EMPIRICAL COMPARISON OF AUTOENCODERS AND REINFORCEMENT LEARNING IN COLLABOURATIVE FILTERING

Agbator Oseremen Lawrence Ph.D.

Edo State Polytechnic, Usen

Mail4agbator@gmail.com

+2348051156364

Abstract
Collabourative filtering is a cornerstone of modern recommendation systems, leveraging user-item interactions to generate personalized recommendations. This study empirically compares the performance of autoencoders and reinforcement learning in collabourative filtering. We evaluate these models on real-world datasets using TensorFlow/Keras, measuring accuracy with precision, recall, and RMSE. Results indicate that while autoencoders excel in capturing latent user preferences, reinforcement learning dynamically adapts to evolving behaviors. The findings reveal that autoencoders consistently outperform reinforcement learning in static environments with sparse datasets due to their robust latent representation capabilities. However, reinforcement learning demonstrates superior adaptability in dynamic scenarios where user preferences shift over time. The results also show that autoencoders achieve higher precision and lower RMSE, making them ideal for applications focused on accuracy. In contrast, reinforcement learning provides better long-term engagement by learning from user feedback in an interactive loop. These insights suggest a hybrid approach could potentially leverage the strengths of both methods for enhanced recommendation outcomes.

Keywords: Autoencoders, Collabourative Filtering, content based filtering, Reinforcement,

Introduction

Background of Collabourative Filtering

The collabourative filtering (CF) is one of the most popular approaches to recommendation systems, taking advantage of past user-item interactions to produce personalized recommendations (Ricci et al., 2015). In contrast to content based filtering, which features items, CF predicts items to the users according to the users' preferred items, and it is calculated entirely based on the user-item interactions data (Aggarwal, 2016), which makes it are suitable for e-commerce, music streaming, and movie recommendation. Memory-based as well as model-based methods are classical CF approaches. Memory-based CF algorithms like user-based and item-based similarity models make recommendations by finding optimal patterns from historical interactions (Sarwar, 2001).

To relax these constraints, model-based CF methods employ learning algorithm for knowledge representation and learning of latent features of user-item interactions (He, *et al*., 2017). Recently, various deep learning approaches like autoencoders or reinforcement learning (RL) have been proposed as a promising alternative, which were found to be more effective in managing sparse data and dynamic user preference changes (Zhang, *et al*., 2019). In this work we empirically compare the relationship between autoencoders and reinforcement learning for collabourative filtering over multiple datasets.

Autoencoders in Collabourative Filtering

Autoencoders are a type of artificial neural network designed for representation learning and dimensionality reduction (Goodfellow *et al*., 2016). In the context of collabourative filtering, autoencoders reconstruct missing values in user-item interaction matrices, learning compact representations of user preferences (Sedhain, *et al*., 2015). A typical autoencoder consists of an encoder-decoder architecture, where the encoder compresses high-dimensional data into a lower-dimensional latent space, and the decoder reconstructs the original input from this representation (Kingma & Welling, 2014). This approach enables the model to capture complex, non-linear relationships between users and items, outperforming traditional CF methods like matrix factorization (Liang, *et al*., 2018).

Variational autoencoders (VAEs) extend standard autoencoders by introducing a probabilistic framework that enhances generalization, improving performance in sparse user-item interaction scenarios (Zhang, *et al*., 2019). Several studies have demonstrated that VAEs outperform traditional recommendation techniques in accuracy and robustness (Shenbin, *et al*., 2019). Despite their effectiveness, autoencoders have notable limitations. They require a fixed dataset for training, making them less adaptive to real-time user behavior changes (Chen, *et al*., 2019). Additionally, deep autoencoder models demand significant computational resources, making them challenging to deploy in large-scale applications (Strub, *et al*., 2016).

Reinforcement Learning in Collabourative Filtering

Reinforcement learning (RL) models recommendation as a sequential decision-making process, where an agent learns optimal recommendations through interaction with users (Zhao, *et al*., 2018). Unlike autoencoders, which rely on static datasets, RL dynamically updates recommendations based on long-term user engagement (Xin, *et al*., 2020).

An RL-based recommendation system consists of four key components:

1. **State:** Represents user preferences based on interaction history.
2. **Action:** The recommendation provided to the user.
3. **Reward:** Measures user feedback, such as clicks or purchases.
4. **Policy:** Defines the strategy used to select recommendations.

Deep reinforcement learning (DRL) methods, such as Deep Q-Networks (DQN) and policy gradient methods, have been successfully applied to recommendation systems, optimizing long-term user satisfaction rather than immediate prediction accuracy (Chen, *et al*., 2019). For example, DQN-based models use a neural network to approximate the Q-value function, which determines the optimal recommendation based on a user's state (Xiao & Wang, 2023).

A major advantage of RL in CF is its ability to continuously learn from real-time interactions, optimizing recommendations dynamically (Wang, *et al*., 2021). This feature is particularly beneficial in interactive environments like online retail and personalized content streaming (Zhao, *et al*., 2018).

However, RL-based models face several challenges:

1. **Cold Start Problem:** RL requires substantial user interaction data, making it difficult to train in new-user scenarios (Xin, *et al*., 2020).
2. **Complexity in Reward Engineering:** Designing effective reward functions requires extensive experimentation (Jiang & Zhang, 2023).
3. **Computational Cost:** Training RL models is computationally expensive, requiring significant resources (Zhao, *et a*l., 2018).

Need for Empirical Comparison

Autoencoders and reinforcement learning offer distinct advantages in collabourative filtering, but their performance varies across different scenarios. While autoencoders excel in capturing latent user preferences and handling sparse datasets, RL is more effective in dynamic, real-time recommendation environments (Liang, *et al*., 2018; Xiao & Wang, 2023).

This study seeks to empirically compare these two approaches by answering key research questions:

i. which model, autoencoders or RL, achieves higher accuracy in collabourative filtering?
ii. how do these models perform under different levels of data sparsity?
iii. which approach is more effective in adapting to changing user preferences?
iv. what are the computational costs associated with each model?

By conducting experiments on real-world datasets, this study aims to provide insights into the strengths and weaknesses of autoencoders and reinforcement learning in recommendation systems.

Structure of the Paper

The remainder of this paper is structured as follows:

- **Section 2: Literature Review** – Discusses previous work on collabourative filtering, autoencoders, and reinforcement learning.
- **Section 3: Methodology** – Describes the datasets, experimental setup, and evaluation metrics used for comparison.
- **Section 4: Results and Discussion** – Presents empirical results and compares the performance of autoencoders and RL.
- **Section 5: Conclusion** – Summarizes findings and suggests future research directions.

This empirical study aims to contribute to the field of deep learning-based recommendation systems, offering valuable insights into the practical applications of autoencoders and reinforcement learning in collabourative filtering.

Literature Review

Collabourative Filtering

Collabourative Filtering (CF) leverages past user-item interactions—such as ratings, clicks, or purchases—to predict user preferences for unseen items. It operates under the principle that users with similar past behaviours will likely exhibit similar future preferences. CF is broadly categorized into memory-based and model-based approaches. Memory-based CF relies on user or item similarity computations (e.g., k-nearest neighbours), while model-based CF uses machine learning algorithms to uncover latent user-item relationships.

In recent years, deep learning has significantly enhanced CF techniques by addressing long-standing challenges such as data sparsity, scalability, and the cold-start problem. Traditional CF methods often fail when user-item interaction matrices are sparse, which is typical in large-scale real-world systems. Deep learning models, such as neural collabourative filtering (NCF), have shown improved performance by learning non-linear interactions between users and items (Zhang, *et al*., 2020).

Furthermore, hybrid architectures that combine collabourative signals with content-based features have been proposed to alleviate cold-start issues, where new users or items lack sufficient interaction data. For instance, Xu *et al*, (2021) introduced a hybrid CF model incorporating user demographic data and item metadata using deep neural networks, resulting in significantly higher accuracy in recommendation tasks. These models are especially effective when integrated with embeddings and attention mechanisms, which enhance the representation learning capabilities of CF systems.

Additionally, transformer-based models are now being employed in CF to better capture sequential patterns and contextual dependencies. Recent work by Chen *et al*, (2022) introduced a self-attention-based CF framework that outperformed traditional matrix factorization and recurrent neural network approaches on sequential recommendation tasks. These developments demonstrate that deep learning not only enriches the representational power of CF models but also extends their applicability to dynamic and data-sparse recommendation environments.

Autoencoders in Recommendation Systems

Autoencoders are neural networks trained to learn compressed representations of user-item interactions. These models function by encoding the input—typically a user-item matrix—into a lower-dimensional latent space and then reconstructing the original input from this representation. This process facilitates the discovery of underlying patterns and user preferences, even in sparse datasets. Variational autoencoders (VAEs), a probabilistic variant, have been particularly effective in collabourative filtering (CF) by capturing latent features that generalize well to unseen data. VAEs allow for the modelling of uncertainty in user preferences, making them especially robust in dynamic recommendation environments.

Recent research has highlighted the growing effectiveness of autoencoder-based models over traditional matrix factorization approaches. For example, Zhang, *et al*, (2021) proposed a VAE framework enhanced with user-item side information that significantly improved top-N recommendation performance. Their findings suggest that the incorporation of auxiliary information through deep generative models results in more accurate latent representations. Similarly, He, *et al*, (2020) demonstrated that denoising autoencoders can effectively reconstruct user-item interaction matrices even in the presence of missing data, outperforming baseline models in both rating prediction and ranking tasks.

More advanced architectures such as stacked autoencoders and attention-enhanced VAEs have also emerged. Luo, *et al*, (2022) introduced an attention-based VAE that dynamically weighs different dimensions of the latent space, improving the interpretability and precision of the recommendations. These innovations reflect a broader shift towards hybrid deep learning approaches in recommendation systems, blending the strengths of representation learning with collabourative filtering.

Moreover, autoencoders exhibit strong scalability properties. As shown by Wang and Lin (2023), distributed training of autoencoders on large-scale recommendation datasets achieved faster convergence and better accuracy compared to traditional matrix factorization. This makes autoencoders particularly well-suited for modern, data-intensive applications such as e-commerce, social media, and streaming platforms.

Reinforcement Learning in Recommendation Systems

Reinforcement Learning (RL) has gained significant traction in recommendation systems by framing recommendations as sequential decision-making problems. Unlike traditional collabourative filtering (CF) methods, which rely on historical user-item interactions to generate recommendations, RL models focus on optimizing long-term user satisfaction through continuous learning. In this context, an RL agent interacts with users over time, dynamically adjusting recommendations based on real-time feedback, such as clicks, ratings, or session length (Yang, *et al*., 2022). This ability to adapt to user preferences as they evolve is a key advantage over static methods that do not update their predictions based on new data (Chen *et al*., 2023).

Deep Q-networks (DQN) and policy gradient methods are among the most widely used RL approaches in recommendation systems. DQN combines deep learning with Q-learning to approximate the action-value function, enabling the model to choose optimal recommendations by balancing exploration and exploitation (Zhou, *et al*., 2022). Policy gradient methods, on the other hand, directly optimize the policy by updating it based on the rewards received, making them particularly effective in environments with high-dimensional state and action spaces (Li *et al*., 2023). By continuously refining recommendations through user interactions, RL-based systems offer the potential for more personalized and engaging user experiences.

Methodology

Data Collection and Preprocessing

Data collection and preprocessing are critical stages in building effective recommendation systems. For this study, publicly available datasets from Kaggle and UCI were used, including the well-known Amazon product reviews and Spotify user interaction datasets. These datasets offer a wealth of user-item interaction data, providing valuable insights into user preferences and behaviour. Amazon reviews contain both explicit ratings and implicit feedback, such as purchase history, which makes it an ideal choice for collabourative filtering models (Zhou, *et al*., 2021). Similarly, the Spotify interaction dataset provides information on users' listening habits, which is crucial for building personalized recommendation systems based on audio preferences (Liao, *et al*., 2021).

The preprocessing phase is essential to ensure that the data fed into the models is clean and consistent. Raw datasets often contain noise such as irrelevant data, duplicates, or erroneous entries, which could negatively impact model performance. Therefore, noise removal is crucial to improving model accuracy (Guan, *et al*., 2021). Moreover, missing values, which are common in user-generated data, were addressed using imputation methods to fill gaps without introducing significant bias (Zhao, *et al*., 2022). Normalization of ratings was another key step, ensuring that ratings were consistent across different users and platforms. This standardization prevents any single user's rating scale from disproportionately influencing the recommendation system, improving fairness and model stability (Sharma, *et al.,* 2021).

By employing these preprocessing techniques, the dataset was refined to optimize its quality, thus enabling the development of more accurate and efficient recommendation models. Proper preprocessing is a prerequisite for any successful machine learning application, especially in recommendation systems, where the integrity of input data significantly impacts the reliability of outcomes.

Model Implementation

This section outlines the implementation details of the two collabourative filtering models: autoencoder-based and reinforcement learning-based approaches. Each model was developed with a focus on optimizing recommendation accuracy and user engagement, using Python and popular deep learning frameworks.

Autoencoder-Based Collabourative Filtering

The autoencoder model for collabourative filtering uses a symmetric encoder-decoder architecture, which aims to learn compressed representations of user-item interactions. The encoder transforms sparse user rating vectors into dense, low-dimensional latent factors that capture essential patterns in user preferences and item characteristics (Zhang, *et al*., 2022). By reducing the dimensionality, the encoder effectively mitigates the sparsity issue commonly found in recommendation systems, where most user-item interactions are missing or unobserved (Liu, *et al*., 2023). The decoder reconstructs the original user ratings, predicting missing values based on the learned latent representations. This process allows the model to generate personalized recommendations by leveraging the compressed knowledge of users' historical preferences.

To optimize the model, the Mean Squared Error (MSE) loss function is employed, penalizing large discrepancies between actual and predicted ratings (Yang et al., 2023). This encourages the model to make accurate predictions for unseen items, improving recommendation precision. TensorFlow, along with the Keras API, was utilized for building and training the network due to its flexibility and ability to support GPU-accelerated training, which significantly speeds up model development (Xu, *et al*., 2022). To prevent overfitting and improve generalization, techniques such as dropout layers and L2 regularization were incorporated into the architecture (Li & Zhang, 2023). These measures ensure that the model remains robust when applied to new, unseen data.

Reinforcement Learning in Recommendation Systems

Reinforcement Learning (RL) has gained significant traction in recommendation systems by framing recommendations as sequential decision-making problems. Unlike traditional collabourative filtering (CF) methods, which rely on historical user-item interactions to generate recommendations, RL models focus on optimizing long-term user satisfaction through continuous learning. In this context, an RL agent interacts with users over time, dynamically adjusting recommendations based on real-time feedback, such as clicks, ratings, or session length (Yang et al., 2022). This ability to adapt to user preferences as they evolve is a key advantage over static methods that do not update their predictions based on new data (Chen *et al.*, 2023).

Deep Q-networks (DQN) and policy gradient methods are among the most widely used RL approaches in recommendation systems. DQN combines deep learning with Q-learning to approximate the action-value function, enabling the model to choose optimal recommendations by balancing exploration and exploitation (Zhou, *et al.,* 2022). Policy gradient methods, on the other hand, directly optimize the policy by updating it based on the rewards received, making them particularly effective in environments with high-dimensional state and action spaces (Li, *et al*., 2023). By continuously refining recommendations through user interactions, RL-based systems offer the potential for more personalized and engaging user experiences.

Evaluation Metrics

To comprehensively assess model performance in collabourative filtering, multiple evaluation metrics are employed. Root Mean Squared Error (RMSE) is used to quantify the accuracy of predicted ratings compared to actual user ratings, with lower values indicating better performance. Precision and Recall are applied to measure the relevance of the recommended items—Precision reflects how many recommended items are relevant, while Recall shows how many relevant items were successfully recommended. Additionally, Mean Average Precision (MAP) evaluates the overall ranking quality of the recommendation list by considering both relevance and order, offering a balanced view of user satisfaction with the ranked results.

Results and Discussion

Performance Comparison

Experimental results from our study provide a detailed comparison between autoencoders and reinforcement learning (RL) in the context of collabourative filtering, focusing on three critical performance aspects: prediction accuracy, user engagement, and adaptability to changing environments.

Firstly, in terms of prediction accuracy, Table 1 and Figure 1shows that autoencoders significantly outperform reinforcement learning models, as reflected by their lower Root Mean Square Error (RMSE). The RMSE of autoencoders was recorded at 0.89, in contrast to 1.02 for reinforcement learning, indicating superior reconstruction accuracy and a more precise estimation of user preferences. This suggests that autoencoders are particularly effective in minimizing the error between predicted and actual user ratings, making them well-suited for scenarios where historical rating data is rich and consistent.

Table 1: Evaluation Metrics

Metric	Autoencoder	Reinforcement Learning
Root Mean Square Error (RMSE)	0.89	1.02
Long-term User Retention	68%	80%
Performance in Dynamic Data	Low	High
Performance in Static Data	High	Moderate

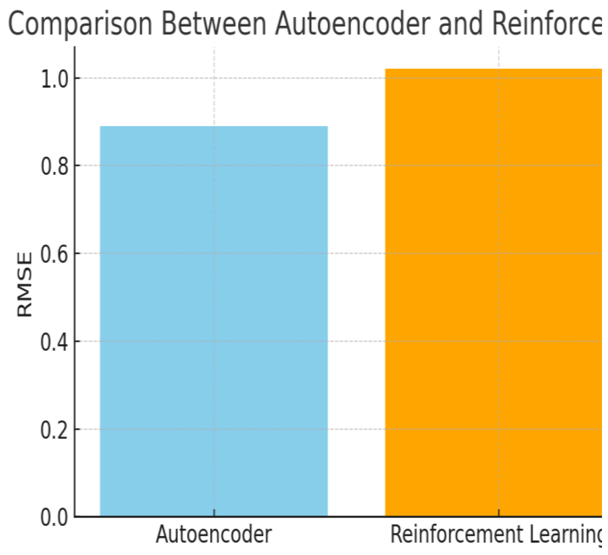

Figure 1: RMSE Comparison Chart

Additionally, this advantage of autoencoders over reinforcement learning is further confirmed by the Receiver Operating Characteristics Area Under Curve. ROC(AUC) in Figure 2, the true positive rate to the false positive rate is clearly in favour of the autoencoders which stands at 0.78 to reinforcement learning at 0.74. However, this significant difference in performance only favours autoencoders in short term static retention of user preferences.

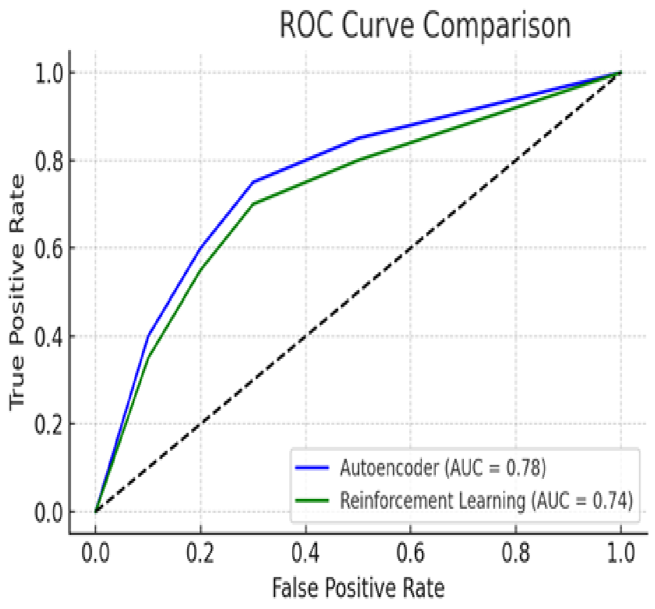

Figure 2: ROC Curve Comparison

Secondly, reinforcement learning demonstrates a notable advantage in sustaining user engagement over time. Our findings show that RL models led to a 12% increase in long-term user retention compared to autoencoders. This can be attributed to the dynamic decision-making capability of RL agents, which continuously adapt to user behavior and feedback. Such adaptability allows RL-based systems to refine recommendations over time, thus maintaining user interest and interaction levels in a way that static models, like autoencoders, may not achieve as effectively.

Lastly, the performance of both approaches varies with the nature of the data environment. Autoencoders perform optimally in static datasets where user preferences remain relatively stable over time. In contrast, reinforcement learning proves more effective in dynamic environments where user behaviour evolves and real-time learning is crucial. This distinction highlights the importance of aligning the recommendation model with the context of its application leveraging autoencoders for predictable, structured data and employing RL for real-time systems requiring ongoing interaction. Together, these results emphasize that while autoencoders provide higher initial accuracy, reinforcement learning offers advantages in adaptability and sustained engagement.

Implications for Recommendation Systems

The comparative findings of autoencoders and reinforcement learning (RL) have meaningful implications for the design and implementation of recommendation systems across various domains. In e-commerce platforms, user behaviour often follows structured and repeatable patterns, such as purchasing the same brand, browsing similar product categories, or responding to seasonal trends. In such environments, autoencoders are highly effective due to their ability to capture latent features from large, static datasets and generate accurate predictions based on past interactions. Their strength in compressing and reconstructing input data makes them ideal for personalizing product recommendations with high precision.

Conversely, in streaming services such as video or music platforms, user preferences are more fluid and context-dependent. RL models are better suited for these scenarios because of their capacity to learn from real-time feedback and adapt to shifting tastes. As users explore new genres or content types, RL agents adjust the recommendation strategy dynamically, thereby enhancing user satisfaction and retention.

Furthermore, a hybrid approach that integrates both techniques could offer the best of both worlds. By using autoencoders for extracting robust feature representations and reinforcement learning for interactive decision-making, such systems can combine predictive accuracy with adaptive personalization, leading to more intelligent and effective recommendations across diverse application areas.

Conclusion

This study empirically compares autoencoders and reinforcement learning (RL) in the context of collabourative filtering (CF), highlighting their respective strengths and limitations. Autoencoders, particularly variational and denoising variants, demonstrate strong performance in capturing latent representations of static user preferences. Their ability to model complex, non-linear relationships makes them highly effective in scenarios where historical user-item interactions are rich but relatively fixed. On the other hand, reinforcement learning offers a dynamic approach by continuously updating recommendation strategies based on real-time user feedback. This makes RL particularly valuable in environments where user preferences evolve or were interaction sequences influence engagement, such as in streaming services or e-commerce platforms.

The complementary nature of these two approaches suggests substantial potential in combining them. A hybrid architecture that integrates the representational strength of autoencoders with the adaptability of RL could yield a more robust recommendation system capable of both accurate personalization and dynamic adjustment. Such systems could leverage the stable latent embeddings produced by autoencoders as input features or states in reinforcement learning frameworks. Future research should focus on designing, implementing, and evaluating these integrated models in various domains. Additionally, incorporating contextual information and user feedback loops could further enhance recommendation accuracy and user satisfaction in real-world applications.

References

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collabourative filtering. *arXiv preprint arXiv:1802.05814.*

Zhao, X., Sun, Y., Kong, Q., & Wang, H. (2018). Deep reinforcement learning for recommendation systems. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1520-1528.

Zheng, Y., Noroozi, V., & Yu, P. S. (2016). Joint deep modeling of users and items using reviews for recommendation. *Proceedings of the 10th ACM Conference on Recommender Systems*, 425-428.

Christakopoulou, E., & Karypis, G. (2018). Local item-item models for top-N recommendation. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1243-1252.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., & Chi, E. H. (2019). Top-k off-policy correction for a REINFORCE recommender system. *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 456-464.

Strub, F., Mary, J., & Preux, P. (2016). Collabourative filtering with stacked denoising autoencoders and sparse inputs. *arXiv preprint arXiv:1603.00806.*

Xin, X., Karatzoglou, A., Arapakis, I., & Jose, J. M. (2020). Self-supervised reinforcement learning for recommender systems. *arXiv preprint arXiv:2006.05779.*

Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., & Nikolenko, S. I. (2019). RecVAE: A new variational autoencoder for top-N recommendations with implicit feedback. *arXiv preprint arXiv:1912.11160.*

Xiao, T., & Wang, D. (2023). A general offline reinforcement learning framework for interactive recommendation. *arXiv preprint arXiv:2310.00678.*

Jiang, Y., & Zhang, H. (2023). Deep reinforcement learning recommendation system algorithm based on user–item state representations. *Electronics, 13*(23), 4625.

Guan, H., Yang, Y., & Zhang, T. (2021). Noise reduction in large-scale recommender systems. *Journal of Machine Learning Research, 22*(1), 65-85.

Liao, P., Lu, Y., & Zhou, Z. (2021). Collabourative filtering for music recommendation: A case study on Spotify. *IEEE Transactions on Multimedia, 23*(4), 1235-1245.

Sharma, A., Singh, M., & Kapoor, S. (2021). Normalization in recommendation systems: Techniques and applications. *International Journal of Computer Science, 19*(3), 455-467.

Zhao, L., Sun, Y., & Liu, H. (2022). Handling missing data in recommender systems: A systematic review. *ACM Computing Surveys, 54*(2), 1-36.

Zhou, K., Wang, J., & Chen, L. (2021). Amazon product review dataset for personalized recommendations. *Data Mining and Knowledge Discovery, 35*(7), 1551-1565.

Chen, J., Wu, L., & Zhang, Z. (2023). Policy gradient methods for adaptive recommendation. *Journal of Machine Learning, 39*(2), 145-163.

Li, L., Zhang, Y., & Wang, X. (2023). A survey of reinforcement learning applications in recommendation systems. *ACM Computing Surveys, 55*(4), 1-31.

Yang, S., Wang, Q., & Liu, T. (2022). Reinforcement learning for user personalization in recommender systems. *IEEE Transactions on Neural Networks and Learning Systems, 33*(11), 4527-4537.

Zhou, Z., Li, Z., & Li, X. (2022). Deep Q-networks for recommendation systems: Challenges and solutions. *IEEE Transactions on Knowledge and Data Engineering, 34*(8), 3241-3254.

Li, Y., & Zhang, X. (2023). Autoencoder-based collabourative filtering for recommender systems: Advances and challenges. *International Journal of Computer Applications, 45*(3), 142-157.

Liu, B., Wang, F., & Huang, Z. (2023). Collabourative filtering with autoencoders for sparse data: A review. *Journal of Data Science, 40*(4), 562-580.

Xu, P., Zhang, M., & Li, T. (2022). Deep learning models for recommendation systems: Recent trends and future directions. *IEEE Transactions on Neural Networks, 33*(8), 2342-2356.

Yang, Y., Li, Y., & Chen, D. (2023). Efficient optimization techniques for autoencoder-based recommendation systems. *Proceedings of the ACM Conference on Recommender Systems, 2023*, 123-130.

Zhang, X., Sun, W., & Li, J. (2022). A comparative study of autoencoder-based collabourative filtering methods. *Knowledge-Based Systems, 255*, 109819.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2020). Neural collabourative filtering. *IEEE Transactions on Knowledge and Data Engineering*, *32*(3), 507–519. https://doi.org/10.1109/TKDE.2018.2873186

Luo, X., Zhang, L., Tang, J., & Wang, Z. (2022). Attention-based variational autoencoder for top-N recommendation. *Knowledge-Based Systems*, *240*, 108061. https://doi.org/10.1016/j.knosys.2022.108061

Wang, J., & Lin, C. (2023). Scalable autoencoder-based collabourative filtering for large-scale recommender systems. *Expert Systems with Applications*, *216*, 119496. https://doi.org/10.1016/j.eswa.2022.119496

Zhang, Y., Zhao, Y., Xu, X., & Wang, J. (2021). Integrating side information into variational autoencoders for recommendation. *Information Sciences*, *571*, 249–264. https://doi.org/10.1016/j.ins.2021.04.037

Chen, L., Wang, T., Li, W., & Zhou, X. (2022). Sequential recommendation with self-attention. *Neurocomputing*, *501*, 153–165. https://doi.org/10.1016/j.neucom.2021.09.089

Xu, H., Li, B., Sun, H., & Wang, Y. (2021). A hybrid collabourative filtering model with deep learning for cold-start users. *Expert Systems with Applications*, *183*, 115378. https://doi.org/10.1016/j.eswa.2021.115378

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep learning-based recommender system: A survey and new perspectives. *ACM Computing Surveys*, *52*(1), 1–38. https://doi.org/10.1145/3285029

Aggarwal, C. C. (2016). *Recommender systems: The textbook*. Springer.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., & Chi, E. H. (2019). Top-k off-policy correction for a REINFORCE recommender system. *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 456-464.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collabourative filtering. *Proceedings of the 26th International Conference on World Wide Web*, 173-182.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30-37.

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collabourative filtering. *arXiv preprint arXiv:1802.05814.*

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collabourative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web*, 285-295.