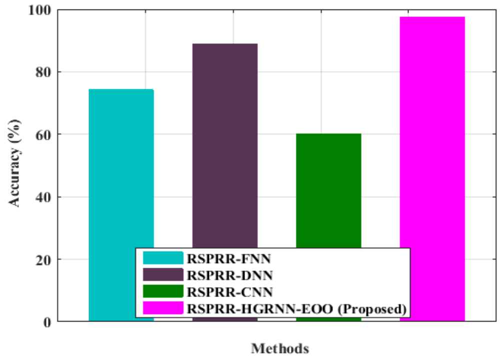

Fig 3: Comparison of proposed method accuracy with existing methods

Fig 3 depicts the Comparison of proposed method Accuracy with existing methods. The accuracy of RSPRR-FNN is 75% and the accuracy of RSPRR-DNN is 90% and the accuracy of RSPRR-CNN is 60%. Then the accuracy of proposed method RSPRR-HGRNN-EOO is 98%. It shows that the proposed method have better accuracy than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

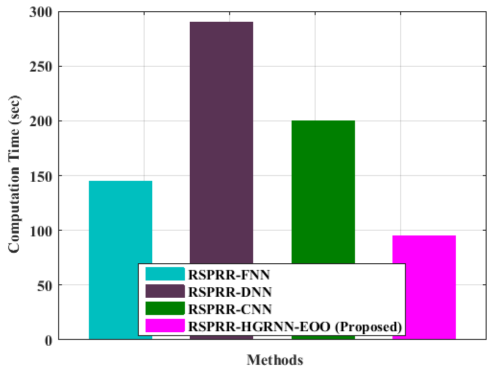

Fig 4: Comparison of proposed method Computation time with existing methods

Fig 4 depicts the comparison of proposed method computation time with existing methods. The computation time of RSPRR-FNN is 145 sec and the computation time of RSPRR-DNN is 290 sec and the computation time of RSPRR-CNN is 200 sec. Then the computation time of proposed method RSPRR-HGRNN-EOO is 95 sec. It shows that the proposed method have efficient computation time than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

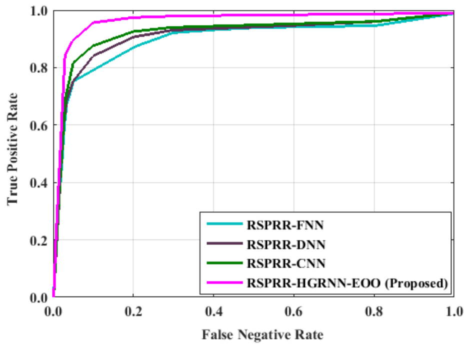

Fig 5: Comparison of proposed method ROC with existing methods

Fig 5 depicts the Comparison of proposed method ROC with existing methods. The ROC of RSPRR-FNN is 0.9 attains at 0.3 false negative rate and the ROC of RSPRR-DNN is 0.9 attains at 0.3 false negative rate and the ROC of RSPRR-CNN is 0.9 attains at 0.3 false negative rate. Then the ROC of proposed method RSPRR-HGRNN-EOO is 0.95 attains at 0.1 false negative rate. It shows that the proposed method have better ROC than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

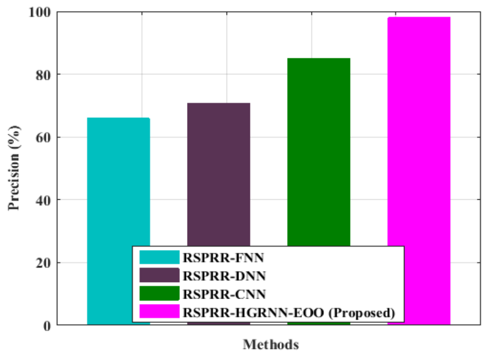

Fig 6: Comparison of proposed method precision with existing methods

Fig 6 depicts the Comparison of proposed method precision with existing methods. The precision of RSPRR-FNN is 65% and the precision of RSPRR-DNN is 70% and the precision of RSPRR-CNN is 85%. Then the precision of proposed method RSPRR-HGRNN-EOO is 98%. It shows that the proposed method have better precision than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

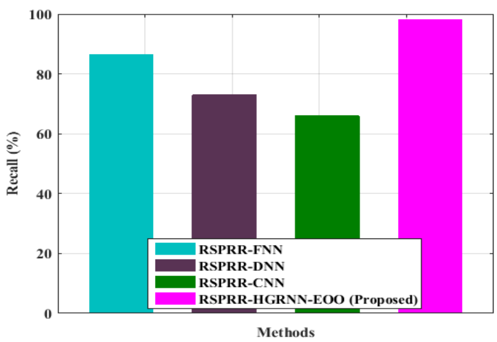

Fig 7: Comparison of proposed method recall with existing methods

Fig 7 depicts the Comparison of proposed method recall with existing methods. The recall of RSPRR-FNN is 85% and the recall of RSPRR-DNN is 75% and the recall of RSPRR-CNN is 65%. Then the recall of proposed method RSPRR-HGRNN-EOO is 98%. It shows that the proposed method have better recall than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

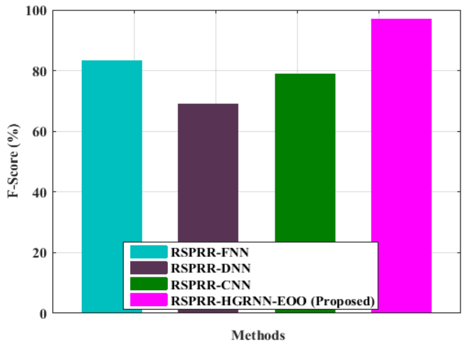

Fig 8: Comparison of proposed method F-Score with existing methods

Fig 8 depicts the Comparison of proposed method F-Score with existing methods. The F-Score of RSPRR-FNN is 83% and the F-Score of RSPRR-DNN is 70% and the F-Score of RSPRR-CNN is 78%. Then the F-Score of proposed method RSPRR-HGRNN-EOO is 97%. It shows that the proposed method have better F-Score than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

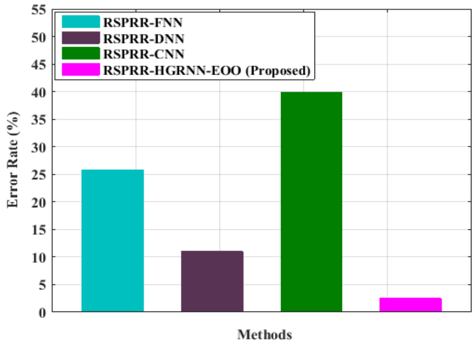

Fig 9: Comparison of proposed method error rate with existing methods

Fig 9 depicts the Comparison of proposed method error rate with existing methods. The error rate of RSPRR-FNN is 26% and the error rate of RSPRR-DNN is 11% and the error rate of RSPRR-CNN is 40%. Then the error rate of proposed method RSPRR-HGRNN-EOO is 2.5%. It shows that the proposed method have lower error rate than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

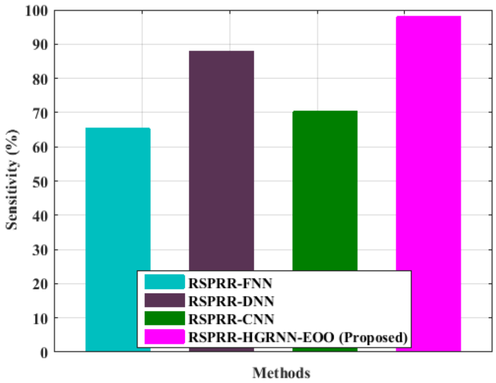

Fig 10: Comparison of proposed method sensitivity with existing methods

Fig 10 depicts the Comparison of proposed method sensitivity with existing methods. The sensitivity of RSPRR-FNN is 65% and the sensitivity of RSPRR-DNN is 88% and the sensitivity of RSPRR-CNN is 70%. Then the sensitivity of proposed method RSPRR-HGRNN-EOO is 98%. It shows that the proposed method have better sensitivity than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

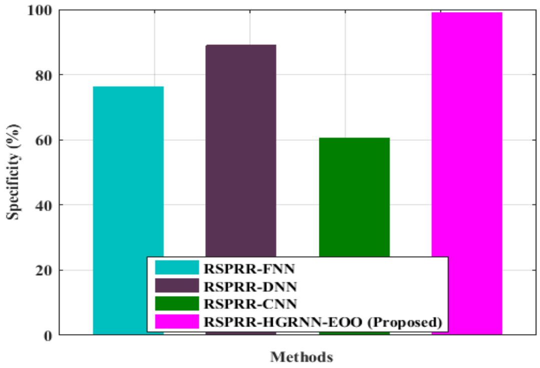

Fig 11: Comparison of proposed method specificity with existing methods

Fig 11 depicts the Comparison of proposed method specificity with existing methods. The specificity of RSPRR-FNN is 76% and the specificity of RSPRR-DNN is 88% and the specificity of RSPRR-CNN is 61%. Then the specificity of proposed method RSPRR-HGRNN-EOO is 99%. It shows that the proposed method have better specificity than RSPRR-FNN, RSPRR-DNN and RSPRR-CNN.

V. CONCLUSION

This study is used for the recommender system in personalized reading recommendation and literature discovery using the RSPRR-HGRNN-EOO approach. Recommender systems in personalized reading recommendations and literature discovery are used to address information overload, enhance personalization, and improve user experience by offering customized recommendations based on personal tastes, thereby increasing engagement and facilitating content discovery. The proposed strategy is assessed using the MATLAB Simulink platform and compared to various other approaches that are currently in use. The proposed method is evaluated in a wide range of scenarios, including optimal and random scheduling as well as a complex EOO algorithm. The outcome indicates that, in comparison to current methods, the proposed approach's based error is lower. The result shows that the accuracy level of proposed RSPRR-HGRNN-EOO approach is 98% that is higher than the other existing methods. The proposed RSPRR-HGRNN-EOO approach has a specificity and F-score of 99% and 97%, respectively. Comparing the proposed RSPRR-HGRNN-EOO approach to other methods currently in use, the error rate is remarkably low at 2.5%. Comparing the proposed approach-based recommendation system against other methods, the results show that it is accurate. The goal of the proposed model is to recommend for the personalized reading recommendation and literature discovery. The results also show that compared to the other optimization techniques, the proposed approach performs noticeably better.

REFERENCES

[1] Quadrana, M., Cremonesi, P., & Jannach, D. (2018). Sequence-aware recommender systems. *ACM computing surveys (CSUR)*, *51*(4), 1-36.

[2] Alhijawi, B., & Kilani, Y. (2020). A collaborative filtering recommender system using genetic algorithm. *Information Processing & Management*, *57*(6), 102310.

[3] Aljunid, M. F., & Dh, M. (2020). An efficient deep learning approach for collaborative filtering recommender system. *Procedia Computer Science*, *171*, 829-836.

[4] Gazdar, A., & Hidri, L. (2020). A new similarity measure for collaborative filtering based recommender systems. *Knowledge-Based Systems*, *188*, 105058.

[5] Bobadilla, J., Alonso, S., & Hernando, A. (2020). Deep learning architecture for collaborative filtering recommender systems. *Applied Sciences*, *10*(7), 2441.

[6] Zhou, K., Zhou, Y., Zhao, W. X., Wang, X., & Wen, J. R. (2020). Towards topic-guided conversational recommender system. *arXiv preprint arXiv:2010.04125*.

[7] Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. (2019). Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*.

[8] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *applied sciences*, *10*(21), 7748.

[9] Etemadi, M., Abkenar, S. B., Ahmadzadeh, A., Kashani, M. H., Asghari, P., Akbari, M., & Mahdipour, E. (2023). A systematic review of healthcare recommender systems: Open issues, challenges, and techniques. *Expert Systems with Applications*, *213*, 118823.

[10] Cui, Z., Xu, X., Fei, X. U. E., Cai, X., Cao, Y., Zhang, W., & Chen, J. (2020). Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Transactions on Services Computing*, *13*(4), 685-695.

[11] Iqbal, M., Ghazanfar, M. A., Sattar, A., Maqsood, M., Khan, S., Mehmood, I., & Baik, S. W. (2019). Kernel context recommender system (KCR): A scalable context-aware recommender system algorithm. *IEEE Access*, *7*, 24719-24737.

[12] Ferrari Dacrema, M., Boglio, S., Cremonesi, P., & Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, *39*(2), 1-49.

[13] Feng, C., Liang, J., Song, P., & Wang, Z. (2020). A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences*, *521*, 365-379.

[14] Li, J., Lu, K., Huang, Z., & Shen, H. T. (2019). On both cold-start and long-tail recommendation with social data. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 194-208.

[15] Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, *149*, 113248.

[16] Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, *7*, 439-457.

[17] Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, *92*, 507-520.

[18] Chen, T. T., & Lee, M. (2018). Research paper recommender systems on big scholarly data. In *Knowledge Management and Acquisition for Intelligent Systems: 15th Pacific Rim Knowledge Acquisition Workshop, PKAW 2018, Nanjing, China, August 28-29, 2018, Proceedings 15* (pp. 251-260). Springer International Publishing.

[19] Zhang, F., Qi, S., Liu, Q., Mao, M., & Zeng, A. (2020). Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. *Expert Systems with Applications*, *149*, 113346.

[20] Anthony Jnr, B. (2021). A case-based reasoning recommender system for sustainable smart city development. *AI & SOCIETY*, *36*(1), 159-183.

[21] Sharma, S., Rana, V., & Kumar, V. (2021). Deep learning based semantic personalized recommendation system. *International Journal of Information Management Data Insights*, *1*(2), 100028.

[22] Nair, A. M., Benny, O., & George, J. (2021). Content based scientific article recommendation system using deep learning technique. In *Inventive Systems and Control: Proceedings of ICISC 2021* (pp. 965-977). Springer Singapore.

[23] Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology*, *18*(1), 1-12.

[24] Guan, Y., Wei, Q., & Chen, G. (2019). Deep learning based personalized recommendation with multi-view information integration. *Decision Support Systems*, *118*, 58-69.

[25] Li, Q., & Kim, J. (2021). A deep learning-based course recommender system for sustainable development in education. *Applied Sciences*, *11*(19), 8993.

[26] Jeong, C. S., Lee, J. Y., & Jung, K. D. (2020). Adaptive recommendation system for tourism by personality type using deep learning. *International Journal of Internet, Broadcasting and Communication*, *12*(1), 55-60.

[27] Nitu, P., Coelho, J., & Madiraju, P. (2021). Improvising personalized travel recommendation system with recency effects. *Big Data Mining and Analytics*, *4*(3), 139-154.

[28] https://ailab.criteo.com/download-criteo-1tb-click-logs-dataset/

[29] Zhang, Z. (2021). A method of recommending physical education network course resources based on collaborative filtering technology. *Scientific Programming*, *2021*, 1-9.

[30] Poon, H.K., Yap, W.S., Tee, Y.K., Lee, W.K. and Goi, B.M., 2019. Hierarchical gated recurrent neural network with adversarial and virtual adversarial training on text classification. *Neural Networks*, *119*, 299-312.

[31] Salim, A., Jummar, W. K., Jasim, F. M., & Yousif, M. (2022). Eurasian oystercatcher optimiser: New meta-heuristic algorithm. *Journal of Intelligent Systems*, *31*(1), 332-344.