International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Location Aware Content Priority based Recommendation System Flying Squirrel Optimization - Deep Alternative Neural Network (FSO-DANN)

P. Christopher Arokiaraj[1], Dr. D. Hari Prasad[2]

Submitted: 23/12/2023 **Revised**: 29/01/2024 **Accepted**: 07/02/2024

Abstract: Social networks collect a lot of customer information, and this data may be utilized to develop knowledge for a variety of mobile and web applications. The Recommendation System (RS) is a domain garnering much attention these days. There are numerous itinerary and location aware content in RS available right now, some of that are nearly exclusively business. However, a thorough analysis demonstrates the need for study and advancement in this field. The data supporting this study show that the majority of systems are essentially destination RSs, and the great majority do not dynamically build routes but instead require the customer to choose appropriate locations. Some need greater user involvement while others fail to account for the length of presence at the selected sites. In certain frameworks, the community finding technique was ineffective, while in others, the routes are not the best. The RS was developed to fill the holes in the existing itinerary RS that were discovered through a thorough analysis. A Flying Squirrel Optimization - Deep Alternative Neural Network (FSO-DANN) based location RS was in charge of giving customers to get the priority of the best location. A backtracking-based approach is implemented in the genetic algorithm-based itinerary planning component to create itineraries. The Hadoop Map Reduce programming method was used to build the system in parallel. An extensive investigation of the system's assessment reveals that it is effective and competent enough to offer a trip schedule to a user that was better suitable in 25% less time than existing systems.

Keywords: *Recommendation System; Location Aware Content; Neural Network; Optimization; Priority based System; Performance Measures*

1. Introduction

Image Classification in satellite pictures assumes a crucial function in removing and deciphering helpful data from a gathering of pictures. The arrangement is essentially the forecast of pictures that has a place with a particular class. While anticipating the number of pictures having a place with a specific class or not relies upon our arrangement execution, it tends to be finished by utilization of subtitle words to anticipate whether a picture has a place in a particular class. Without past data an ordinary picture arrangement undertaking would require the investigation of a larger than usual and advanced component space.

The demonstrated position RS combines the best elements of content-based and CF-based methods to create a hybrid RS [1]. It makes use of implied consumer reviews for places. Recommendations are improved by taking into account user tips or feedback on various locations. This stand-alone location RS was designed to provide consumers with recommendations made up of specific locations that meet their tastes, such as restaurants, popular destinations, or cities [2].

The system's development aims to minimize user involvement in recommendations [3]. The implicit evaluations provided by customers in a group that best reflects the user for whom the recommendation was to be made are used to determine how to make the advice [4]. Based on a person's behaviors and location history, this approach offers a destination recommendation. Additionally, it helps to create individualized or tailored gatherings where the recommendation was strengthened through the variety of input sources or preferences [5].

[1, 2] Sri Ramakrishna College of Arts & Science, Coimbatore, India
** Corresponding Author Email: christo4u2005@gmail.com*

Each has advantages and disadvantages. The proposed approach addresses many issues with the current methodologies, such as cold start issues, data sparsity, over specialization, and flexibility [6]. A cold start issue arises when there are new individuals introduced to the system for which little data was given, and the meager information made accessible was hardly sufficient for the system to provide recommendations [7]. When a person who previously liked a location is recommended the same one, overspecialization has taken place. The recommended approach can manage each of these issues [8].

2. Related Works

The approach employed in the thesis employs a hybrid methodology for location recommendations, integrating content-based and collaborative filtering-based methods to offer better recommendations and, in addition, manage recommendations to inexperienced consumers [9] .The system is hence resistant to user cold start issues while still needing to retain user profiles. The approach also takes into account generalized local opinion in some areas [10]. The method encourages serendipity and lessens the effects of overspecialization by taking into account nearby destinations that are similar to those recently attended when making recommendations [11].

To determine which of the ranked sites the customer might want to visit, NN are implemented [12]. There are four levels of neurons in the NN. The provided input in the form of an attribute vector was accepted by the input layer. A hidden layer that receives input from the input level is the following layer [13-15]. This phase processes learning by calculating the variation between the input and the learning vectors. The output from this level is sent to the summing layer, which is in charge of increasing the likelihood that features will be classified into different categories. The summation level totalizes the differential values [16]. The output level places the specific input set in the appropriate class and suggests it to the consumer.

Locations will constitute the output classes when NNs used to recommend locations. The NN will accurately group people into suitable areas, where they could be advised to go. The customer can proceed with this recommendation if they just want one location proposed. One or more context-dependent sites can be included in the system's recommendations [17]. This is a lesser-known variation of the ultimate system. Figure 1 depicts the NN's infrastructure for site recommendation

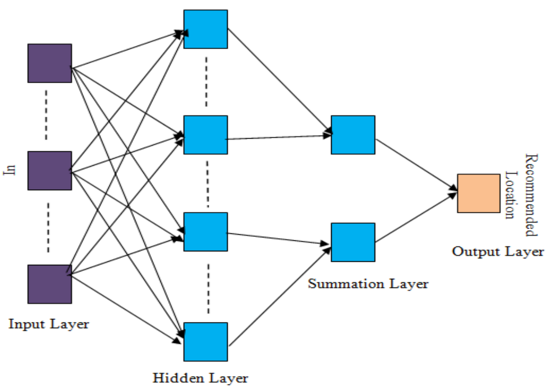

Fig 1: Location Aware RS recommendation using Neural Network architecture

3. Proposed System

This section presents the recommended DANN-based location recommendation method's step-by-step processing. The recommendation engine was created to match user tastes with locations based on user ratings for visits and tips. The position suggestion algorithm first calculates and ranks the candidate destinations. These selected sites are then supplied to a DANN algorithm, which selects appropriate destinations to suggest to the consumer.

Communities identified during the community identification stage serve as the system's input. The algorithm then locates candidate places, or possible locations for recommendations. These are either places that locals frequent or places that are equivalent to those places. Based on the state of visits there, candidate locations are given weights. A location that has been visited or is comparable to a location that other community members have visited and that the individual has visited receives the highest priority, with a maximum weight of 0.4. Locations that are not frequented by the consumer but are frequented by other community members are given a weight of 0.3. A location that is frequented by the user and one that is the same as or similar to one frequented by other community members will receive a weight of 0. A weight of 0.1 is given to a place that a user has only ever visited. This weight distribution aids in addressing the issue of over specialization. For instance, users A and B might have gone to the movie theatre, then the restaurant, then the museum, and then the cafeteria. They have visited the same places in the same city in the same order, but at various theatres, eateries, museums, and cafeterias that are also nearby. This means that rather than just using identical locations, related sites should also be taken into account when making location recommendations.

The sentiment score and relevance of user recommendations or remarks on visited places are computed. Based on these ratings, a weight is given to each place, and the rankings of the locations are then determined by these weights. It has been noted that customers can occasionally choose the top-ranked recommended location. Although the system calculates a specific location as the one to be recommended to the consumer, he or she is free to choose a different location and utilize the recommended location only as a suggestion. A DANN classifier is implemented into the location suggestion system to handle such circumstances. User id, context, proposed location, and accepted location make up the input vector for the DANN classifier. This is done so that the classifier can accurately categorize the input for a location. The idea is that, given a context and the proposed location, the classifier can determine which location he may be thinking of. For this, the classifier has been well-trained using historical data. As a result, the system must keep track of previous user consent. The DANN classifier may increase system overhead, but it will significantly increase the recommendation's precision.

In this stage, the system is made to be context-aware, which makes it more appealing and appropriate. The customer is asked to specify the context in which the recommendation is desired, and as a result, the customer's choices may change. The DANN classification was well-suited to handle this situation. The system only takes into account a small number of contexts, but it is anticipated that these contexts are sufficient to fulfill the needs of professional customers.The recommendation engine has been tuned to deliver more precise recommendations in a variety of circumstances. Figure 2 displays a block diagram of the location RS.

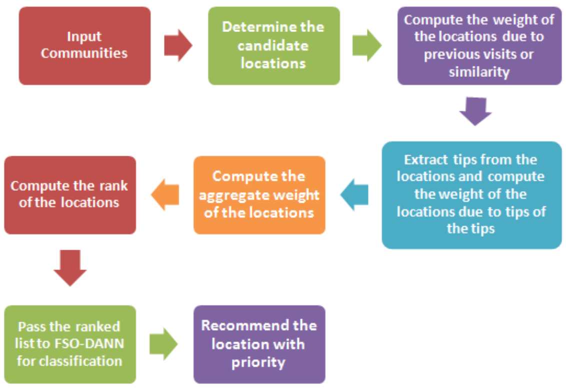

Fig 2: Proposed FSO-DANN-based location aware content priority

The key techniques used for location recommendation are covered in the sections that follow.

Algorithm 1: Location_Recommendation()

Input: (1) Com_{u_r} – Community of u_r (2) u_r – userid of user

Output: Location aware content recommend for user u_r

Steps

1. Access Com_{u_r} // The community of u_r
2. $L_c \leftarrow$ Compute_candidate_locations(Com_{u_r})
3. $(L, L_w) \leftarrow$ Compute_locationWeight(L_c)
4. $L_R \leftarrow Sort(L, L_w)$
5. Pass L_R to Itinerary recommendation phase for itinerary computation
6. Ranked the locations and DANN to find and recommend the location based on priority
7. end

The primary method used in location suggestion is called Location_Recommendation. The major input for suggesting destinations to the consumer was the output from the user community during the community discovery stage. The candidate sites are computed by the technique. The Compute_Location using FSO-DANN method gives them weights based on their locations. To determine the best suitable site to recommend to the user, the candidate locations are ordered according to the weights and provided to the DANN classification as an input vector.

Algorithm 2: FSO-DANN

Input: (1)S_x //Split (2) Com_{u_r} – the Community of u_r (3) u_r – userid

Output: Pair of key element <key, 1> for choosing the candidate locations

Step 1: Map the key of writable; Text (Iterable); context

Step 1.2: Assign the Tokens to value.toString().split(\t)

Step 1.3: Assign userid to Tokens[0]

Step 1.4: Assign locationid to Tokens[1]

Step 2: for each u' ε Com_{u_r}

{

Step 2.1: Assign loc to extractLocations(u')

Step 3: for each l in loc

{

contect.write(l,1)

}

}

3.1 Candidate Location Generation

CL is those that have been visited by comparable to, at least one member of the group to which the user participates. These places are taken from the community members' location histories and are listed as candidate places. These places should resemble the ones the person for whom the suggestion is to be made has visited rather than being an exact match. This is based on the idea that users will be more familiar with previously visited places than new ones, thus they won't need to be recommended to them again. Additionally, this assists in adding new destinations to the recommendation in addition to already visited ones. According to studies, more than 75% of people prefer to travel to new places rather than places they have already been. The MapReduce method is also used to do this operation, making it faster to complete than a traditional technique. The following table contains the technique for locating potential locations.

The prospective sites for recommendations are computed using the Compute_candidate_locations function. The customer's and his or her friends' most recent visits to certain sites, and comparable locations, make up a set of candidate locations. The system's location-similarity matrix can be used to identify sites that are related to one another.The matrix, which was initially created by the system offline, is updated dependent on whether the user accepts the recommendations. The overhead associated with managing this matrix rises, but the variety of recommendations is significantly greater and outweighs the overhead. The MapReduce framework is utilized to accomplish the method.

3.2 Location Ranking

In this part, a Map Reduce method is also used to execute the time-consuming task of location ranking. As part of this task, tips submitted by users in potential areas are examined, and the sentiment score and decay are computed. A tip's decay is the gradual decline in its usefulness over time. The decay is multiplied by the sentiment score, totaled for each place, and the value is normalized over a predetermined period. Considering the suggestions provided by the customer's friends, the combined weight is considered as the weight of the place. Every place is given a weight based on its status as a visit by the individual or his or her friends and its proximity to other places that community members have already visited. When the two weights indicated above are summed, the location's weight is obtained. Similar destinations are taken into account to give the customer the best itinerary or selection of locations, rather than the identical ones chosen by his or her buddy. A parallel algorithm is used to rank the locations according to the weights. This decreases the amount of time the system takes to execute.

3.3 Tip Extraction

Customers who have previously visited a place may provide relevant recommendations for those who are considering going there. Depending on the users' individual experiences in various regions, tips could be either positive or bad. Recommendations would be more precise if real tips were taken into consideration. Check-in logs are used to extract location recommendations. The system could have features that allow consumers to submit tips explicitly.

3.4 Sentiment score computation

The polarity of a tip was indicated by its sentiment score. Due to many causes, consumers may have had a poor experience visiting a visually appealing location, which may be reflected in their reviews. If these suggestions are ignored, the algorithm can suggest to aspirants areas that are perfect but not selected. The technology gathers user-submitted tips or feedback and independently calculates the sentiment score of all remarks on every website. The sentiment score estimation was implemented using Senti Strength v2.2 and might very well be included in the proposed location RS.

3.5 Decay computation

A tip's or comment's decay value is its value diminishing with time. The essential premise is that the consumer might have provided feedback—positive or negative—about a location at a specific time as a result of his or her interactions there. According to the input, the relevant authorities may have taken either favorable or unfavorable action. This is taken into account when assuming that the tip's usefulness will wane with time. In this work, the decay is computed over two years, however, it may change according to the situation and environment.

3.6 FSO-DANN Classification

The FSO-DANN classification is the final step. It has been discovered that a small percentage of users visit a place other than the one that was proposed, which lowers the RS accepting rate. As a result, the recommendation issue has been transformed into a classification issue where the output classes are the locations. The location names of these places are listed separately. The FSO-DANN uses the ranking list of locales as its input and outputs the place the user is most likely to go. In comparison to its rivals, the FSO-DANN trains more quickly and has much higher classification accuracy.

Context awareness is presumed for location recommendations. As a result, the system approves the user's request for the recommendation's context. Depending on the situation, the advice will change. The quadruple f = "user-id, context," "recommend_location," and "accepted_location" serves as the DANN's input feature vector. The DANN categorization is incorporated into the system with the understanding that not all users would always choose or accept proposed places. The system records previous recommendations and choices made by consumers and their fellow community members. This vectorized historical dataset is employed to teach the DANN. The most appropriate site is proposed to consumers using a certified DANN algorithm, thereby strengthening the recommendation.

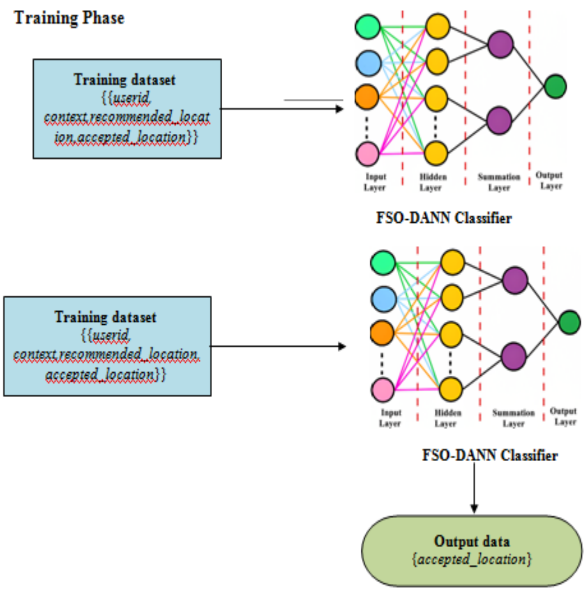

Fig 3: FSO-DANN-based location recommendation

Optimized Location Recommendation

The customer is given recommendations for the best places to go. In general, the FSO-DANN classification could be iterated to increase the number of places included in the suggestion. The output position could be identified and left out for the following iteration during every iteration.

4. Experiments and Analysis

Data Acquisition

The information used to recommend locations comes from the stage of community exploration. The location recommendation phase receives input from user communities—groups of users with similar interests, preferences, location histories, and past behaviors—that were found in the first stage. The system starts the implicit rating process. In other words, if a person has visited a place, it is assumed that they have given it a one-star rating. Recurring trips to the same place raise its score.

Performance Analysis

Execution time and F-measure are metrics used to assess the system's effectiveness. Separate analyses are done for the location ranking and the site recommendation. The FSO-DANN location ranking method is contrasted with two other location ranking algorithms that the authors have earlier recommended.

Experimental Results

Table 1 displays the process timings for several location ranking methods. In Figure 4, a comparison of the same was shown. In comparison to its competitors, the MapReduce-based location recommendation method has the shortest execution time.

Table 1: The execution times in milliseconds for different location ranking methods

No. of users	Proposed system	CF with CNN	Naïve Bayes algorithm	TOP K RS
10	201	393	512	556
20	289	447	666	643
30	339	511	709	756
40	379	599	819	847
50	413	699	899	932
60	487	789	981	1021

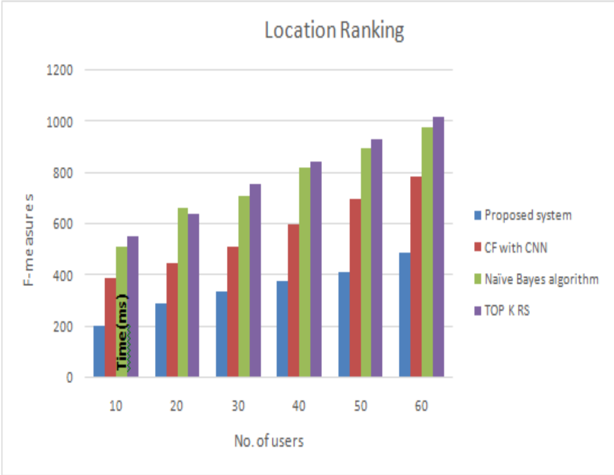

Fig 4: Execution times of several location ranking methods

The methods' Accuracy values are presented in Table 2, and a comparison of those values is displayed in Figure 5. The location ranking method built on FSO-DANN has better accuracy values.

Table 2: Accuracy for different location ranking methods

No. of users	Proposed system	CF with CNN	Naïve Bayes algorithm	TOP K RS
10	0.791	0.776	0.78	0.754
20	0.821	0.801	0.79	0.779
30	0.851	0.843	0.81	0.824
40	0.873	0.867	0.845	0.834
50	0.888	0.878	0.867	0.856
60	0.899	0.89	0.867	0.869

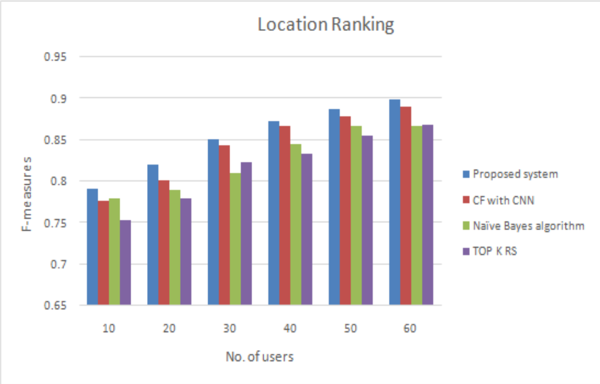

Fig 5: Compares the Accuracy for different location ranking methods

The FSO-DANN-based location recommendation algorithm is compared with the hybrid algorithm CF-based location recommendation algorithm and content-based location recommendation algorithm. Table 3 lists the execution times of the various location recommendation algorithms. The comparison is shown in Figure 6.

Table 3: The execution times in milliseconds for different location RS methods

No. of users	Proposed system	CF with CNN	Naïve Bayes algorithm	TOP K RS
10	202	393	499	615
20	295	467	666	643
30	340	511	709	756
40	380	599	819	847
50	414	699	899	932
60	488	789	981	1021

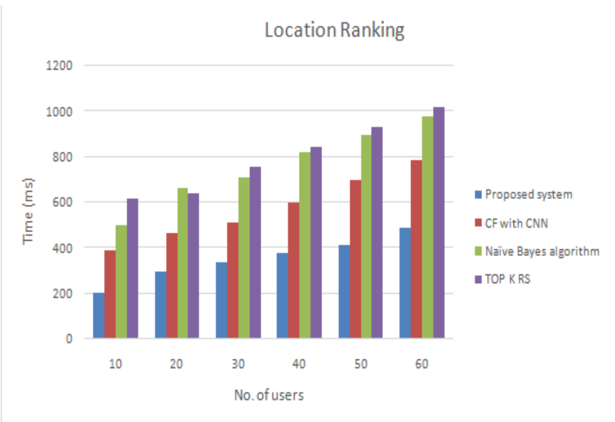

Fig 6: Time evaluation of the different location

The methods' F-measure values are presented in Table 4, and the relationship is shown in Figure 7. The location recommendation technique based on FSO-DANN has the highest F-measure values of all the methods.

Table 4: F-measures for several location recommendation methods

No. of users	Proposed system	CF with CNN	Naïve Bayes algorithm	TOP K RS
10	0.818	0.722	0.71	0.65
20	0.821	0.756	0.723	0.711
30	0.839	0.777	0.768	0.697
40	0.849	0.789	0.69	0.714
50	0.89	0.819	0.79	0.721
60	0.899	0.823	0.799	0.729

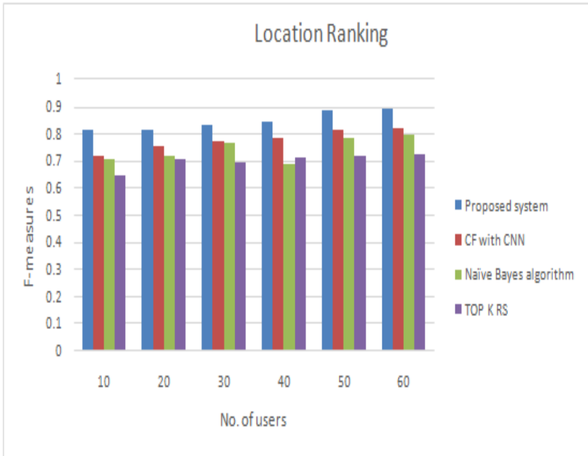

Fig 7: F-measures for several location recommendation methods

The FSO-DANN-based location recommendation method is unique from the ones already in use in that it possesses the following features.

• It predicts locations using a conditional neural network.

• It employs a Map Reduce-based method for rating locations.

• Local customer tips are taken into account.

5. Conclusion

The algorithm takes a methodical method to site recommendation. Candidate locations are chosen in such a way as to only include the most suitable areas for the customer. The location ranking method improves the ranking approach by taking into account customer feedback from prior visits to the areas under consideration. In terms of execution speed and accuracy of forecasting, the new FSO-DANN-based location RS performs better than the ones currently in operation.The method forecasts suitable destinations for the user based on location rating as opposed to forecasting and offering the top-ranked location to them.

References

[1] Gao, Q., Wang, W., Huang, L., Yang, X., Li, T., & Fujita, H. (2023). Dual-grained human mobility learning for location-aware trip recommendation with spatial–temporal graph knowledge fusion. *Information Fusion*, *92*, 46-63.

[2] Canturk, D., Karagoz, P., Kim, S. W., & Toroslu, I. H. (2023). Trust-aware location recommendation in location-based social networks: A graph-based approach. *Expert Systems with Applications*, *213*, 119048.

[3] Bhaskaran, S., & Marappan, R. (2023). Enhanced personalized recommendation system for machine learning public datasets: generalized modeling, simulation, significant results and analysis. *International Journal of Information Technology*, *15*(3), 1583-1595.

[4] Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., & He, X. (2023). Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, *41*(3), 1-39.

[5] Pramod, D. (2023). Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda. *Data Technologies and Applications*, *57*(1), 32-55.

[6] Vu, S. L., & Le, Q. H. (2023). A deep learning based approach for context-aware multi-criteria recommender systems. *Computer Systems Science and Engineering*, *44*(1), 471-483.

[7] Sivamayil, K., Rajasekar, E., Aljafari, B., Nikolovski, S., Vairavasundaram, S., & Vairavasundaram, I. (2023). A systematic study on reinforcement learning based applications. *Energies*, *16*(3), 1512.

[8] Fang, J., Meng, X., & Qi, X. (2023). A top-k POI recommendation approach based on LBSN and multi-graph fusion. *Neurocomputing*, *518*, 219-230.

[9] Liang, W., Li, Y., Xu, J., Qin, Z., Zhang, D., & Li, K. C. (2023). Qos prediction and adversarial attack protection for distributed services under dlaas. *IEEE Transactions on Computers*.

[10] Lin, W., Leng, H., Dou, R., Qi, L., Pan, Z., & Rahman, M. A. (2023). A federated collaborative recommendation model for privacy-preserving distributed recommender applications based on microservice framework. Journal of Parallel and Distributed Computing, 174, 70-80.

[11] Yang, L., Wang, S., Tao, Y., Sun, J., Liu, X., Yu, P. S., & Wang, T. (2023, February). DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (pp. 661-669).

[12] Wei, X., Liu, Y., Sun, J., Jiang, Y., Tang, Q., & Yuan, K. (2023). Dual subgraph-based graph neural network for friendship prediction in location-based social networks. ACM Transactions on Knowledge Discovery from Data, 17(3), 1-28.

[13] Pham, P., Nguyen, L. T., Nguyen, N. T., Kozma, R., & Vo, B. (2023). A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. Information Sciences, 620, 105-124.

[14] Dadoun, A., Defoin-Plate, M., Fiig, T., Landra, C., & Troncy, R. (2023). How recommender systems can transform airline offer construction and retailing. In Artificial Intelligence and Machine Learning in the Travel Industry: Simplifying Complex Decision Making (pp. 93-107). Cham: Springer Nature Switzerland.

[15] Casillo, M., Colace, F., Conte, D., Lombardi, M., Santaniello, D., & Valentino, C. (2023). Context-aware recommender systems and cultural heritage: a survey. Journal of Ambient Intelligence and Humanized Computing, 14(4), 3109-3127.

[16] Praneesh, M., and R. Annamalai Saravanan. "Deep Stack Neural Networks Based Learning Model for Fault Detection and Classification in Sensor Data." *Deep Learning and Edge Computing Solutions for High Performance Computing* (2021): 101-110.

[17] McKitrick, M. K., Schuurman, N., & Crooks, V. A. (2023). Collecting, analyzing, and visualizing location-based social media data: review of methods in GIS-social media analysis. GeoJournal, 88(1), 1035-1057.

[18] Zhou, Z., Liu, Y., Ding, J., Jin, D., & Li, Y. (2023, April). Hierarchical knowledge graph learning enabled socioeconomic indicator prediction in location-based social network. In Proceedings of the ACM Web Conference 2023 (pp. 122-132).

[19] A. Kumar, R. S. Umurzoqovich, N. D. Duong, P. Kanani, A. Kuppusamy, M. Praneesh, and M. N. Hieu, ‘‘An intrusion identification and prevention for cloud computing: From the perspective of deep learning,’’ Optik, vol. 270, Nov. 2022, Art. no. 170044.