An Explainable Deep Learning Recommender System Integrating Trust and Tag Information

1st Malek Ghanem
University of Tunis-ISG, LARODEC
Tunis, Tunisia
malek.ghanem@isg.rnu.tn

2nd Wided Guezguez
Université de Tunis-ISG, LARODEC
Université de Tunis-ESSECT
Tunis, Tunisia
widedguezguez@gmail.com

3rd Raouia Ayachi
Université de Tunis-ISG, LARODEC
Université de Tunis-ESSECT
Tunis, Tunisia
raouia.ayachi@isg.rnu.tn

***Abstract*—With the rapid growth of online platforms, recommender systems play a crucial role in supporting decision-making in smart digital environments. However, many existing approaches suffer from data sparsity and lack transparency, which can reduce user trust and limit human–AI interaction. To address these challenges, we propose *TTHDL++*, an explainable hybrid deep learning recommender system that jointly integrates trust relationships, semantic tag information, and an Autoencoder-based feature extraction module. The proposed architecture combines multiple neural layers to capture complex interaction patterns while mitigating sparsity through compact latent representations. Beyond accuracy, TTHDL++ incorporates an explainability mechanism based on an *Item–Friend Interaction vector* that highlights the influence of trusted peers, enhancing transparency and user-centered decision-making. Experimental evaluation on two public benchmarks—*Last.fm* and *Delicious*—demonstrates that TTHDL++ consistently outperforms state-of-the-art baselines, achieving an F-measure of 0.875 at Top-5 and 0.855 at Top-10 on Last.fm, and 0.711 at Top-5 and 0.704 at Top-10 on Delicious, confirming both its recommendation quality and its interpretability in smart system environments.**

***Index Terms*—Explainable AI, Explainable Recommender Systems, Trust-Aware Recommendation, Tag-Based Recommendation**

I. Introduction

Recommender Systems (RS) have become essential components of modern online platforms, helping users navigate large volumes of information and discover relevant content [7], [20]. Among the most widely adopted paradigms, Collaborative Filtering (CF) infers user preferences from historical interaction patterns [16]. Nevertheless, CF techniques are frequently hindered by the well-known *data sparsity* problem, as users typically engage with only a small fraction of available items [2]. This limitation degrades the reliability and predictive accuracy of recommendation models [6], [8].

To alleviate sparsity and improve recommendation quality, recent research has explored the integration of supplementary contextual signals. In particular, *trust relationships* between users encode reliable social connections among individuals with compatible preferences [1], [6]. Likewise, *tags*—short textual labels voluntarily assigned by users to characterize items—provide semantic information that enriches item representations and supports personalized recommendations [31].

Beyond accuracy, transparency has recently emerged as a critical requirement for intelligent systems. Many deep learning-based recommender models operate as black boxes, making it difficult for users to understand the rationale behind suggested items. This opacity may erode user confidence and impede effective Human–Machine Interaction (HMI). As a result, *Explainable AI (XAI)* techniques have attracted increasing attention for their capacity to deliver interpretable recommendations and strengthen user trust [4], [15], [21].

Deep learning approaches offer powerful representation capabilities by modeling complex, non-linear relationships across heterogeneous data sources, making them particularly well-suited for recommender systems [6], [7]. Motivated by these advantages, we propose **TTHDL++**, an explainable hybrid recommender framework that integrates trust relationships and tag information within a unified deep learning architecture. The model operates in two main stages: (i) an Autoencoder-based feature extraction module designed to learn robust latent representations from sparse interaction data, and (ii) a neural recommendation module that generates ranked Top-N recommendations while simultaneously providing trust-aware explanations through an *Item–Friend Interaction vector*. This mechanism highlights the influence of trusted peers in the recommendation process, thereby improving transparency and user understanding.

The remainder of this paper is organized as follows. Section II reviews related work on explainable, trust-aware, and tag-based recommender systems, with an emphasis on deep learning approaches. Section III presents the TTHDL++ model in detail, covering the feature extraction, recommendation, and explainability phases, as well as human–machine interaction mechanisms. Section IV describes the experimental setup, datasets, and evaluation metrics, and provides a comparative analysis against state-of-the-art models. Finally, Section V concludes the paper and outlines directions for future research.

II. Related Work

Recommender systems have been extensively studied to assist users in discovering relevant items from large information spaces.

A. Explainable Recommender Systems

Explainable Recommender Systems (ERS) aim to provide transparent justifications for suggested items, enabling users to better understand the reasoning behind recommendations and increasing trust in intelligent systems. Existing approaches generally fall into three main categories:

1) **Post-hoc explanation methods** produce explanations after the recommendation has been generated. These approaches typically rely on interpretable rules or textual descriptions derived from user–item interactions.
2) **Feature-based explanations** identify specific attributes or characteristics that influence recommendations, such as shared item features, user preferences, or contextual signals.
3) **Attention-based models** employ neural attention mechanisms to quantify the contribution of individual components to a given recommendation.

Several techniques have been proposed to support explainability in recommender systems. Knowledge graph-based methods exploit semantic relationships between entities to generate human-readable justifications. Attention networks reveal the relative importance of different features during the recommendation process. Rule-based approaches produce explicit reasoning patterns that describe why a particular item is proposed. Despite these advances, seamlessly integrating explainability into high-performing deep learning models remains an open research challenge.

B. Deep Learning-Based Trust- and Tag-Aware Recommender Systems

User–item rating matrices are typically sparse, which limits the system's capacity to deliver accurate recommendations [7], [13]. Trust relationships capture correlations between users with compatible preferences, thereby helping to identify reliable recommendations [1], [6]. Tags—short descriptive words or phrases assigned by users—summarize item characteristics and reflect individual preferences, enabling richer profile construction and improved personalization [8].

Several deep learning-based approaches have been proposed in this space. TTDNN [8] jointly models trust links and tag information using a sparse Autoencoder to derive latent representations. To mitigate data sparsity, [9] employs a deep Marginalized Denoising Autoencoder, while DLIR [10] uses RBM-based Autoencoders for trust-aware initialization. TDAE [11] incorporates ratings alongside explicit trust signals, and TDAE++ extends this design by further accounting for implicit trust. A trust-enhanced deep model in [14] combines Autoencoder initialization with community-aware social influence. Beyond trust, [12] employs BERT with a hybrid CNN and LSTM for sentiment- and genre-based recommendations. Explainability is reinforced in [29] through multi-view comments and conversational updates. Finally, TRAL [30] introduces an attention-based tag-aware model that blends deep and linear components for robust user–item representations.

C. Deep Learning for Explainable Recommender Systems

Deep learning techniques have substantially advanced recommender systems by enabling the modeling of complex, nonlinear relationships between users and items. Neural architectures such as autoencoders, convolutional networks, and recurrent neural networks have been widely adopted to learn latent representations from sparse interaction data. Explanations in recommender systems are commonly assessed across three key dimensions:

1) **Effectiveness** measures how well explanations assist users in making informed choices by clarifying whether recommendations align with their preferences [26].
2) **Persuasiveness** reflects the system's capacity to encourage users to consider or adopt recommended items by influencing their decision process [27].
3) **Transparency** provides insight into how recommendations are generated, helping users understand the rationale behind suggestions [28].

Autoencoder-based models are particularly effective for feature extraction, as they can uncover hidden structures within user–item matrices and generate compact latent representations. Recent research has also explored neural explainability techniques, including attention-based explanations, feature importance analysis, and hybrid frameworks combining neural networks with interpretable components.

Hybrid recommender models integrating multiple information sources have attracted considerable interest in recent years. However, only limited work has examined the joint combination of trust relationships, tag-based semantic information, deep learning representations, and explainability mechanisms within a single unified framework. Addressing this gap, the proposed TTHDL++ model brings together these complementary elements to enhance both recommendation performance and interpretability.

D. Positioning of the Proposed Approach

As summarized in Table I, most existing approaches address only a subset of the key challenges in recommender systems. While TTDNN [8] integrates trust and tags, it offers no built-in explainability mechanism. DLIR [10] and TDAE [11] incorporate trust within a deep learning framework but disregard semantic tag information. TRAL [30] leverages attention-based learning over tags but does not explicitly model trust relationships between users. In contrast, **TTHDL++** is the only framework that simultaneously unifies trust, tags, Autoencoder-based feature extraction, and a dedicated explainability module, thereby offering a comprehensive and transparent solution for trust- and tag-aware recommendation.

TABLE I
COMPARISON OF RELATED APPROACHES ACROSS KEY DESIGN DIMENSIONS.

Model	**Trust**	**Tags**	**Deep**	**Expl.**	**FE**
TTDNN [8]	✓	✓	✓	×	✓
DLIR [10]	✓	×	✓	×	✓
TDAE [11]	✓	×	✓	×	✓
TRAL [30]	×	✓	✓	✓	×
TTHDL++	✓	✓	✓	✓	✓

Deep: Deep Learning architecture; *Expl.*: Built-in Explainability Mechanism; *FE*: Feature Extraction.

III. Proposed Method: TTHDL++ Explainable Trust–Tag Deep Recommender

This section presents the **TTHDL++** model, an explainable deep learning framework designed to improve recommendation performance while enhancing transparency. The model integrates trust relationships and tag information within a neural architecture to address data sparsity and provide interpretable recommendations.

A. Notation

- Let $\mathcal{U} = \{u_1, \ldots, u_n\}$ denote the set of users.
- Let $\mathcal{I} = \{i_1, \ldots, i_m\}$ denote the set of items.
- Let $\mathcal{T} = \{t_1, \ldots, t_q\}$ denote the set of tags.
- The observed rating of user u for item i is denoted $R_{u,i}$.
- $P_{u,i}$ represents the predicted rating generated by the model.
- Trust relationships are encoded in a trust matrix Tr, where $Tr_{u,v}$ denotes the level of trust that user u assigns to user v.

Tag information is encoded through user–tag and item–tag association matrices that capture semantic preferences and item characteristics.

B. Model Overview

The TTHDL++ framework integrates trust and tag information into a deep learning pipeline that produces both recommendation predictions and explanation outputs. As illustrated in Figure 1, the architecture comprises two main phases:

1) **Feature Extraction Phase:** An Autoencoder is applied to the User–Trust–Tag matrix in order to learn compact latent representations and mitigate sparsity effects.
2) **Recommendation and Explainability Phase:** The extracted latent features are processed through a neural recommendation module that generates Top-N recommendations while simultaneously producing interpretable explanations based on trusted peer interactions.

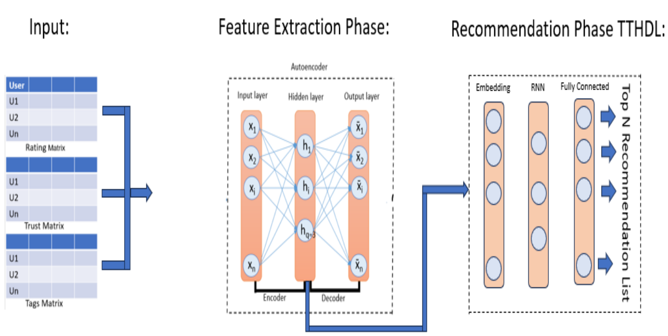

Fig. 1. Architecture of the proposed TTHDL++ framework.

C. Feature Extraction Phase

The Feature Extraction (FE) phase aims to obtain informative latent representations from sparse interaction data. A deep Autoencoder is employed to compress the User–Trust–Tag matrix into a lower-dimensional latent vector z.

Autoencoders are well suited for recommender systems due to three key properties: (i) *information compression*, whereby the encoder learns compact representations that retain the most relevant signals; (ii) *generalization capability*, allowing the model to capture patterns shared across users and items; and (iii) *non-linear representation learning*, enabling the discovery of hidden structures within sparse data.

The encoder consists of two fully connected layers with dimensions 512 and 256 (both using ReLU activation functions), followed by a bottleneck layer of dimension 128 that produces the latent representation $z \in \mathbb{R}^{128}$. The decoder mirrors this structure in reverse to reconstruct the original input. The resulting latent vector z is subsequently fed into the recommendation module.

D. Recommendation and Explainability Phase

The recommendation module transforms the latent representation z into prediction scores for candidate items through a sequence of neural layers:

- An **Embedding layer** (size 64) that projects categorical inputs into dense vector representations.
- A **Simple RNN layer** (hidden size 128) that captures sequential dependencies in user interactions and contextual tag information.
- A **Dropout layer** ($p = 0.3$) applied to prevent overfitting.
- A **Dense output layer** with a softmax activation function that produces the Top-N recommendation scores.

In addition to generating recommendation scores, the module produces an explanation output through an *Item–Friend Interaction vector*. This vector identifies trusted peers who have previously interacted with each recommended item, providing a transparent justification for every suggestion.

The complete recommendation and explanation procedure is formalized in Algorithm 1.

Algorithm 1 TTHDL++ Recommendation and Explanation Generation

```
Input: User–Trust–Tag matrix; friend–item interaction list L
Output: Top-N recommendation list and explanation vector V
for each user u ∈ 𝒰 do
    z ← Autoencoder(User–Trust–Tag[u])
    rec-list ← NeuralRecommendation(z)
    V ← ∅
    for each item i in rec-list do
        if i was interacted with by at least one trusted peer of u then
            Append corresponding peer IDs to V
        end if
    end for
end for
return rec-list, V
```

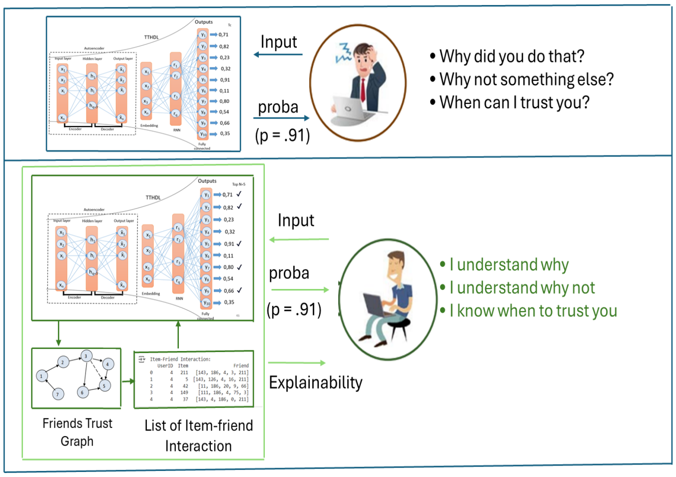

Fig. 2. Explainability mechanism based on Item–Friend interactions within the trust network.

E. Complexity Analysis

The computational cost of TTHDL++ can be decomposed across its two main phases.

Feature Extraction Phase. Given an input vector of dimension d (i.e., the size of the User–Trust–Tag representation) and a latent space of dimension h, the Autoencoder training complexity per epoch is $\mathcal{O}(n \cdot d \cdot h)$, where n is the number of users. This is comparable to the cost of standard sparse matrix factorization.

Recommendation Phase. The neural recommendation module processes the latent vector $z \in \mathbb{R}^h$ through an Embedding layer and an RNN with hidden size r, yielding a per-user complexity of $\mathcal{O}(n \cdot h \cdot r)$.

Explainability Phase. Computing the Item–Friend Interaction vector requires iterating over the trusted neighbourhood $\mathcal{F}_u$ of each user u and verifying item interactions, with complexity $\mathcal{O}(n \cdot |\mathcal{F}_u| \cdot N)$, where N is the recommendation list size. This step is lightweight relative to the neural components and adds negligible computational overhead.

Overall, TTHDL++ scales linearly with the number of users and quadratically with the hidden dimension, making it tractable for datasets of the scale examined in this work.

F. Explainability and Human–Machine Interaction

A central objective of TTHDL++ is to deliver interpretable recommendations that enhance Human–Machine Interaction (HMI). Users frequently inquire about the reasons behind a specific suggestion. By exposing trusted peers who previously interacted with the recommended item, the system provides a clear and verifiable justification.

This explanation mechanism increases transparency and strengthens user confidence in the system. When users can understand the reasoning behind recommendations, they are more inclined to accept suggestions and engage further with the platform. Thus, the integration of explainability with trust and tag information simultaneously improves recommendation quality and fosters greater user engagement.

Mathew et al. [23] present a comprehensive survey of XAI, examining existing techniques, application domains, and the main challenges associated with interpretable AI systems. Their work underscores the critical importance of transparency, trust, and ethical considerations in deploying XAI across sectors such as healthcare, finance, education, and autonomous systems. In TTHDL++, item i is recommended because one or more trusted users with compatible preferences have previously interacted with it, providing a socially grounded and transparent justification.

IV. Experimental Study

A. Datasets and Experimental Protocol

Datasets. We evaluate TTHDL++ on two publicly available GroupLens benchmarks widely used in trust- and tag-based recommendation research (Table II): *Last.fm*, a music social network, and *Delicious*, a social bookmarking platform. Both datasets provide explicit user–item ratings, pairwise trust relationships between users, and tag annotations.

TABLE II
Dataset Statistics

Dataset	Users	Items	Tags	Trust
Last.fm	1,892	17,632	11,946	12,717
Delicious	1,867	69,226	53,388	7,668

Evaluation Protocol. A stratified 75/25 train/test split is applied together with 5-fold cross-validation (`StratifiedKFold`, k=5) to ensure robust and reproducible results. Performance is measured using standard Top-N ranking metrics:

$$\text{P@}N = \frac{|\hat{R}_u \cap R_u^+|}{N}, \quad \text{R@}N = \frac{|\hat{R}_u \cap R_u^+|}{|R_u^+|}, \quad \text{F@}N = \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}}$$

where $\hat{R}_u$ is the Top-N recommendation list for user u and R_u^+ is the set of ground-truth relevant items, for $N \in \{5, 10\}$.

Implementation Details. TTHDL++ is implemented in Python using TensorFlow/Keras and trained on a standard GPU. The Autoencoder encoder comprises two dense layers (dimensions 512 and 256, ReLU activation) followed by a bottleneck of dimension 128; the decoder mirrors this structure symmetrically. The recommendation module uses an Embedding layer of size 64, a single-layer RNN with 128 hidden units, a Dropout layer with p=0.3, and a Dense output layer with softmax activation. The model is optimized with Adam (η=0.001, batch size 64) over 100 epochs, with early stopping (patience 10) based on validation loss.

Baselines. We compare TTHDL++ against the following models, all evaluated under identical conditions:

- **DLIR** [10]: a deep trust-aware initialization recommendation model based on RBM Autoencoders.
- **TTDNN** [8]: a deep neural network jointly modeling trust and tag information.
- **TRAL** [30]: an attention-based tag-aware model combining deep and linear components.

TABLE III
RESULTS ON THE LAST.FM DATASET

	Top-5			Top-10		
Model	P	R	F	P	R	F
DLIR	0.858	0.474	0.602	0.855	0.991	0.918
TTDNN	0.583	0.657	0.616	0.556	0.692	0.616
TRAL	–	–	–	–	–	–
TTHDL	0.579	0.555	0.423	0.537	0.526	0.530
TTHDL++	**0.868**	**0.881**	**0.875**	**0.860**	**0.851**	**0.855**

- **TTHDL**: an ablation of our model without the Autoencoder feature extraction phase, used to quantify the contribution of the FE stage.

Graph-based models such as LightGCN [32] and NGCF [33] were excluded from the comparison as they operate on implicit feedback without trust or tag signals, making direct evaluation under our experimental protocol unsuitable.

B. Results and Discussion

Table IV reports Precision (P), Recall (R), and F-measure (F) for Top-5 and Top-10 settings on both datasets. TTHDL++ achieves the best results across all configurations.

On *Last.fm*, TTHDL++ reaches F@5 = 0.875 and F@10 = 0.855, compared to 0.616 for TTDNN in both settings. The high Recall (0.881 at Top-5) indicates that the model effectively covers the relevant items in the user's preference profile. On *Delicious*, TTHDL++ scores F@5 = 0.711 and F@10 = 0.704, substantially outperforming TTDNN (0.632 / 0.518) and TRAL (F@10 = 0.113).

The slight Precision decrease observed when moving from $N=5$ to $N=10$ is accompanied by a Recall gain, reflecting broader coverage of relevant items—a normal and expected behavior for Top-N ranking metrics. The ablation model TTHDL consistently underperforms TTHDL++ across all settings, confirming the positive contribution of the Autoencoder-based feature extraction phase. DLIR fails to converge on Delicious due to extreme sparsity [10], and TRAL results for Top-5 on Last.fm are unavailable from the original paper.

V. CONCLUSION

In this paper, we presented **TTHDL++**, a hybrid deep learning recommender system that jointly exploits trust relationships, semantic tags, and an Autoencoder-based feature extraction phase to address data sparsity and improve recommendation accuracy. The latent representations produced by the Autoencoder are processed by the TTHDL neural architecture to generate personalized Top-N suggestions, while the Item–Friend Interaction vector provides interpretable, peer-justified explanations grounded in the user's social trust network.

By embedding explainability directly into the recommendation pipeline, TTHDL++ enhances user trust, system transparency, and platform engagement, thereby supporting more informed decision-making in human–machine interactions. Experimental results on two public benchmarks confirm that TTHDL++ consistently outperforms competitive state-of-the-art baselines across all evaluation metrics.

For future work, we plan to incorporate indirect trust propagation mechanisms, as well as explicit user feedback on both recommendations and explanations, in order to develop a more interactive, adaptive, and context-aware recommender system.

ACKNOWLEDGMENT

The authors thank the LARODEC Laboratory at the Higher Institute of Management of Tunis for their support. Language refinement assistance was provided by OpenAI's ChatGPT; all scientific content is solely the authors' own.

DECLARATIONS

- **Data availability:** Last.fm and Delicious are publicly available via GroupLens (https://grouplens.org).

REFERENCES

[1] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, A survey of active learning in collaborative filtering recommender systems. *Egyptian Informatics Journal* (2015).

[2] S. Milano, M. Taddeo, L. Floridi, Recommender systems and their ethical challenges. *AI & Society* **35**, 957–967 (2020).

[3] J. Donahue et al., Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE CVPR*, pp. 2625–2634 (2015).

[4] S.-Z. Zhang, P.-H. Li, X.-N. Chen, Collaborative Convolution AutoEncoder for Recommendation Systems (2019).

[5] N. Vo, J. Hays, Generalization in metric learning: Should the embedding layer be embedding layer? In *Proc. IEEE WACV*, pp. 589–598 (2019).

[6] M. Dong et al., A survey for trust-aware recommender systems: A deep learning perspective. *Knowledge-Based Systems* **249**, 108954 (2022).

[7] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* **52**(1), 1–38 (2019).

[8] S. Ahmadian, M. Ahmadian, M. Jalili, A deep learning based trust- and tag-aware recommender system. *Neurocomputing* **488**, 557–571 (2022).

[9] L. Wan et al., Deep matrix factorization for trust-aware recommendation in social networks. *IEEE Trans. Network Science and Engineering* **8**(1), 511–528 (2020).

[10] T. Liu, Z. He, DLIR: A deep learning-based initialization recommendation algorithm for trust-aware recommendation. *Applied Intelligence* **52**(10), 11103–11114 (2022).

[11] M. Wang et al., Trust-aware collaborative filtering with a denoising autoencoder. *Neural Processing Letters* **49**, 835–849 (2019).

[12] C.N. Dang, M.N. Moreno-García, F.D. la Prieta, An approach to integrating sentiment analysis into recommender systems. *Sensors* **21**(16), 5666 (2021).

[13] G. Bathla, H. Aggarwal, R. Rani, AutoTrustRec: Recommender system with social trust and deep learning using AutoEncoder (2020).

[14] S. Deng et al., On deep learning for trust-aware recommendations in social networks. *IEEE Trans. Neural Networks and Learning Systems* **28**(5), 1164–1177 (2016).

[15] R. Mu, A survey of recommender systems based on deep learning. *IEEE Access* **6**, 69009–69022 (2018).

[16] J. Bennett, S. Lanning et al., The Netflix Prize. In *Proc. KDD Cup Workshop*, vol. 2007, p. 35 (2007).

[17] D. Charte et al., A practical tutorial on autoencoders for nonlinear feature fusion. *Information Fusion* **44**, 78–96 (2018).

[18] K. Berahmand et al., Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review* **57**(2), 28 (2024).

[19] Y. Yang, S. Mandt, L. Theis, An introduction to neural data compression. *Foundations and Trends in Computer Graphics and Vision* **15**(2), 113–200 (2023).

[20] S. Ahmadian, M. Meghdadi, M. Afsharchi, Incorporating reliable virtual ratings into social recommendation systems. *Applied Intelligence* **48**, 4448–4469 (2018).

[21] K. Kalasampath et al., A literature review on applications of explainable artificial intelligence (XAI). *IEEE Access* **13**, 41111–41140 (2025).

TABLE IV
PRECISION (P), RECALL (R), AND F-MEASURE (F) FOR TOP-$N = 5$ AND TOP-$N = 10$ ON LAST.FM AND DELICIOUS. "–": RESULT UNAVAILABLE.

	Last.fm						Delicious					
	Top-5			Top-10			Top-5			Top-10		
Model	P	R	F	P	R	F	P	R	F	P	R	F
DLIR	0.858	0.474	0.602	0.855	0.991	0.918	–	–	–	–	–	–
TTDNN	0.583	0.657	0.616	0.556	0.692	0.616	0.473	0.956	0.632	0.451	0.609	0.518
TRAL	–	–	–	–	–	–	–	–	–	0.060	0.620	0.113
TTHDL	0.579	0.555	0.423	0.537	0.526	0.530	0.565	0.529	0.546	0.549	0.501	0.523
TTHDL++	**0.868**	**0.881**	**0.875**	**0.860**	**0.851**	**0.855**	**0.709**	**0.712**	**0.711**	**0.699**	**0.709**	**0.704**

[22] C. Laroussi, R. Ayachi, A deep meta-level spatio-categorical POI recommender system. *International Journal of Data Science and Analytics* **16**(2), 285–299 (2023).

[23] D.E. Mathew, D.U. Ebem, A.C. Ikegwu, P.E. Ukeoma, N.F. Dibiaezue, Recent emerging techniques in explainable artificial intelligence to enhance the interpretability and understanding of AI models. *Neural Processing Letters* **57**(1), 16 (2025).

[24] K. Zeeshan, The impact of regularization on convolutional neural networks. Master's thesis (2018).

[25] Q.V. Liao, D. Gruen, S. Miller, Questioning the AI: Informing design practices for explainable AI user experiences. In *Proc. ACM CHI*, pp. 1–15 (2020).

[26] M.A. Chatti, M. Guesmi, A. Muslim, Visualization for recommendation explainability: A survey and new perspectives (2023).

[27] I. Nunes, D. Jannach, A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* **27**, 393–444 (2017).

[28] P. Pu, L. Chen, R. Hu, Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction* **22**(4-5), 317–355 (2012).

[29] Z. Chen et al., Towards explainable conversational recommendation. In *Proc. IJCAI*, pp. 2994–3000 (2021).

[30] Y. Zuo, S. Liu, Y. Zhou, H. Liu, TRAL: A tag-aware recommendation algorithm based on attention learning. *Applied Sciences* **13**(2), 814 (2023).

[31] M. Ghanem, W. Guezguez, R. Ayachi, A Multimodal DeBERTa-Based Recommender System for Low-Resource and Sparse Data Environments. *International Conference on Agents and Artificial Intelligence* (2026).

[32] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proc. ACM SIGIR*, pp. 639–648 (2020).

[33] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering. In *Proc. ACM SIGIR*, pp. 165–174 (2019).