SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Robust Learning Architecture For Incomplete Data Environments Through Positive-Unlabeled Recommendation Learning

Dr. A. Vijayalakshmi[1], Milind Patil[2], Shree Jayaram K[3], Ms. Arpita A. Prajapati[4], Ashish Kumar Sharma[5], Vijayalakshmi Pasupathy[6], Seethaladevi S[7], Anjali Bhardwaj[8]

[1] Assistant Professor, Department of English, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India. Email: vijayalakshmi_english@cbit.ac.in
ORCID: 0000-0002-3422-6409
[2] Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: milind.patil@vit.edu
[3] Innovation & Incubation Centre, Department of Research, Meenakshi Academy of Higher Education and Research, India. Email: shreejayaram@maher.ac.in
[4] Lecturer, Faculty of Engineering, Gokul Global University, Sidhpur, Gujarat, India.
Email: aaprajapati.ce.hcet@gokuluniversity.ac.in ORCID: 0009-0003-8878-9762
[5] Assistant Professor, Department of Computer Application, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.asishkumar@dbuu.ac.in ORCID: 0009-0008-1487-8663
[6] Associate Professor, Department of Computer Science and Engineering, Panimalar Engineering College, India. Email: pvijayalakshmi@panimalar.ac.in
[7] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: seethaladevi@maher.ac.in
[8] Assistant Professor, Department of School of Engineering & Technology, Noida International University, Greater Noida, Uttar Pradesh, India. Email: anjali.bhardwaj@niu.edu.in

Abstract
Incomplete and biased data environments present significant challenges in domains such as Computer Vision (CV), Natural Language Processing (NLP), and Recommender Systems (RS), where the absence of explicit negative labels reduces prediction reliability and decision-making accuracy. Existing approaches for handling implicit feedback primarily rely on pseudo-negative sampling, confidence screening, and preference inference methods. However, the Missing-Not-At-Random (MNAR) features of incomplete data and the impact of hidden confounding variables are not sufficiently resolved by these methods, leading to poor applicability and limited resilience in practical applications. An incomplete Interaction Dataset employed for experimental evaluation, 12,000 user-item interactions, and 49 attributes make up the interaction data. Min-Max normalization and missing interaction imputation techniques are used during data preprocessing to improve data quality and consistency. Feature extraction is performed using a Variational Autoencoder (VAE) to learn latent embeddings and interaction-based features for capturing hidden behavioral patterns. A robust Deer Hunting Optimized Light Gradient Boosting (DHO-LGB) model is proposed to address incomplete data environments effectively. Positive-Unlabeled (PU) with exposure imputation and doubly robust debiasing reduces bias effectively. DHO maximizes performance by finding optimal features and parameters. LGBM allows fast classification with great accuracy and scalability. DHO helps optimize LGBM by effectively choosing the best features and improving predictions. The model is implemented using Python and TensorFlow. The experimental findings show that the DHO-LGB model developed performs better with an accuracy of 0.95, recall of 0.94, ILD of 0.92, and coverage of 0.96 with superior robustness, reliability, and suggestion efficacy in comparison to current cutting-edge methods.

Keywords: Incomplete Data, Positive-Unlabeled Learning, Debiasing, Recommender Systems, Robust Learning Architecture.

1. Introduction

As contemporary systems, large volumes of data from various fields have missing values, making the data unreliable. Robust learning architectures tackle the problem of missing values through the use of sophisticated imputation techniques and learning mechanisms [1]. As regards data mining and analytics, original data is sometimes incomplete since it contains some missing values that may affect the overall performance of the model used. There are, however, robust learning models that have been designed specifically to cater for incomplete data [2]. Missing values are common in practical applications because of issues related to the integration of data, sensor failure, or incomplete answers. Robust learning models are designed to solve such problems by ensuring that predictions are accurate and models are valid in the presence of partial data [3]. It is a frequent occurrence in recommendation systems and similar fields that there are missing and biased entries, making the prediction unreliable. Robust learning methods are intended to be able to cope with such problems, thus enhancing their estimation, calibration, and learning capabilities [4]. Missing values occur in practice within tabular data sets because of data collection problems and system failures that could affect accuracy in prediction. Imputation algorithms should enable robust machine learning methods to handle missing values in their training data [5]. The incompleteness and imbalance of the dataset due to missing values will adversely affect the performance of semi-supervised learning. Self-training algorithms require completely balanced datasets and are highly affected by outliers; thus, they provides biased results [6]. Active learning algorithms generally use complete databases; however, the actual databases contain incomplete records. The traditional methods depend on prior imputation, which may lead to errors; hence, such errors can adversely affect query selection and negatively influence model accuracy for ordinal classification [7]. Today's recommendation engines running under incomplete data conditions face challenges associated with the lack of interaction information, exposure bias, and hidden confounders that limit the accuracy of predictions and recommendations generated. Standard techniques such as collaborative filtering (CF), negative sampling, confidence scoring, adaptive sampling, and preference inference have not adequately handled MNAR features, and they often classify possible positive cases as negative, limiting their robustness, generalizability, and overall recommendation capability [8].

Research Aim: To design and implement a PU learning-based recommendation model capable of addressing MNAR interaction and implicit feedback in biased and partial data scenarios. To increase the effectiveness of the recommendations through the adoption of doubly robust exposure and confounding debiasing techniques and exposure state imputation so that a distinction can be made between missing and unseen interactions. Moreover, the proposed DHO-LGB algorithm uses the combination of DHO and LGBM methods for effective feature selection.

Research Organization: The history, motivation, issue definition, and significance of resilient recommendation systems in biased and incomplete contexts are presented in Section 1. Section 2 discusses related work on optimization techniques, debiasing algorithms, MNAR data processing, recommender systems, PU learning, and some identified research gaps. The methodology design, which includes doubly robust debiasing, exposure imputation, data preparation, and DHO-LGB, is covered in Section 3. Section 4 covers experiments, dataset descriptions, assessment metrics, and implementation specifics. Experiments, findings, conclusions, ablations, and recommendations for future research will all be included in Section 5.

2. Related Works

A summary of the comparison of present research methodologies regarding the aims, methods used, findings, and limitations is shown in Table 1 below. Dealing with insufficient data, bias elimination, and the generation of reliable recommendation learning algorithms have been identified as key problems faced.

Table 1: Benchmark Comparison of Recommendation Learning Methods

Ref	Objective	Method	Result	Limitations
[9]	Evaluate imputation methods for missing vital signs	Linear, spline, and statistical interpolation techniques are used	Linear interpolation achieved the lowest median imputation error	Imputation introduced signal bias and risk of misclassification

[10]	Minimize recommendation bias under incomplete data	Approximate upper-bound debiasing with randomized learning	Improved recommendation robustness and generalization performance	Limited handling of hidden, unlabeled uncertainty
[11]	Impute missing data using reinforcement learning (RL)	RL-based policy learning for value imputation	Preserves variance and improves performance	Works mainly in a univariate column-wise setting
[12]	Address incomplete feedback and mitigate MNAR bias through robust PU learning.	Two-phase debiasing process using exposure imputation and robust deconfounding	Achieved superior recommendation accuracy across multiple benchmark real-world datasets	Performance depends on accurate exposure estimation and imputation quality
[13]	Mitigate exposure bias while incorporating user satisfaction signals	Causal inference system modeling exposure satisfaction and matching	Improved recommendation performance across four benchmark datasets significantly	Limited effectiveness under inaccurate causal assumptions and modeling
[14]	Predict MNAR ratings without costly explicit MAR feedback	Lightweight iterative probabilistic matrix factorization using user preferences	outperformed cutting-edge techniques in MNAR feedback prediction tasks	Assumes user preferences adequately represent missingness propensities consistently
[15]	Learn unbiased recommendations from implicit feedback with ranking	Unbiased pairwise learning method reducing variance without bias	Superior performance validated through experiments and online testing	Focuses mainly on variance reduction, limited bias analysis
[16]	Improve robust recommendations under incomplete feedback	Diversity-aware weighted positive-unlabeled learning model	Enhanced diversity with maintained recommendation accuracy	Limited handling of missing interaction bias

2.1 Research Gap: However, the current approaches suffer from various limitations, such as interaction bias, exposure estimation dependency, and hidden and unknown uncertainties. The techniques that prove this limitation are Robust PU Learning with Exposure Imputation [12], Randomized Debiasing Learning [10], and Diversity-Aware PU Learning [16]. Despite this limitation, DHO-LGB tries to address such problems through the use of PU learning, debiasing, and optimization-based feature learning techniques.

3. Methodology

Figure 1 depicts the whole process of recommending in the DHO-LGB algorithm in interactionally incomplete environments. Initially the raw data in sparse form undergoes minmax normalization and missing interaction data replacement. VAE to extract the features from the input data. After that, DHO with selected parameters and features and perform prediction using the LGB model. The proposed method results in greater recommendation accuracy and robustness in incomplete interaction settings.

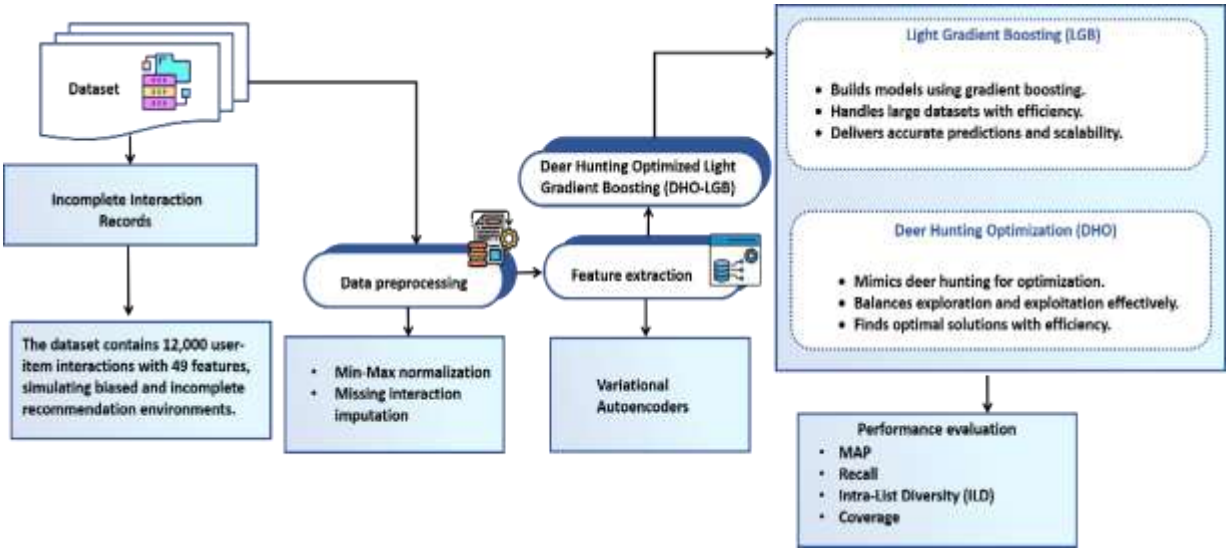


Figure 1: DHO-LGB Architecture for Robust Recommendation Systems

3.1 Data Collection

The number of Interaction data consists of 12,000 user-item interactions along with 49 features of the Incomplete Interaction Data Set, and is designed to mimic the biased and incomplete recommendations environment. The data set comprises information on users' profiles, items' features, their interaction history, exposure, non-interaction conditions, context, interaction value, and interaction category. An interaction data entry refers to one interaction for a particular user-item pair. The impact of the interaction is explained through the target feature named "Interaction Category." Representative samples were selected and split with an 80%-20% time-aware train-test split to create a recommendation system.

Kaggle: https://www.kaggle.com/datasets/colabsss/incomplete-interaction-records

3.2 Data Preprocessing

Managing missing data, eliminating inconsistencies, and normalizing the data are all part of the data preparation procedure and organizing it for use in machine learning algorithms. Preprocessing of the interaction data uses the Min-Max normalization method to ensure consistency and reliability of the recommendation learning.

- **Min-Max normalization for Scales features for efficient learning**

The data transformation approach known as min-max normalization keeps intact the original data distribution while converting user-item interaction properties into a normalized form. Through the process of conversion of raw interaction properties, behavior scores, and other context properties to their normalized forms, the variance among different properties is reduced, making the learning procedure effective. The normalization process is represented as:

$$MMN, w'_{j,m} = \frac{w_{j,m} - min(w_j)}{max(w_j) - min(w_j)} (new_{Max} - new_{Min}) + new_{Min} \quad (1)$$

In Equation (1), MMN is an abbreviation for Min-Max Normalization, which is a method of feature scaling. $w'_{j,m}$ is the normalized form of feature m of instance j, and $w_{j,m}$ is the actual value of the feature. $min(w_j)$ stands for the minimum value of feature j, and $max(w_j)$ stands for its maximum value. new_{Min} is the lower bound of the normalized scale, and new_{Max} is the upper bound of the normalized scale. The expression $max(w_j) - min(w_j)$ is called the range of the original data.

- **Missing interaction imputation for Fills missing interactions for learning.**

Imputation of missing interactions is one of the methods employed to determine and substitute for the missing interactions between users and items in recommendation systems' data sets. The method employed in research for handling situations where missing interactions could be actual negative samples and missing values. Elimination of missing data, reduction of bias, and betterment of features applied in learning processes are made possible by using this method of data preprocessing.

3.3 Feature Extraction using Variational Autoencoders (VAE) for Extracts latent features from data.

The process of feature extraction through VAE is considered one of the generative methods for representation learning, where complex datasets are encoded using compact and informative latent spaces using trained neural networks. The encoding process creates the mapping from input to latent space that consists of the variance and mean parameters. The decoder learns the capability to decode the sampled latent variables into input data.

$$Y \sim o(Y) = N(Y; 0, J) \qquad W|Y \sim o(W|Y, \theta) = N(W; h(Y; \theta), u1) \quad (2)$$

In Equation (2), where Y stands for latent user-item representation, $o(Y)$ is the distribution of Y. $N(Y; 0, J)$ is the normally distributed, with 0 being its mean and an identity covariance matrix. The identity covariance matrix J indicates the uncorrelated features of latent variable Y. W stands for observed interaction, while $o(W|Y, \theta)$ stands for the probability of interaction W when Y is known. In this case, θ is a set of parameters used in the decoder neural network, and $h(Y; \theta)$ stands for the mapping function from latent to data space. $N(W; h(Y; \theta), u1)$ means that the interaction is normally distributed with mean $h(Y; \theta)$ and variance $u1$, where $u1$ is observation noise.

$$log\ o(W|\theta) \geq \langle log\ o(W|Y, \theta)\rangle_{r(Y|W,\eta)} + \langle log\ o(Y)\rangle_{r(Y|W,\eta)} + G[r(Y|W, \eta)] \equiv \mathcal{B}(\theta, \eta) \quad (3)$$

Here, $log\ o(W|\theta)$ refers to the log-likelihood of observed interaction data W where $r(Y|W, \eta)$ refers to the variational posterior, which is used to approximate the latent distribution, where η refers to encoder parameters. $\langle log\ o(W|Y, \theta)\rangle_{r(Y|W,\eta)}$ refers to the expectation under the variational distribution. $log\ o(W|Y, \theta)$ is the reconstruction likelihood while $\langle log\ o(Y)\rangle_{r(Y|W,\eta)}$ is the prior distribution of latent variables. $G[r(Y|W, \eta)]$ refers to the entropy of the variational distribution. In general, $\mathcal{B}(\theta, \eta)$ refers to the Evidence Lower Bound (ELBO) objective, which is presented in Equation (3).

$$\mathcal{KL}(\mathcal{Q}|\mathcal{P}) = \langle log\ (\mathcal{Q}/\mathcal{P})\rangle_{\mathcal{Q}} \quad \mathcal{Q} = \pi(W)r(Y|W, \eta) \quad \mathcal{P} = o(W|Y, \theta)o(Y) \quad (4)$$

In Equation (4), $\mathcal{KL}(\mathcal{Q}|\mathcal{P})$ is the Kullback-Leibler divergence measure used to compare two distributions. Here, Q refers to the approximate joint distribution, and P refers to the true generative joint distribution of the data and the latent variables. W is the observed user-item interaction dataset, while Y is the latent representation of user-item data. $\pi(W)$ refers to the data distribution of observed interactions, whereas $r(Y|W, \eta)$ is the variational posterior approximation performed through the encoder, with η being the encoder neural network parameters. Finally, $o(W|Y, \theta)$ is the observed data likelihood with respect to latent variables and decoder parameters θ, while $o(Y)$ is the prior distribution of latent variables.

3.4 Robust Deer Hunting Optimized Light Gradient Boosting (DHO-LGB)

DHO-LGB is a mix of learning methods that enhances the effectiveness of the recommendation system when handling sparsity and incompleteness of the dataset by combining the benefits of global searching with gradient-based prediction. In DHO, the high-dimensional feature space is efficiently explored to find the ideal set of features to improve model representativeness. The LGB model is then trained using those chosen features, turning weak learners into strong ones to produce extremely accurate predictive models. Higher degrees of learning stability, generalization, and overfitting resistance are thereby attained by DHO-LGB.

- **Light Gradient Boosting (LGB) for Performs accurate and efficient prediction**

Recommendations on partially labeled and PU interaction data are generated with the help of LightGBM. To minimize losses, the algorithm learns through the growth of decision trees. To minimize computation time and maximize useful interactions, the approach of gradient boosting, Gradient-based One-Side Sampling (GOSS), and Exclusive Feature Bundling (EFB) is applied.

$$\hat{e} = \underset{e}{argmin}\, F_{TR,tr^q}\,[K(tr^q, e(TR))] \quad (5)$$

Where $\hat{e}$ is the feature transformation or feature subset selected through the optimization process. Here, $\underset{e}{argmin}$ refers to the method used for selecting the feature subset, which results in the minimum value of the objective function. In this case, e refers to the feature selection/feature transformation function, and TR refers to the training set. tr^q is referred to as the test/query instance. $e(TR)$ represents the transformed training set features after feature selection. $[K]$ provides the error or cost value based on the deviation from the query data and transformed training data in Equation (5).

$$E_l(TR) = E_{l-1}(TR) + g_l(TR)\left[\underset{e}{argmin} \sum_{j=1}^{n} K(tr_j^q, E_{l-1}(TR_j) + \gamma g_l(TR_j))\right] \quad (6)$$

In Equation (6), $E_l(TR)$ is the new representation of the model at iteration l, while $E_{l-1}(TR)$ is the old representation of the model at iteration $l-1$. $g_l(TR)$ is the weak classifier or updating rule applied to step l. tr_j^q is the query or test pattern for the sample j, whereas TR_j stands for the training pattern. $K(\cdot)$ is the loss function which determines the loss made by the learner in its predictions, and $\sum_{j=1}^{n}$ calculates the overall loss over the

training dataset. $\underset{e}{argmin}$ selects the best possible update or feature transformation. γ is the learning rate, and l and n denote the iterations and the number of training samples, respectively.

$$U_i(o) = \frac{1}{m}\left[\frac{\left(\sum_{tr_j \in B_v} h_j + \frac{1-b}{a}\sum_{tr_j \in A_v} h_j\right)^2}{m_v^i(o)} + \frac{\left(\sum_{tr_j \in B_u} h_j + \frac{1-b}{a}\sum_{tr_j \in A_u} h_j\right)^2}{m_u^i(o)}\right] \quad (7)$$

In Equation (7), $U_i(o)$ is an objective function of instance i, in which o refers to the state of the optimization problem or parameters, while m indicates the total normalization constant required for sample size adjustment. $\sum_{tr_j \in B_v} h_j + \frac{1-b}{a}\sum_{tr_j \in A_v} h_j$ cluster, where B_v refers to the reliable samples while A_v indicates the ambiguous samples in relation to the positive and unlabeled data. The h_j is a function of the contribution or score of samples j, a is the scaling constant related to the positive cluster, while b denotes the bias constant. $m_v^i(o)$ indicates the normalization or variance constant of the positive cluster. The $\sum_{tr_j \in B_u} h_j + \frac{1-b}{a}\sum_{tr_j \in A_u} h_j$ denotes the unlabeled cluster in which B_u and A_u indicate the reliable and ambiguous unlabeled samples, respectively, in $m_u^i(o)$.

- **Deer Hunting Optimization (DHO) for Optimizes features and model parameters**

DHO modeling is employed for simulating cooperative behavior in the hunting of deer to identify appropriate parameter settings. The solutions are modified based on leaders and successors to attain an equilibrium between exploration and exploitation processes. The optimization process employs neighborhood search movement and wind angles for updating the position of solutions.

$$e(w) = max\ (accuracy) \quad (8)$$

Equation (8) refers to $e(w)$, which is the representation of the fitness function based on the weight vector w. Vector w represents the parameters of the model. The term " $accuracy$ " stands for the performance metric of the model, such as its accuracy of classification or recommendation. In this case, the symbol $max()$ is an operation that implies that the optimization is done in order to maximize the accuracy.

$$W_{i+1} = W_k - Z.o: |K \times W_k - W_i| \quad (9)$$

Equation (10), W_{i+1} indicates the new location or solution in iteration $i+1$, whereas W_k signifies the current best solution or leader solution, and W_i indicates the current solution in iteration i. Z is a random scaling or control factor, whereas o is an adjustment factor that indicates the step size or helps in controlling the step size during the updating procedure. Here, K indicates a directional or weighting factor matrix/vector. $|K \times W_k - W_i|$indicates the distance between the position influenced by the leader and the current solution.

Algorithm 1: DHO-VAE-LGB for Robust Recommendation and Prediction Model

Input:

- User–Item interaction dataset D
- Training epochs E
- Population size for Deer Hunting Optimization P
- Latent dimension d
- LightGBM parameters search space Ω

Output:

- Predicted interaction scores $\hat{Y}$
- Optimized recommendation model M

Begin

1. Data Preprocessing
2. Load dataset D
3. Feature Extraction using Variational Autoencoder (VAE)
4. $w'_{j,m} = \frac{w_{j,m} - min(w_j)}{max(w_j) - min(w_j)}(new_{max} - new_{min}) + new_{min}$
5. Encoder:
6. $z = \mu(x) + \sigma(x) \cdot \epsilon, \epsilon \sim \mathcal{N}(0, I)$
7. Prior: $p(Y) = \mathcal{N}(0, I)$
8. Compute KL divergence: $L_{KL} = KL(q(Y \mid X) \parallel p(Y))$
    a. DHO for Feature & Hyperparameter Selection: W_i: $e(W) = max$(AUC / Accuracy)
9. Train LightGBM using selected features:
10. $\hat{Y} = LGB(F^*; W^*)$

11. Model Optimization Loop
    a. For each iteration l: $E_l = E_{l-1} + g_l(F^*)$
12. Final Prediction
    a. Train final optimized model Mon full dataset $F^*\hat{Y} = M(F^*)$
13. Return
14. Predicted interaction scores $\hat{Y}$
15. Optimized trained model M
16. End

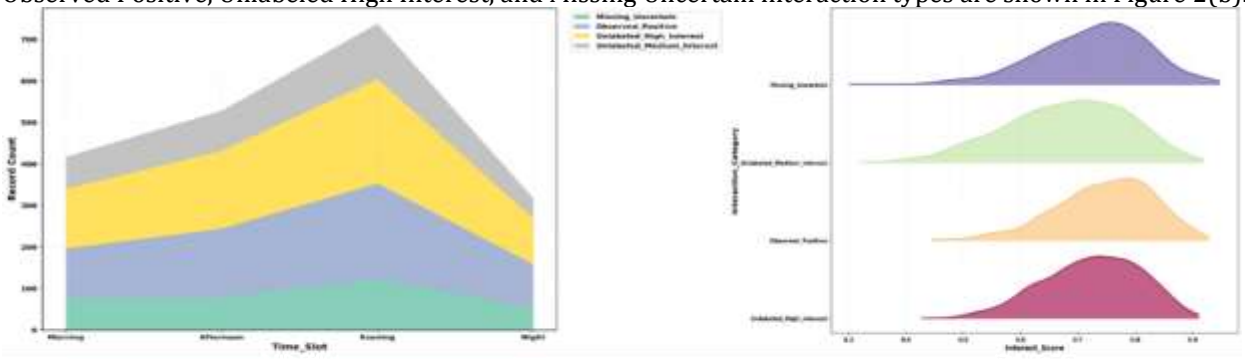
The DHO-LGB, data pre-processing, learning representation, training, and prediction are embedded into a single recommendation process. First, the interaction data undergoes normalization based on the Min-Max scaler, which is followed by the extraction of latent features via VAE. Finally, DHOs are used for the selection of optimal feature attributes and Light GBM hyperparameters, with the obtained feature set utilized for building the final predictive LGBM. Algorithm 1 provides an overview of the methodology.

4. Result and Discussion

The experiment was carried out using Python since the TensorFlow model was used with the aid of the Windows environment. It is possible to claim that even if there is no positive unlabeled information, learning and suggestion stability have increased. The use of different strategies improved recommendation performance.

Mean Average Precision (MAP): Given the incomplete data regarding interactions between users and items, it calculates the efficiency of ranking relevant items through the recommendation system. **Recall:** This metric evaluates the recommendation system's ability to identify the relevant objects from unlabeled data in terms of the positive-unlabeled setting. **Intra-List Diversity (ILD):** It indicates the diversity of the recommended items, proving the ability of the recommendation system to recommend various items. **Coverage:** It indicates the proportion of effectively recommended products, thus proving the system's ability to increase the users' visibility.

Data feature exploration: The Pattern Distribution of User-Item Interactions in the Absence of Recommendations is presented in Figure 2. Patterns of interaction types along different timeslots are presented in Figure 2(a). The Unlabeled High Interest interaction type shows an increase from 0 to 250 at night, while Observed Positive increases from 195 during the day to 350 at night. There are over 150 records, which is the highest number of records among all the other interaction categories. Interest score distributions for Observed Positive, Unlabeled High Interest, and Missing Uncertain interaction types are shown in Figure 2(b).



(a) (b)

Figure 2: Interaction Pattern Analysis of (a) Time-Based Interaction (b) Interest Score

Performance Evaluation using the Movie Lens dataset: The proposed DHO-LGB was trained on the Movie Lens dataset [16] with previous method includes Collaborative Filtering (CF), Content-Based Filtering (CBF), Diversity-Augmented Models, Neural Collaborative Filtering (NCF), and Multidimensional Recommender Model [16], which includes basic item metadata like genres and textual descriptions, along with explicit ratings on a range of 1 to 5. The findings obtained through experimentation reveal that the proposed DHO-LGB works better than any other existing methods in terms of recommendation performance measures. These include

MAP = 0.78, Recall = 0.72, ILD = 0.64, and Coverage = 0.84, showing a highly recommendable ability and increased diversity in recommendations. The performance results are illustrated in Table 2 and Figure 3.

Table 2: Quantitative Evaluation on MovieLens Dataset

Model	MAP	Recall	ILD	Coverage
CF[16]	0.75	0.68	0.30	0.45
CBF [16]	0.72	0.65	0.32	0.50
Diversity-Augmented Models [16]	0.70	0.63	0.50	0.70
NCF [16]	0.74	0.67	0.34	0.52
Multidimensional Recommender Model [16]	0.73	0.66	0.60	0.80
DHO-LGB [proposed]	0.78	0.72	0.64	0.84

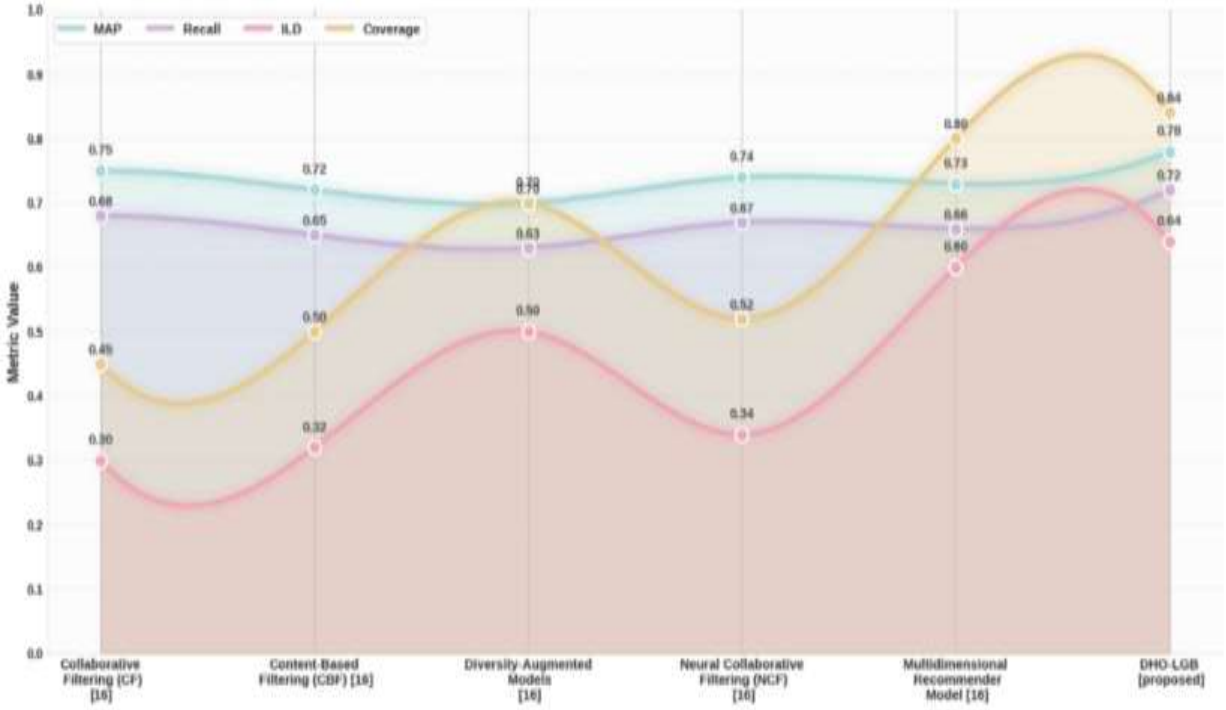


Figure 3: Glassmorphism Recommender System Performance Comparison

Performance Evaluation using the Incomplete Interaction Records: Both the proposed and existing methods were trained on the Incomplete Interaction Records. It is clear from Table 3 and Figure 4 that DHO-LGB is the best-performing model among all, with MAP = 0.95, Recall = 0.94, ILD = 0.92, and Coverage = 0.96. This result indicates that the proposed model works better in terms of coverage, diversity, generalization, and suggestion.

Table 3: Comparative Analysis of MAP, Recall, ILD, and Coverage

Model	MAP	Recall	ILD	Coverage
CF	0.83	0.82	0.74	0.78
CBF	0.84	0.83	0.76	0.80
Diversity-Augmented Models	0.86	0.85	0.84	0.87
NCF	0.88	0.87	0.79	0.84
Multidimensional Recommender Model	0.91	0.90	0.89	0.92
DHO-LGB [Proposed]	0.95	0.94	0.92	0.96

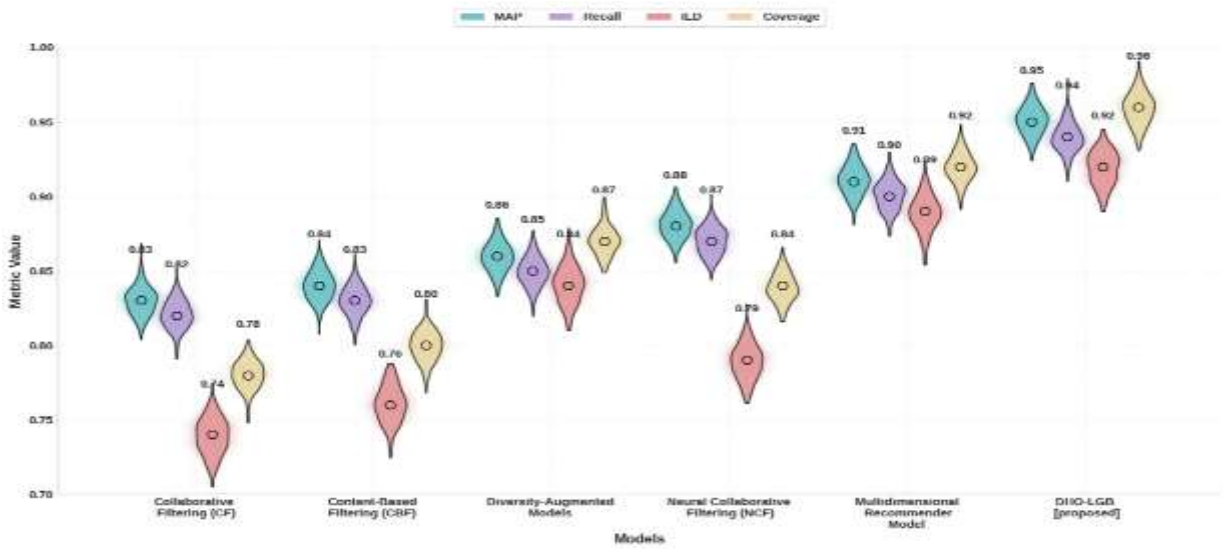


Figure 4: Analysis of Comparative Recommendation Performance Models

Sparsity, hidden interactions, and user-item bias interactions are some common challenges associated with the existing techniques such as CF, CBF, Diversity-Augmented Models, NCF, and Multidimensional Recommender Model [16] for making predictions less effective. However, the problems can be overcome using the proposed DHO-LGB architecture that employs the VAE method to extract features and predict interactions. DHO facilitates improved prediction performance through selection of proper features and parameter setting, the latter allowing LGB to play its part in accurate prediction.

5. Conclusion

The proposed DHO-LGBM method leverages PU learning, exposure-based imputation, and doubly robust debiasing to address missing feedback and MNAR bias in incomplete datasets. VAE-based latent feature extraction and DHO-driven feature selection and hyperparameter tuning enhance recommendation performance. The model achieves 0.95 accuracy, 0.94 recall, 0.92 ILD, and 0.96 coverage, outperforming existing methods. The proposed DHO-LGBM is prone to higher complexities while optimizing and requires exact exposure as well. Sparsity and dynamics in recommendations may affect the performance of the model. Future research could explore the use of graph neural networks (GNNs) and self-supervised learning algorithms that can help enhance representation learning and computation efficiencies for the recommendation system.

Reference

1. Khan, H., Wang, X. and Liu, H., 2022. Handling missing data through deep convolutional neural network. Information Sciences, 595, pp.278-293. https://doi.org/10.1016/j.ins.2022.02.051
2. Lin, W.C., Tsai, C.F. and Zhong, J.R., 2022. Deep learning for missing value imputation of continuous data and the effect of data discretization. Knowledge-Based Systems, 239, p.108079. https://doi.org/10.1016/j.knosys.2021.108079
3. Josse, J., Chen, J.M., Prost, N., Varoquaux, G. and Scornet, E., 2024. On the consistency of supervised learning with missing values. Statistical Papers, 65(9), pp.5447-5479. https://doi.org/10.1007/s00362-024-01550-4
4. Gong, S. and Ma, C., 2025. Gradient-Based Multiple Robust Learning Calibration on Data Missing-Not-at-Random via Bi-Level Optimization. Entropy, 27(2), p.196. https://doi.org/10.3390/e27020196
5. Meng, X., Li, X., Zhao, L., Shen, Y., Sun, H., Cong, X., Cao, X., Bi, Y., Liu, J. and Bian, Y., 2025. Robust representation of domain incomplete tabular data via cross-modal debiasing and prototype-fused reconstruction. Journal of King Saud University Computer and Information Sciences, 37(7), p.209. https://doi.org/10.1007/s44443-025-00225-w

6. Chen, M., Dou, J., Fan, Y. and Song, Y., 2023. Robust semi-supervised classification for imbalanced and incomplete data. Journal of Intelligent & Fuzzy Systems, 45(2), pp.2781-2797. https://doi.org/10.3233/JIFS-230658
7. He, D., 2023. Active learning for ordinal classification on incomplete data. Intelligent Data Analysis, 27(3), pp.613-634. https://doi.org/10.3233/IDA-226664
8. Yang, W.T., Chen, C.T., Sang, C.Y. and Huang, S.H., 2023. Reinforced PU-learning with hybrid negative sampling strategies for recommendation. ACM Transactions on Intelligent Systems and Technology, 14(3), pp.1-25. https://doi.org/10.1145/3582562
9. van Rossum, M.C., da Silva, P.M.A., Wang, Y., Kouwenhoven, E.A. and Hermens, H.J., 2023. Missing data imputation techniques for wireless continuous vital signs monitoring. Journal of clinical monitoring and computing, 37(5), pp.1387-1400. https://doi.org/10.1007/s10877-023-00975-w
10. Liu, D., Cheng, P., Lin, Z., Luo, J., Dong, Z., He, X., Pan, W. and Ming, Z., 2022. KDCRec: Knowledge distillation for counterfactual recommendation via uniform data. IEEE Transactions on Knowledge and Data Engineering, 35(8), pp.8143-8156. https://doi.org/10.1145/3582002
11. Awan, S.E., Bennamoun, M., Sohel, F., Sanfilippo, F. and Dwivedi, G., 2022. A reinforcement learning-based approach for imputing missing data. Neural Computing and Applications, 34(12), pp.9701-9716. https://doi.org/10.1007/s00521-022-06958-3
12. Wang, S., Xia, T. and Yang, L., 2025. Positive-Unlabeled Learning in Implicit Feedback from Data Missing-Not-At-Random Perspective. Entropy, 28(1), p.41. https://doi.org/10.3390/e28010041
13. Liao, J., Yang, M., Zhou, W., Zhang, H. and Wen, J., 2024. Modeling item exposure and user satisfaction for debiased recommendation with causal inference. Information Sciences, 676, p.120834. https://doi.org/10.1016/j.ins.2024.120834
14. Xu, Y., Zhuang, F., Wang, E., Li, C. and Wu, J., 2024. Learning without missing-at-random prior propensity-a generative approach for recommender systems. IEEE Transactions on Knowledge and Data Engineering, 37(2), pp.754-765. https://doi.org/10.1109/TKDE.2024.3490593
15. Lin, S., Zhou, S., Chen, J., Feng, Y., Shi, Q., Chen, C., Li, Y. and Wang, C., 2024. ReCRec: Reasoning the causes of implicit feedback for debiased recommendation. ACM Transactions on Information Systems, 42(6), pp.1-26. https://doi.org/10.1145/3539618.3592077
16. Bojic, L., Ilić, V. and Felfernig, A., 2026. Advancing diversity in recommender systems: a model for enhancing societal well-being. Journal of Intelligent Information Systems, 4 pp.1-35. https://doi.org/10.1007/s10844-026-01056-5