



Sequence-Based Explainable Hybrid Song Recommendation

Khalil Damak^{1*}, Olfa Nasraoui¹ and William Scott Sanders²

¹Knowledge Discovery and Web Mining Lab, Department of Computer Science and Engineering, University of Louisville, Louisville, KY, United States, ²Department of Communication, University of Louisville, Louisville, KY, United States

Despite advances in deep learning methods for song recommendation, most existing methods do not take advantage of the sequential nature of song content. In addition, there is a lack of methods that can explain their predictions using the content of recommended songs and only a few approaches can handle the item cold start problem. In this work, we propose a hybrid deep learning model that uses collaborative filtering (CF) and deep learning sequence models on the Musical Instrument Digital Interface (MIDI) content of songs to provide accurate recommendations, while also being able to generate a relevant, personalized explanation for each recommended song. Compared to state-of-the-art methods, our validation experiments showed that in addition to generating explainable recommendations, our model stood out among the top performers in terms of recommendation accuracy and the ability to handle the item cold start problem. Moreover, validation shows that our personalized explanations capture properties that are in accordance with the user's preferences.

Keywords: song recommendation, hybrid recommender system, recurrent neural networks, explainability, item cold start problem, deep learning, collaborative filtering, transparency and fairness in AI

OPEN ACCESS

Edited by:

Shoujin Wang,
University of Technology Sydney,
Australia

Reviewed by:

Longxiang Shi,
Zhejiang University City College, China
Ke Liu,
Beijing Normal University, China
Quanguo Zhang,
Chongqing University of Arts and
Sciences, China

*Correspondence:

Khalil Damak
khalil.damak@louisville.edu

Specialty section:

This article was submitted to
Data Mining and Management,
a section of the journal
Frontiers in Big Data

Received: 11 April 2021

Accepted: 08 July 2021

Published: 28 July 2021

Citation:

Damak K, Nasraoui O and Sanders WS
(2021) Sequence-Based Explainable
Hybrid Song Recommendation.
Front. Big Data 4:693494.
doi: 10.3389/fdata.2021.693494

1 INTRODUCTION

Among the diverse domains in which automated recommendations play an important role is music. In music, like in other domains, the most accurate recommender systems have been relying on increasingly complex (black-box) machine learning models that cannot explain their output predictions. Hence, one main challenge in designing a recommender system is mitigating the trade-off between recommendation performance (i.e., prediction accuracy) and the ability to explain predictions (i.e., explainability) (Abdollahi and Nasraoui, 2017). State-of-the-Art techniques in music recommendation include Matrix Factorization (MF)-based approaches (Mehta and Rana, 2017) and Deep Learning (DL) architectures (Zhang et al., 2017). MF builds a model that captures the similarities between users and items in a latent space obtained by factorizing the rating matrix into user and item latent factor matrices (Koren et al., 2009). Among all deep learning architectures, deep sequence models (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Lipton, 2015) are designed to model sequential data. Sequence model-based recommender systems follow three main approaches. The first approach uses sequence models to predict the next interaction given the previous interactions (Hidasi et al., 2015; Hidasi et al., 2016; Tan et al., 2016; Wu et al., 2016; Smirnova and Vasile, 2017) in a session-based fashion. The second approach uses sequence models to model the temporal dependencies, in terms of seasonal evolutions of items and user preferences, to generate recommendations (Wu et al., 2017a; Wu et al., 2017b). Finally, the third approach uses sequence models as a feature representation learning tool on textual data (Bansal et al., 2016). Despite the advances in deep learning for song recommendation and even though the sequential nature of songs

makes them naturally amenable to sequence models, no work has used sequence models with the *content* of songs for recommendation. Furthermore, current black-box music recommender systems cannot explain their recommendations based on content. On music streaming platforms, new users and songs are constantly added, and since these additions have few, if any, ratings, they cannot be handled by classical CF algorithms. This problem, known as the *cold start problem*, thus adds another challenge to collaborative filtering (CF) recommender systems in addition to the demands of recommendation accuracy and explainability (Abdollahi, 2017).

1.1 Contributions

In this work, we take advantage of the sequential nature of songs' content, the prediction power of MF, and the superior capabilities of DL sequence models to present the following contributions:

- We propose a method to transform the Musical Instrument Digital Interface (MIDI) format of songs into multidimensional time series to be the input for sequence models and, hence, capture rich information about the song;
- We integrate content-based filtering using DL sequence models with CF to build a hybrid model that provides accurate predictions compared to state-of-the-art CF recommender systems, while also providing personalized explanations and handling the item cold start problem
- We propose a new type of content-based explanation that consists of presenting a short personalized MIDI segment from the song that characterizes the portion that the user is predicted to like the most;
- We present two evaluation methodologies of the personalized music explanation segments based, respectively, on the concordance of musical sound and the preferred user tags. Given the absence of any prior technique for song explanation based on segments, our evaluation approach attempts to evaluate why a song's segment serves as an explanation for a given user; and
- We perform an online user study that demonstrates the validity of personalized segment explanations and their ability to improve user satisfaction, effectiveness, and transparency.

2 RELATED WORK

2.1 Sequence Models in Recommendation

Various recommender systems rely on sequence models (Wang et al., 2019a; Wang et al., 2019b). However, not all of them use them for recommendation with user preferences. In fact, some are session-based CF models that predict the next interaction (Hidasi et al., 2015; Tan et al., 2016; Wu et al., 2016; Hidasi and Karatzoglou, 2018; Yuan et al., 2020), or basket of interactions (Yu et al., 2016; Wang Z. et al., 2018; Wang et al., 2020a; Wang et al., 2020b), in a sequence of interactions regardless of the user's personal preferences. Similarly, other approaches relied on self-attention networks (Vaswani et al., 2017) to predict the next item recommendation given a sequence of consecutive interactions (Kang and McAuley, 2018; Sun et al., 2019; Li et al., 2020; Tan

et al., 2021). Other methods integrated content into session-based recommendation (Hidasi et al., 2016; Smirnova and Vasile, 2017) and proved that side information enhances the recommendation quality (Zhang et al., 2017). Other sequence-model-based recommender systems take into consideration the user's identification (Wu et al., 2017a; Wu et al., 2017b). These engines model temporal dependencies for both users and movies (Wu et al., 2017a; Wu et al., 2017b) and generate reviews (Wu et al., 2017a). The main objective of the aforementioned models is to predict ratings of users to items using seasonal evolutions of items and user preferences in addition to user and item latent vectors. Alternative models aimed to generate review tips (Li et al., 2017), predict the returning time of users, and predict items (Jing and Smola, 2017) or produce next item recommendations for a user by using a novel Gated Recurrent Unit (Cho et al., 2014) (GRU) structure (Donkers et al., 2017). Finally, some recommender systems use sequence models as a feature representation learning tool for text recommendation (Zhang et al., 2017). For instance, (Bansal et al., 2016), created a latent representation of items and used it as input to a CF model with a user embedding to predict ratings.

Our proposed approach differs from the aforementioned recommender systems in the goal towards which the sequence model is used. In fact, as opposed to the other approaches, we use a sequence model to encode the sequential evolution of the song content, and leverage this kind of information later in the rating prediction process.

2.2 Hybrid Song Recommender Systems

In contrast to all the aforementioned efforts, song recommendation has attracted the attention of only a few hybrid models, that differ significantly from one another in terms of the input data and the features created. In fact, music items can be represented by features derived from audio signals, social tags or web content (Vall and Widmer, 2018). Among the most noticeable hybrid song recommender systems, (Wang and Wang, 2014) learns latent factors of users and items using matrix factorization and then sums their product with the product obtained from the constructed user and song features. Meanwhile, (Benzi et al., 2016) combines non-negative MF and graph regularization to predict the inclusion of a song in a playlist. Another approach (Oramas et al., 2017) learns artist embeddings from biographies and track embeddings from audio spectrograms, and then aggregates and multiplies them by user latent factors obtained by weighted MF to predict ratings. Van den Oord et al. (2013) trains a Convolutional Neural Network (LeCun et al., 1999) on spectrograms of song samples to predict latent features for songs with no ratings. Finally, Andjelkovic et al. (2018) positions the users in a mood space, given their favorite artists, and recommends new artists using similarity measures.

2.3 Explainability in Recommendation

According to Arrieta et al. (2020), explainability can either come from transparent models or post-hoc techniques that try to explain predictions after they are generated. Explaining recommendations using transparent models can vary from

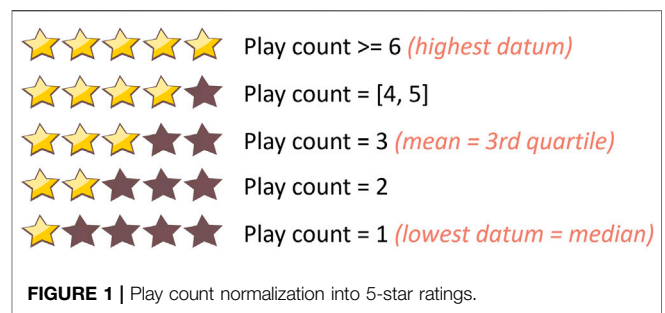
using simple user or item-based (Sarwar et al., 2001) CF approaches that rely on rating matrix similarities, to building white-box models (Zhang and Chen, 2018). The methods that are most related to our work rely either on MF or deep learning. Among the MF-based white-box models, we find (Abdollahi and Nasraoui, 2017), which optimizes a measure of explainability with the recommendation accuracy yielding explainable recommendations with user or item-based neighbor style explanations. Coba et al. (2019) and Wang S. et al. (2018) extended the idea by, respectively, trying to improve the novelty of the recommendations and modifying the calculation of the explainability matrix by integrating the neighbors' weights. Other works (Zhang et al., 2014; Zhang, 2015) used sentiment analysis on user review data along with MF-learned latent features to generate explainable recommendations. The explanations, in this case, are in the form of either user or item features (Zhang et al., 2014), textual sentences (Zhang et al., 2014), or word clusters (Zhang, 2015). On the other hand, among deep learning-based explainable models, we find (Chen et al., 2018) which uses memory-based structures, such as sequence models, to introduce users' historical records into a MF-based model. The explanations in this case are generated using an attention mechanism (Luong et al., 2015) in sequence models which provide insight on how the user's historical records affect their current and future decisions (Chen et al., 2018). For instance, (Seo et al., 2017), models user preferences and item properties using attention-based (Luong et al., 2015) Convolutional Neural Networks (CNNs) (Lecun et al., 1998) for review rating prediction. The explanation, in this case, is an importance heatmap of each word in the review. On the other hand, (Li et al., 2017), proposes a multimodal attention network that explains fashion recommendations using image regions and their correspondences with the user's review. Because there is usually a tradeoff between explainability and recommendation accuracy, some research has focused on post-hoc explainability of powerful black-box models. Such work includes (Rastegarpanah et al., 2017) which explains MF-based recommender systems using influence functions to determine the effect of each user rating on the recommendation. Cheng et al. (2019) also uses an influence-based approach to measure the impact of user-item interactions on a prediction and provides neighborhood-style explanations. Finally, Wang X. et al. (2018) proposes a model-agnostic reinforcement learning framework that was demonstrated with sentence-level explanations.

We propose a model-specific post-hoc explainable recommender system (Arrieta et al., 2020) that, aside from reaching competitive recommendation performance compared to state-of-the-art methods, succeeds in explaining a song recommendation using a personalized 10-second instrumental segment from the recommendation.

3 METHODS

3.1 Data Preparation for MIDI Content and Ratings

To build a dataset that includes both user to item interactions and song content data, we used two datasets from the Million Song



Dataset (MSD) (Bertin-Mahieux et al., 2011). The Echo Nest Taste Profile Subset (Bertin-Mahieux et al., 2011) includes 48, 373, 586 play counts of 1, 019, 318 users to 384, 546 songs collected from The Echo Nest's undisclosed partners. The Lakh MIDI Dataset, on the other hand, includes 45, 129 unique MIDI files matched to the MSD songs (Raffel, 2016a; Raffel, 2016b). We combined both datasets by taking the intersection in terms of songs. Then, we followed the same methodology used in He et al. (2017) to reduce the sparsity of the data, and filtered out users that interacted with fewer than 20 unique songs. Consequently, we ended up with a dataset consisting of 32, 180 users, 6, 442 songs with available MIDI files, and 941, 044 play counts.

We pre-processed our dataset by first mapping the play counts to ratings to remove outliers. To do so, we used the statistics of the play counts to map them to ratings as shown in **Figure 1**. Next, we created the inputs to train sequence models by transforming each MIDI file into a multidimensional time series. MIDI files are polyphonic digital instrumental audios that are used to create music. They are composed of event messages that are consecutive in time¹. Each message includes a type (such as a note), notation (the note played), time (the time it is played), and velocity (how rapidly and forcefully it is played). These events are distributed over 16 channels of information, which are independent paths over which messages travel¹. Each channel can be programmed to play one instrument. We first used "MIDICSV"² to translate the MIDI files into sheets of the event messages. We only considered the "Note on C" events to focus our interest on the sequences of notes played throughout time. In fact, the "Note on C" event represents the event of a note being played. It includes features such as the note being played, its velocity, the channel of information, and the time stamp during which it is being played. Thus, we extracted the notes that are played within the 16 channels with their velocities. As a result, each transformed multidimensional time series consists of a certain number of rows representing the number of "Note on C" events and 32 features representing the notes and velocities played within the 16 channels. The transformation process is summarized in **Figure 2**.

We then normalized the number of time steps to the median number of time steps of the songs in our dataset (2,600) to be able to train models with mini-batches (Li et al., 2014). To avoid duplicates of the same song in the input and ensure memory

¹<https://cecm.indiana.edu/361/midi.html>

²<http://www.fourmilab.ch/webtools/midicsv/>

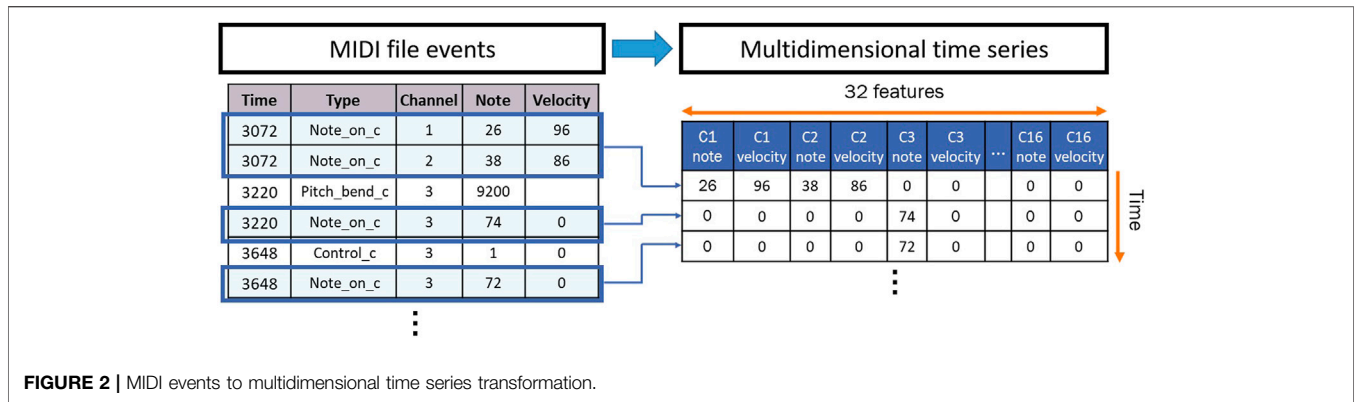


FIGURE 2 | MIDI events to multidimensional time series transformation.

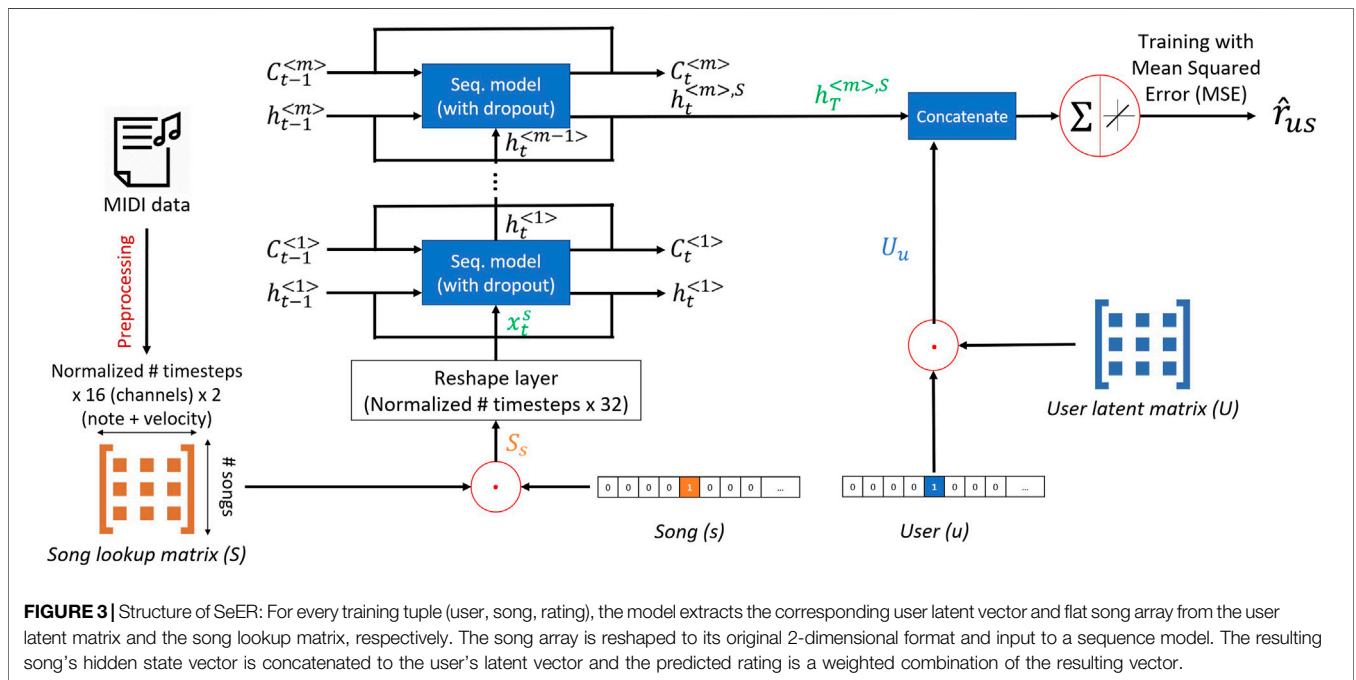


FIGURE 3 | Structure of SeER: For every training tuple (user, song, rating), the model extracts the corresponding user latent vector and flat song array from the user latent matrix and the song lookup matrix, respectively. The song array is reshaped to its original 2-dimensional format and input to a sequence model. The resulting song’s hidden state vector is concatenated to the user’s latent vector and the predicted rating is a weighted combination of the resulting vector.

efficiency, we created a song lookup matrix by flattening each multidimensional time series into one row in this matrix.

3.2 Sequence-Based Explainable Recommender System

We designed a model, that we call “SeER”: a sequence-based explainable hybrid song recommender system (Figure 3), which takes as input the song lookup matrix and a user embedding matrix. The user embedding matrix consists of learnable weights that are randomly initialized, and are updated during the model training to represent hidden characteristics of the users. For each user, song, and rating triplet (u, s, r_{us}) in the training data R , we extract the corresponding latent factor vector U_u of the user and the flattened song array S_s . The latter process is illustrated in Figure 3 with multiplications of the user embedding and song

lookup matrices with one hot vectors of u and s respectively. The song array is next reshaped into its two-dimensional original shape (2,600 time steps by 32 features). The resulting array x^s is input to a sequence model and, finally, the last layer (m^{th} layer) at the last time step ($T = 2,600$), produces the hidden state $h_T^{<m>,s}$, which is concatenated to the user’s latent vector U_u and then used as input to a neuron with a linear activation to predict a rating of the user to the song such that $\hat{r}_{us} = a[U_u, h_T^{<m>,s}]$. Where $(.,.)$ represents a concatenation and a is a weight vector. The intuition is that the weights $[a_k | k \in 1..|(U_u, h_T^{<m>,s})|]$ would regulate the flow of information coming from the user’s latent factor and the hidden state, which is a representation of the song’s content, to predict the rating. We chose the size of the hidden state to be the same as the number of user latent features to constrain the model to represent the user and the song in the same latent space size. The model is trained using the Mean

Squared Error (MSE) (Lehmann and Casella, 1998), with the loss between the actual rating r_{us} and the predicted rating \hat{r}_{us} , given by:

$$J_{SeER} = \frac{1}{|R|} \sum_{(u,s,r_{us}) \in R} (\hat{r}_{us} - r_{us})^2 = \frac{1}{|R|} \sum_{(u,s,r_{us}) \in R} [a[U_u, h_T^{(m),s}] - r_{us}]^2 \tag{1}$$

Note that in **Figure 3**, the cell states $C_t^{<m>}$ can be ignored when using Recurrent Neural Networks (Lipton, 2015) (RNNs) or Gated Recurrent Units (Cho et al., 2014) (GRUs).

3.3 Segment Forward Propagation Explainability

After generating a song recommendation s to a user u , we explain it by presenting a 10-second MIDI segment $x_u^{s,exp}$ of the song that strives to justify the recommendation using the most important portion of the recommended song for the user. To do so, we sample 10-s segments from the recommended song array x^s using a sliding window of 1 s. Then, we input the segments along with the user’s latent vector U_u to the trained model to predict a rating for each segment. Finally, the segment with the highest predicted rating is selected as the explanation for the song recommendation. The insight is that the segment with the highest predicted rating is the segment that is predicted by the model to best match the preferences of the user. Thus, it could be considered as the segment that had the most influence on the rating prediction of the song for the user. That is why we rely on it to explain the recommendation. The explanation process is summarized in **Algorithm 1**. To illustrate the SeER recommendation and explainability processes, we provide a link to a video³ demo that demonstrates the top 10 explained recommendations for user 1,000 in our dataset.

Note that the approach of learning on entire objects and then explaining using sub-objects is intuitive and commonly used in classifying objects that can be decomposed (e.g., using regions or pixels for images or words for text). We relied on the same intuition to design our Segment Forward Propagation Explainability mechanism which extends the mechanism to the music content. Also note that the MIDI format of the explanations is intended to match the type of content that the model used to generate the recommendations. Thus, when the user listens to the MIDI-based explanation, they would understand that the song recommendation was based on the MIDI (melodial) segment presented regardless of any other type of content such as the lyrics. Finally, since this is a new approach to explain song recommendations, we could not rely on any known standards for selecting the optimal segment length. Our choice of 10 s for the length of the explanation is largely justified by the fact that 10 s song previews are common on music platforms. Moreover, 10 s seemed long enough to form

a consistent explanation but still short enough to constitute a small portion of a song. We leave studying the effectiveness of various sequence lengths to future work.

ALGORITHM 1 Segment Forward Propagation Explainability

Input recommended song s , length of s in seconds L , song array x^s , user latent vector U_u , number of timesteps T , trained model $SeER$
Output Explainability segment array $x_u^{s,exp}$

- 1: **procedure** SEGMENT_FORWARD_PROPAGATION
- 2: **Step 1: Offline Pre-Segmentation:**
- 3: $abs_time_x^s \leftarrow \left\lceil \frac{MIDI_time(x_i^s)}{Division(x_i^s)} \cdot Tempo(x_i^s) \mid t = 1..T \right\rceil$ ▷ match timesteps to absolute times in x^s
- 4: $abs_time_seg \leftarrow [(i, i + 9) \mid i = 1..L - 9]$ ▷ create absolute time segments
- 5: $song_segments \leftarrow [x^{s,k} = x^s[i : j] \mid (abs_time_x^s[i], abs_time_x^s[j]) \text{ in } abs_time_seg]$ ▷ create 10 second segments of x^s
- 6: **Step 2: Online Forward-Propagation:**
- 7: $seg_ratings \leftarrow [r_{us}^k = SeER(x^{s,k}, U_u) \mid x^{s,k} \text{ in } song_segments]$ ▷ predict ratings for each segment
- 8: $x_u^{s,exp} \leftarrow song_segments[\text{argmax}_k(seg_ratings)]$ ▷ determine explainability segment
- 9: **end procedure**

4 OFFLINE EXPERIMENTAL EVALUATION

In this section, we assess the proposed model’s recommendation accuracy and ability to handle the item cold start problem by comparing it to state-of-the-art baselines. Then we validate the explanation segments with offline experiments.

4.1 Experimental Setting

We used the same 80/20% train/test split for all the experiments. We report the best results within 20 epochs in terms of recommendation ranking using Mean Average Precision at cutoff K (MAP@K) and Normalized Discounted Cumulative Gain at cutoff K (NDCG@K). We also compared the models in terms of rating prediction using the Root Mean Squared Error (RMSE). The code, data, and trained models are available for reproducibility⁴.

4.2 Hyperparameter Tuning

We fixed the number of sequence model layers to one and the batch size to 500 due to our limited memory budget. Also, we relied on the Adaptive Moment Estimation (Adam) (Kingma and Ba, 2014) optimizer. Finally, we tuned the number of latent features from 50 to 200 with increments of 50, the sequence model type by trying RNN, GRU, and Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM) and, finally, the normalized sequence lengths within the set {2,600, 1,000, 500, 300, 100}. We relied on a greedy approach, that consists of varying the hyperparameters one by one independently from each other. Note that tuning the sequence length aims to avoid overfitting and vanishing gradient issues due to long-term dependencies. We reached the best performance with 150 latent features, LSTM as the sequence model and a sequence length of 500.

4.3 Research Questions

To evaluate the prediction ability of our model, we match it to state-of-the-art baseline recommender systems regardless of their

³https://drive.google.com/open?id=1E5SZ3I6WKNKFTSIodyuzYG9S_Szppz0I9

⁴https://github.com/KhalilDMK/SeER_Keras

types and nature of input data. This leads us to formulate our first research question:

RQ1: How does our model compare to baseline recommender systems?

In addition, we run experiments to demonstrate how well our model overcomes the item cold start problem compared to pure CF models in the second research question:

RQ2: How well does our model solve the item cold start problem?

Finally, we assess whether our explanations share similar characteristics based on pure MIDI content. The logic behind this is that the shared characteristics may be interpreted as user preferences that could be captured in the explanations. This is translated in the following question:

RQ3: Do the personalized explanations share similar characteristics?

4.4 RQ1: How Does Our Model Compare to State-of-the-Art and Baseline Recommender Systems?

We compare against the following recommender system models, which include competitive state-of-the-art models (i.e., the top 2 performing models on the MSD data⁵) as well as simpler baselines:

- 1) **Matrix Factorization** (Mehta and Rana, 2017): This is one of the most common CF techniques. We optimize its loss with Stochastic Gradient Descent, and tuned the number of latent factors from 50 to 200 with an increment of 50.
- 2) **NeuMF** (He et al., 2017): This is a state-of-the-art CF technique that combines Generalized Matrix Factorization (He et al., 2017) and Multi-Layer Perceptron (LeCun et al., 1988). We replaced its output layer with a dot product and used MSE as a loss function because we are using ratings. We followed the same tuning process that was employed in He et al. (2017).
- 3) **RecVAE** (Shenbin et al., 2020): This is a state-of-the-art variational autoencoder-based implicit feedback recommender system. It is the second to best model in ranking performance on the MSD data according to⁵. We fixed the hyperparameters to the values recommended for the MSD in Shenbin et al. (2020), and tuned the latent dimension from 50 to 200 with increments of 50.
- 4) **EASE** (Steck, 2019): This is a linear model based on shallow autoencoders. It is the best model in terms of ranking performance on the MSD dataset according to⁵. It is an implicit feedback model. We tuned the regularization parameter λ within the set of values {0.5, 1, 100, 200, 500, 1,000}.
- 5) **ItemPop** (Rendle et al., 2009): This is the most popular item recommendation model, a simple baseline to benchmark the performance.

For each of the implicit feedback models, we converted the ratings into either interactions or normalized ratings in the same way described in their respective papers and compared the results in

TABLE 1 | Comparison of SeER with baseline models: Average performance over five replications. **Best scores are in bold** and second to best scores are underlined. First three models are implicit feedback models, and last three are explicit feedback models. Tukey test groups are between parenthesis, ordered from A (best) to D (worst). Based on the group based ranking, SeER ranks in the top performance group (A).

Model	MAP@5	MAP@10	NDCG@5	NDCG@10	RMSE
RecVAE	0.3622 (C)	0.4082 (C)	0.8328 (C)	0.9841 (C)	—
EASE	<u>0.4144 (A/B)</u>	0.4554(A)	0.8535(A)	<u>0.9863 (A/B)</u>	—
ItemPop	0.0978 (D)	0.1450 (D)	0.0566 (D)	0.0752 (D)	—
MF	0.3598 (C)	0.4057 (C)	0.8338 (C)	0.9842 (C)	2.4977 (C)
NeuMF	0.4109 (B)	0.4515 (B)	0.8482 (B)	0.9855 (B)	<u>1.2765 (B)</u>
SeER	0.4145(A)	<u>0.4550(A)</u>	<u>0.8528(A)</u>	0.9867(A)	1.2433(A)

terms of ranking performance. We present the average results over five replications obtained with each model in **Table 1**. We also applied a Tukey test (Haynes, 2013) for pairwise comparison for each metric and report the group-ranked results, categorized into groups ordered from A to D, from best to worst performance, that we list next to the average performances. For instance, models in group A are not significantly different from each other, but they are significantly different from models in group B; and so forth. SeER yielded the best MAP@5 and NDCG@10 scores of 0.4145 and 0.9867 respectively, and was second to best in terms of MAP@10 and NDCG@5 following EASE which is known as the best performing model so far on the MSD dataset⁵. It is worth noting that in the latter two metrics, even though SeER was second to best, the difference with EASE was not significant, as both models were in group A and were significantly better than all the other models. Furthermore, SeER presented significantly better rating prediction performance, in terms of RMSE, than all the other models (i.e., SeER was in group A while the other models were in groups B and C). Note that models belonging to group (A/B) can, statistically, be considered in either group. It is important to mention that in addition to its competitive recommendation accuracy, SeER can handle the cold start problem, as we show next, and has the unique ability to explain its recommendations. Hence, our approach is able to mitigate the cold start problem and provide explanations while still achieving state-of-the-art performance.

4.5 RQ2: How Well Does Our Model Solve the Item Cold Start Problem?

Even with no ratings, *unseen* songs can have their MIDI content propagated through the sequence model, thus allowing SeER to handle the item cold start problem. We validate the robustness to the item cold start problem by splitting the dataset into training and test sets in terms of songs. Specifically, we randomly hold out ratings related to 5% (46,069 ratings), 10% (92,347 ratings), 15% (143,535 ratings), and 20% (191,159 ratings) of the songs from the training set and use the held out songs as a test set. We made sure to include ratings from all the users in the training set to avoid user cold start issues. We assess the prediction capacity of SeER compared to only the baseline *explicit* feedback models because we cannot be guaranteed to have enough items for all the users in the test data to compute ranking metrics. Additionally,

⁵<https://paperswithcode.com/sota/collaborative-filtering-on-million-song>

TABLE 2 | Average RMSE over five replications. We compare SeER to only *explicit* feedback models (MF, NeuMF, and MLP) because we cannot guarantee to have enough items for all the users in the test data to compute ranking metrics. SeER achieves a lower RMSE compared to the other approaches, for increasing item cold start levels, which means that it is more robust in dealing with unseen items. All differences are significant (Tukey test p - values < 0.05).

	% Item cold start				
	0% (no cold start)	5%	10%	15%	20%
MF	2.4977	2.5696	2.5344	2.5100	2.5487
NeuMF	1.2765	1.3273	1.3166	1.2889	1.3237
MLP	1.2750	1.3123	1.3046	1.2750	1.3017
SeER	1.2433	1.3055	1.2914	1.2652	1.2940

we compare to a Multi-Layer Perceptron (MLP) architecture with the same hyperparameter configuration as in the MLP part of NeuMF. **Table 2** compares the average RMSE over five replications for varying item cold start levels for all the explicit feedback models. To predict a rating of an unseen song, the four models rely on the learned user's latent vectors. However, in contrast to SeER, MF, MLP, and NeuMF combine the user's latent vector with the un-updated (thus the randomly initialized) song's latent vector to generate the output. SeER has the unique ability to also employ the song's content as input to the learned sequence model and then combines the user's latent vector with the resulting item's hidden state to predict the rating. The results in **Table 2** demonstrate how SeER significantly outperforms the other baselines, namely MF, MLP, and NeuMF, for all the item cold start settings. This means that our proposed approach is more robust in dealing with unseen items, and demonstrates its ability to mitigate the item cold start problem by relying on the content of songs.

4.6 RQ3: Do the Personalized Explanations Share Similar Characteristics That Capture User-Preferences?

In order to validate the 10-s segment explanations, we tried to determine, for every user, whether their personalized explanations share some common characteristics. This is because explanations that share common properties are likely to have been generated based on user preferences that have been learned by the model. Hence, they may represent relevant sections of the recommended songs instead of being just artifacts. To study the latter property, we propose two approaches based on analysis of the concordance between song content and tags.

4.6.1 MIDI Content-Based Validation

We computed distance measures between the explanations' MIDI content to prove that they share similar characteristics. We randomly selected 100 test users, for whom we generated the top five recommendations and their explanations. We then computed the average Dynamic Time Warping (DTW) (Salvador and Chan, 2007) distance between the explanations (DTWe), which can compare multidimensional time series that do not necessarily have the same size. To compare two lists of

multidimensional time series, we compute the DTW distance matrix between them and take the average of all the values in the matrix. In the case of DTW distances between explanations, both lists are similar and include the song arrays of the explanation segments for the top-5 recommended songs. As a comparison baseline, we selected a random 10-s segment from every recommended song and computed the average DTW distance between these five segments (DTWr) for every user. Note that we compute the average DTW distances between 10-s segments instead of between the entire recommended songs to avoid any bias caused by the different song lengths. Finally, we considered the problem as a Randomized Complete Block Design (RCBD) (Olsson, 1978) and applied a Tukey test (Haynes, 2013) for pairwise comparison. **The null hypothesis** is that when averaged over all the users, the average DTW distances between the explanations (DTWe) and average DTW distances between the random segments (DTWr) are similar. For simplicity, we will call these two quantities "Avg. DTW between explanations" (or \overline{DTWe}) and "Avg. DTW between random segments" (or \overline{DTWr}). We show these average values with the 95% Confidence Intervals (CIs) of the difference ($\overline{DTWe} - \overline{DTWr}$) for SeER and the corresponding statistical test results in **Table 3**. We notice that \overline{DTWe} is significantly smaller than \overline{DTWr} (p -value < 0.05 and 0 is not in the Confidence Interval). This means that for each user, we can assert with 95% confidence that the explanations are significantly close to each other compared to the random segments. Thus, we can assert that our generated 10-s segment explanations share some common characteristics which are likely to represent the learned preferences of the user.

4.6.2 Tag-Based Validation

In addition to pure music content, tags can capture an item's properties in terms that are familiar to humans. In the case of songs, they can include genres, the era, the name of the artist, or subjective emotional descriptions. We used the tags from the "Last.fm" dataset (Bertin-Mahieux et al., 2011) provided with the MSD. These tags are available for almost every song in the MSD and amount to 522,366 tags (Bertin-Mahieux et al., 2011). In our dataset, we selected the songs that intersect with the "Last.fm" dataset and filtered the tags that occur in at least 100 songs in order to remove noisy tags. We obtained 4,659 songs with 227 tags. From the users that interacted with these songs, we filtered the ones that have at least 10 liked songs with the assumption that a rating strictly higher than three means that the user likes the song. Next, we randomly selected 100 users as our test sample. For every user, we determined the top 1, 2, and 3 preferred tags, based on the tags of their liked songs, and generated the top five recommendations with explanations using SeER.

Our objective is to determine how much the personalized recommendations and explanations match the preferred tags of every user. Thus, we needed to determine the tags of both the recommendations and the explanations, which are not necessarily in the tag dataset. To cope with this issue, we trained a multi-label classification model on our tagged dataset to predict the tags of the recommendations and explanations. The

TABLE 3 | Significance testing with 95% confidence of the difference between Avg. DTW between explanation and Avg. DTW between random segments: The explanations are significantly close to each other compared to the random segments. This means that the explanations capture and share some common characteristics that are likely to represent the learned user’s preferences.

Avg. DTW between explanations (DTWe)	Avg. DTW between random segments(DTWr)	95% CI of difference(DTWe – DTWr)	Adjusted p-value
7,949.2	8,467	(25, 1,010)	0.04

classifier is a sequence model layer with 20% dropout, followed by Multi-Layer Perceptron (MLP) (Popescu et al., 2009) layers with ReLU activations and an output layer with 227 nodes, corresponding to the 227 classes (i.e., tags), each with a Sigmoid activation function. The model is trained to optimize the Binary Cross-entropy loss to predict the probability of each tag individually in every node (Lapin et al., 2017).

To tune the tag classification model’s hyperparameters, we started with an LSTM layer followed by the output layer. We tuned the size of the hidden state from 100 to 500 with an increment of 100. Then, we tuned the number of MLP hidden layers from 1 to 5. We chose the number of nodes in the hidden layers to be the optimal size of the hidden state, which is 300. Finally, we tuned the sequence model type of the first layer by additionally trying RNN and GRU. The best model has one LSTM layer with a hidden state size of 300 followed by four MLP layers of the same size and, finally, the output layer. We reached a performance of 93.4% accuracy; and respectively, 51.8, 61.9, and 67.7% top-1, top-2 and top-3 categorical accuracy with 5-fold cross validation. We used top-k categorical accuracy (Lapin et al., 2017) because we are interested in correctly predicting the existing tags in a sparse target space. We used our trained tag classifier to predict the tags of all the recommendations and explanation segments for all the users. Then, we calculated the Average Percentage Match of the recommendations and explanations with the top 1, 2, and 3 user preferred tags.

We define the Percentage Match of a list of songs S with the top k preferred tags $T_k(u)$ of a user $u \in U$ as the percentage of songs from S including at least one of the top k preferred tags $T_k(u)$, as follows:

$$\% Match[S, T_k(u)] = \frac{100}{|S|} |(s \in S | Tags(s) \cap T_k(u) \neq \emptyset)| \quad (2)$$

$Tags(s)$ is the set of tags of the song s . In our case, the set of tags of a recommendation or an explanation is predicted using the multi-label classification model. The Average Percentage Match over all the test users is computed using:

$$Avg \% Match(S, U, k) = \frac{100}{|U|} \sum_{u=1}^{|U|} \% Match(S(u), T_k(u)) \quad (3)$$

$S(u)$ is either the set of recommendations or explanations for user u . We varied k , considered every problem as a Randomized Complete Block Design (RCBD) (Olsson, 1978), and applied Tukey tests (Haynes, 2013) for pairwise comparison. The null hypothesis for every test is whether the average percentage match of the recommendations and of the explanations with the top k

liked songs (Avg%Match (rec., U, k) and Avg%Match (exp., U, k), respectively) are equal. We show the two average percentage match values with the corresponding 90% CIs of the differences (Avg%Match (rec., U, k) - Avg%Match (exp., U, k)) and adjusted p -values of the Tukey tests in **Table 4**. We notice that for all k , the explanations match the preferred tags of the users more than the recommendations. The difference is significant for $k = 1, 2$, and 3 (CI of the difference does not include 0 and p -value < 0.1). This means that the explanations share similar properties which agree with the preferred tags of the users even more than the overall recommendations. For instance, assuming that the tags represent the genres, if the user’s preferred genre is, for instance, “Rock,” and a “Pop” song gets recommended, the explanation of that song is likely to be a “Rock” segment of the song, which means that the explanations are personalized. We show an example of a user from our test sample in **Table 5**.

5 USER STUDY EVALUATION

We performed a real-life user study⁶ that aims to evaluate the validity of our explainability process. We were granted approval from the Institutional Review Board (IRB) before conducting our user study.

5.1 Hypotheses and Research Questions

Our hypothesis is that an explanation to a relevant recommendation using our model will lead to better **satisfaction**, **effectiveness**, and **transparency** than a random 10-s segment explanation. First, “satisfaction” measures the contentment of the user with explanations accompanying a set of relevant recommendations based on their ratings. Hence, **RQ5** Does the type of explanation (i.e., personalized vs random) impact user satisfaction with the model?

Moreover, we assess “effectiveness,” which is the ability of the explanation to help the user make good decisions (Abdollahi, 2017). Finally, we evaluate “transparency” which is the comprehensibility of how the model works and its ability to

⁶This user study was performed with a slightly different version of SeER, in which the output layer is a dot product of the user and song latent representations. This was an old version of the model that we updated because of a significant gain in performance that we obtained with the current output layer. Even with this difference in the output layer, the model architecture and the explainability process are the same. Hence, any conclusion drawn from the online experiments with the old version of the model could be extended to the current model. We would also like to mention that we were unable to continue the user study, or replicate it with the updated model, because of the in-person setting of the experiment that was prevented by the COVID-19 pandemic.

TABLE 4 | Significance testing with 90% confidence of the difference between the Avg % Match of recommendations and explanations with user top k preferred tags. The results show that explanations can tell more about the recommendation since they capture a user’s expressed tag preferences.

k	Avg%Match (rec., U, k)	Avg%Match (exp., U, k)	90% CI of the difference	Adjusted p-value
1	83.43%	85.85%	(-0.0482, -0.0003)	0.096
2	94.14%	94.94%	(-0.01469, -0.00148)	0.045
3	96.16%	96.96%	(-0.01469, -0.00148)	0.045

TABLE 5 | Example of a Test User (#26647) where the explanations match the favorite tags more than the recommendations: The first recommended song is a “pop” song (in bold). However, the explainability segment is both “pop” and “rock” which matches the favorite tags of the user better than the recommendation itself (value in bold), thus validating this instance.

Recommendation	Recommendation tags	Explanation tags
1	Pop	Pop, rock
2	Pop, rock	Pop, rock
3	Pop, rock	Pop, rock
4	Pop, rock	Pop, rock
5	Pop, rock	Pop, rock
User’s top 3 tags (sorted)		Rock , pop, favorites

K	1	2	3
% Match [rec., $T_k(u)$]	80%	100%	100%
% Match [exp., $T_k(u)$]	100%	100%	100%

justify the recommendations (Tintarev and Masthoff, 2007). This suggests the following questions:

RQ6: Does our type of explanation (i.e., instrumental segment) impact perceived effectiveness of the model?

RQ7: Does our type of explanation increase perceived transparency of the model?

5.2 Experimental Procedure

The user is presented a list of 100 songs randomly selected from our dataset and is asked to rate at least 10 of them. They are also provided a link to every song so that they can listen to any songs with which they are unfamiliar. Based on the ratings, three recommendations, each with two different explanations, are generated and presented to the user. The

first explanation is generated using the Segment Forward Propagation Explainability process while the second explanation is a baseline random 10-second segment of the song. Of course, the subject does not know the difference between the two explanations, they are presented as “EXPLANATION 1” and “EXPLANATION 2” respectively. Each recommendation is accompanied with a related Likert Scale questionnaire. The questions are presented in **Table 6** as Questions 1 and 2. They aim to assess the user satisfaction with the explanation compared to the random segment, and thus, answer **RQ5**. Next, a questionnaire with general questions is presented to the user. This aims to assess the effectiveness and transparency criteria defined in the previous subsection in addition to collecting demographic data about the users to describe our sample. The latter questionnaire is presented in **Table 6** as Questions 3 to 9. Questions 3 and 4 respectively assess the effectiveness and transparency. Finally, questions 5 to 9 collect demographic data about the users. Note that in the following subsections, the results and statistics might not always match the sample size because users have the choice of not answering a question or not submitting a form.

5.3 Subject Sample

Participants (N = 30) were recruited through fliers or emails across a large, urban public university. Participant’s age (Mean = 31) ranged from 18 to 54 and there were 12 male and 15 female participants. The majority of participants were Computer Science majors (78%) followed by mathematics (7%) and education majors (15%) respectively. 74% of the volunteers somewhat or strongly agree that they cannot spend a day without listening to music. Moreover, most of the participants (74%) are familiar with recommender systems.

TABLE 6 | Survey questions.

Questions related to recommendations	
Question 1	The song segment “EXPLANATION 1” explains why someone would like the song
Question 2	The song segment “EXPLANATION 2” explains why someone would like the song
General questions	
Question 3	I will listen to the song based on a 10-s sample that I like
Question 4	The 10-s sample explanation helps me understand how the recommender system works
Question 5	What is your age?
Question 6	What is your sex?
Question 7	What is your major?
Question 8	How familiar are you with automated recommender systems?
Question 9	I cannot spend a day without listening to music

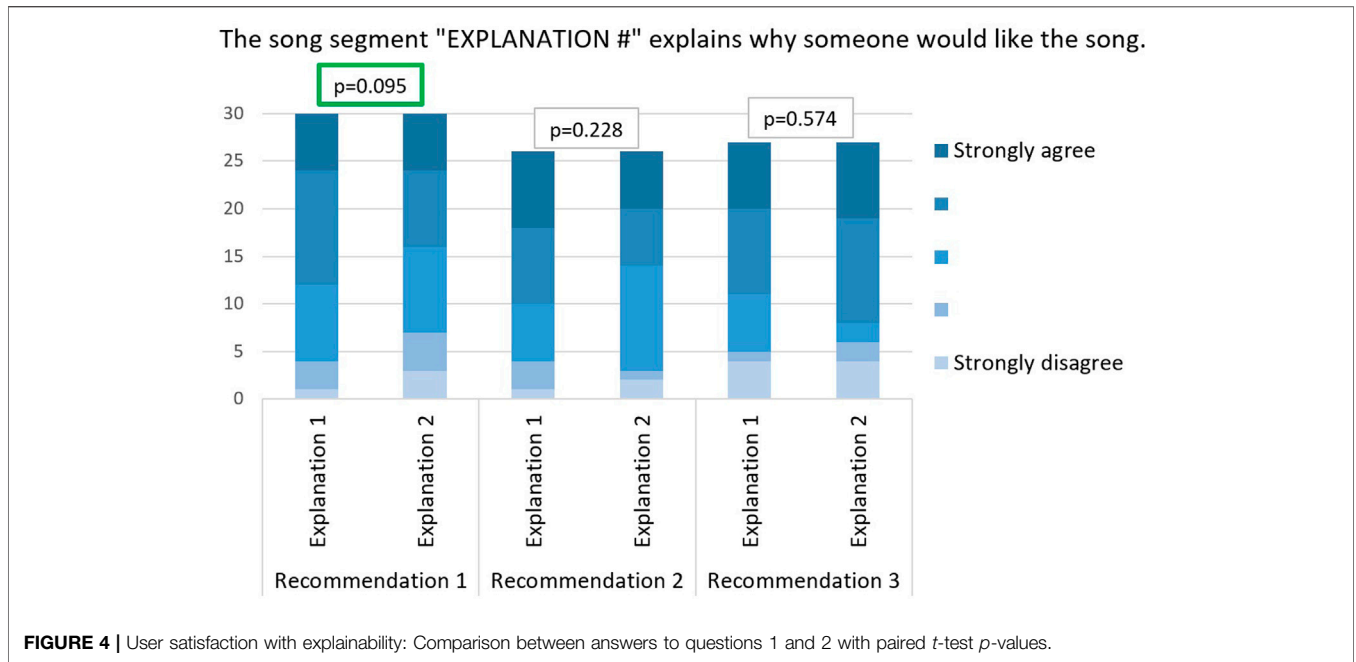


FIGURE 4 | User satisfaction with explainability: Comparison between answers to questions 1 and 2 with paired *t*-test *p*-values.

Our choice of the sample size was based on a prior prospective study. Our goal was to determine the minimum sample size necessary to detect a minimum difference between the average measures of satisfaction of the two types of explanation that corresponds to 0.5, with a power of 95%, and assuming a standard deviation of 0.7. In fact, we considered the Likert scale levels as values from 1 to 5 for our statistical tests, where 1 represents “Strongly disagree” and 5 represents “Strongly agree.” The prospective test suggested a minimum sample size of 28, that we rounded up to 30 participants.

5.4 Analysis of Results

We evaluate the explainability in terms of satisfaction (Questions 1 and 2), effectiveness (Question 3) and transparency (Question 4).

5.4.1 RQ5: Does the Type of Explanation (i.e., Personalized vs Random) Impact the User Satisfaction?

We compared our explanations (“EXPLANATION 1”) to random 10-s segments (“EXPLANATION 2”) in all three recommendations for every user. The comparison was based on the degree of satisfaction of the user towards both explanations which was measured with the two Likert scale questions 1 and 2. The answers to these questions are summarized in **Figure 4**. We can clearly notice, in recommendations 1 and 2, the abundance of the “Strongly agree” and “Somewhat agree” answers in “Explanation 1” compared to “Explanation 2.” In fact, in recommendations 1 and 2 respectively, 18 (60%) and 16 (61.5%) users agree that explanation 1 is relevant against only 15 (50%) and 12 (46.1%) that agree the same for explanation 2. However, for the third recommendation, explanation 2 was more relevant than explanation 1 (16 agreeing

participants in “Explanation 1” versus 19 in “Explanation 2”). This is probably due to the decreasing relevance of the recommendations in general as we go down in the ranked list of recommendations.

5.4.2 RQ6: Does Our Type of Explanation (Instrumental Segment) Impact the Perceived Effectiveness?

In order to study the effectiveness of our explainability process, we asked the users if they would listen to a song based on a 10-s segment that they like (Question 3). The users almost unanimously agreed (88.9%), among which 51.9% strongly agreed, with no participants disagreeing. This validates the effectiveness of our 10-s segment explainability method. The answers to Question 3 are summarized in **Figure 5**.

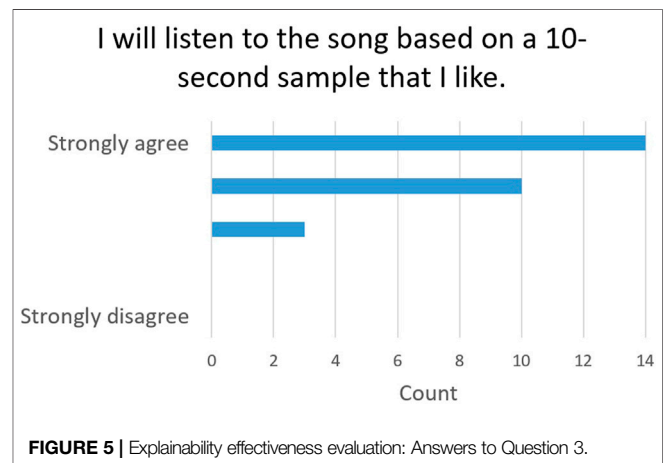
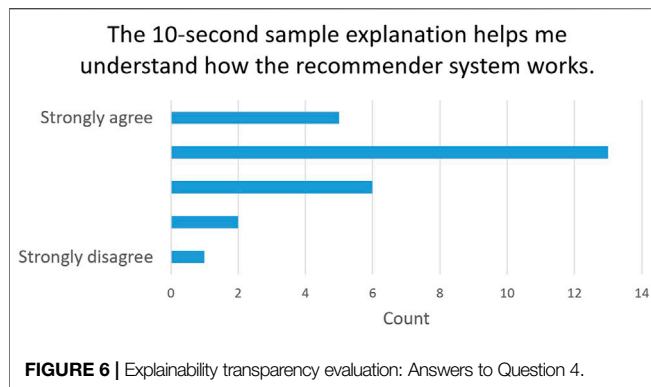


FIGURE 5 | Explainability effectiveness evaluation: Answers to Question 3.



5.4.3 RQ7: Does Our Type of Explanation Increase the Perceived Transparency?

Finally, to evaluate the impact of our explainability method in terms of transparency, we asked the users whether the 10-s segment explanation helps them understand how the recommender system works (Question 4). 18 (66.7%) users agree that the explanation improves the transparency of system [5 of them (18.5%) strongly agree]. This proves that our explanation helps the user understand how our deep learning model works. The answers to Question 4 are summarized in Figure 6.

6 CONCLUSION

We proposed a hybrid song recommender system (SeER) that combines the user ratings with the songs' MIDI content to generate both song recommendations and short MIDI segments that serve as *personalized* explanations for each

REFERENCES

- Abdollahi, B. (2017). *Accurate and Justifiable: New Algorithms for Explainable Recommendations*. Ph.D. thesis, Louisville, KY: University of Louisville
- Abdollahi, B., and Nasraoui, O. (2017). Using Explainability for Constrained Matrix Factorization. In Proceedings of the Eleventh ACM Conference on Recommender Systems (New York, NY, USA: ACM), RecSys '17, 79–83. doi:10.1145/3109859.3109913
- Andjelkovic, I., Parra, D., and O'Donovan, J. (2019). Moodplay: Interactive Music Recommendation Based on Artists' Mood Similarity. *Int. J. Human-Computer Stud.*, 121, 142, 159. doi:10.1016/j.ijhcs.2018.04.004
- Benzi, K., Kalofolias, V., Bresson, X., and Vanderghenst, P. (2016). Song Recommendation with Non-negative Matrix Factorization and Graph Total Variation. *2016 IEEE Int. Conf. Acoust. Speech Signal Process. (Icassp)*, 2439–2443. doi:10.1109/icassp.2016.7472115
- Bansal, T., Belanger, D., and McCallum, A. (2016). Ask the Gru: Multi-Task Learning for Deep Text Recommendations. In proceedings of the 10th ACM Conference on Recommender Systems, September 15th-19th, 2016, New York, NY, USA. 107–114.
- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable Artificial Intelligence (Xai): Concepts, Taxonomies, Opportunities and Challenges toward Responsible Ai. *Inf. Fusion* 58, 82–115. doi:10.1016/j.inffus.2019.12.012
- Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011b). *The Million Song Dataset*. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011).

recommended song. In addition to being the only model that can explain its recommendations, SeER stood out among the top performers in terms of recommendation accuracy, compared to baseline recommender systems. The personalized explanation segments' quality was validated by the fact that they share common properties that capture the user preferences. Moreover, the online survey-based user study validated the approach in terms of relevance, effectiveness and transparency. In addition to its good accuracy and explanation ability, SeER can handle the item cold start problem. In the future, we plan to extend our evaluation and to perform user-based studies.

DATA AVAILABILITY STATEMENT

The dataset generated and used in this article is made available, along with the source code and some pre-trained models, in the Github repository of the project: https://github.com/KhalilDMK/SeER_Keras.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work was partially supported by National Science Foundation grant NSF-1549981.

- Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., et al. (2018). Sequential Recommendation with User Memory Networks. In Proceedings of the eleventh ACM international conference on web search and data mining, Feb. 5-9, 2018. Los Angeles, CA, 108–116. doi:10.1145/3159652.3159668
- Cheng, W., Shen, Y., Huang, L., and Zhu, Y. (2019). Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage, AL, 885–893. doi:10.1145/3292500.3330857
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Association for Computational Linguistics, October, 2014), Doha, 1724–1734. doi:10.3115/v1/D14-1179
- Coba, L., Symeonidis, P., and Zanker, M. (2019). Personalised Novel and Explainable Matrix Factorisation. *Data Knowledge Eng.* 122, 142–158. doi:10.1016/j.datak.2019.06.003
- Donkers, T., Loepf, B., and Ziegler, J. (2017). Sequential User-Based Recurrent Neural Network Recommendations. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy. (New York, NY, USA: ACM), RecSys '17, 152–160. doi:10.1145/3109859.3109877
- Haynes, W. (2013). Tukey's Test. *Encyclopedia Syst. Biol.*, 2303–2304. doi:10.1007/978-1-4419-9863-7_1212
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee), WWW '17, Perth,

- Australia, April 3-7, 2017, Republic and Canton of Geneva, 173–182. doi:10.1145/3038912.3052569
- Hidasi, B., and Karatzoglou, A. (2018). Recurrent Neural Networks with Top-K Gains for Session-Based Recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino Italy, October 22-26, 2018. 843–852. doi:10.1145/3269206.3271761
- Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). *Session-based Recommendations with Recurrent Neural Networks*. arXiv preprint arXiv:1511.06939
- Hidasi, B., Quadrana, M., Karatzoglou, A., and Tikk, D. (2016). Parallel Recurrent Neural Network Architectures for Feature-Rich Session-Based Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15th-19th September 2016. (New York, NY, USA: ACM), RecSys '16, 241–248. doi:10.1145/2959100.2959167
- Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735
- Jing, H., and Smola, A. J. (2017). Neural Survival Recommender. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, February 6-10, 2017, Cambridge, UK. (New York, NY, USA: ACM), WSDM '17, 515–524. doi:10.1145/3018661.3018719
- Kang, W.-C., and McAuley, J. (2018). Self-attentive Sequential Recommendation. In 2018 IEEE International Conference on Data Mining (ICDM) (IEEE, November 17-20, 2018, Singapore), 197–206. doi:10.1109/icdm.2018.00035
- Kingma, D. P., and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR abs/1412.6980*
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 30–37. doi:10.1109/MC.2009.263
- LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. In *Shape, Contour and Grouping in Computer Vision* (Berlin: Springer). 319–345. doi:10.1007/3-540-46805-6_19
- Lapin, M., Hein, M., Schiele, B., Lapin, M., Hein, M., Schiele, B., et al. 2017). Analysis and Optimization of Loss Functions for Multiclass, Top-K, and Multilabel Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 1533–1554. doi:10.1109/TPAMI.2017.2751607
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proc. IEEE*, 86. In Proceedings of the IEEE, Nov. 1998. 2278–2324. doi:10.1109/5.726791
- LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (1988). A Theoretical Framework for Back-Propagation. In Proceedings of the 1988 connectionist models summer school. vol. 1, 21–28.
- Lehmann, E., and Casella, G. (1998). *Theory of Point Estimation* (Springer-Verlag)
- Li, J., Wang, Y., and McAuley, J. (2020). Time Interval Aware Self-Attention for Sequential Recommendation. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, Texas, USA, February 3-7. 322–330. doi:10.1145/3336191.3371786
- Li, M., Zhang, T., Chen, Y., and Smola, A. J. (2014). Efficient Mini-Batch Training for Stochastic Optimization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 24-27, 2014, New York City, NY, USA. (New York, NY, USA: ACM), KDD '14, 661–670. doi:10.1145/2623330.2623612
- Li, P., Wang, Z., Ren, Z., Bing, L., and Lam, W. (2017). Neural Rating Regression with Abstractive Tips Generation for Recommendation. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 345–354. doi:10.1145/3077136.3080822
- Lipton, Z. C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. *CoRR abs/1506.00019*
- Luong, M.-T., Pham, H., and Manning, C. D. (2015). *Effective Approaches to Attention-Based Neural Machine Translation*. arXiv preprint arXiv:1508.04025
- Mehta, R., and Rana, K. (2017). A Review on Matrix Factorization Techniques in Recommender Systems. In 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA, Mumbai, India, April 7-8, 2017). 269–274. doi:10.1109/CSCITA.2017.8066567
- Olsson, D. M. (1978). Randomized Complete Block Design. *J. Qual. Techn.* 10, 40–41. doi:10.1080/00224065.1978.11980811
- Oramas, S., Nieto, O., Sordo, M., and Serra, X. (2017). A Deep Multimodal Approach for Cold-Start Music Recommendation. *CoRR abs/1706.09739*doi:10.1145/3125486.3125492
- Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., and Mastorakis, N. (2009). Multilayer Perceptron and Neural Networks. *WSEAS Trans. Cir. Sys.* 8, 579–588.
- Raffel, C. (2016b). *Learning-Based Methods for Comparing Sequences, with Applications to Audio-To-MIDI Alignment and Matching*. Ph.D. thesis, Columbia University
- Raffel, C. (2016a). *The Lakh Midi Dataset v0.1*. Available at: <https://colinraffel.com/projects/lmd/>
- Rastegarpanah, B., Crovella, M., and Gummadi, K. P. (2017). *Exploring Explanations for Matrix Factorization Recommender Systems*
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, June 18-21, 2009. (Arlington, Virginia, United States: AUAI Press), UAI '09, 452–461.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web. 285–295.
- Salvador, S., and Chan, P. (2007). Toward Accurate Dynamic Time Warping in Linear Time and Space. *Ida* 11, 561–580. doi:10.3233/ida-2007-11508
- Seo, S., Huang, J., Yang, H., and Liu, Y. (2017). Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In Proceedings of the eleventh ACM conference on recommender systems, Como, Italy, 27th-31st August 2017. 297–305. doi:10.1145/3109859.3109890
- Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., and Nikolenko, S. I. (2020). Recvae: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, Texas, USA, February 3-7, 2020. 528–536. doi:10.1145/3336191.3371831
- Smirnova, E., and Vasile, F. (2017). Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. *CoRR abs/1706.07684*doi:10.1145/3125486.3125488
- Steck, H. (2019). Embarrassingly Shallow Autoencoders for Sparse Data. In The World Wide Web Conference, San Francisco, CA, USA, May 13-17, 2019. 3251–3257. doi:10.1145/3308558.3313710
- Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). Bert4rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In Proceedings of the 28th ACM international conference on information and knowledge management, Beijing China November 3-7, 2019. 1441–1450.
- Tan, Q., Zhang, J., Yao, J., Liu, N., Zhou, J., Yang, H., et al. (2021). Sparse-interest Network for Sequential Recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Online, March 8-12, 2021. 598–606. doi:10.1145/3437963.3441811
- Tan, Y. K., Xu, X., and Liu, Y. (2016). Improved Recurrent Neural Networks for Session-Based Recommendations. *CoRR abs/1606.08117*. doi:10.1145/2988450.2988452
- Tintarev, N., and Masthoff, J. (2007). A Survey of Explanations in Recommender Systems. In 2007 IEEE 23rd international conference on data engineering workshop, Apr. 17-20 2007, Istanbul, Turkey (IEEE), 801–810. doi:10.1109/icdew.2007.4401070
- Vall, A., and Widmer, G. (2018). Machine Learning Approaches to Hybrid Music Recommender Systems. *CoRR abs/1807.05858*.
- Van den Oord, A., Dieleman, S., and Schrauwen, B. (2013). Deep Content-Based Music Recommendation. In *Advances in Neural Information Processing Systems*. 2643–2651.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). *Attention Is All You Need*. arXiv preprint arXiv:1706.03762.
- Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M., and Lian, D. (2019a). *A Survey on Session-Based Recommender Systems*. arXiv preprint arXiv:1902.04864.
- Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., and Orgun, M. (2019b). *Sequential Recommender Systems: Challenges, Progress and Prospects*. arXiv preprint arXiv:2001.04830.
- Wang, S., Hu, L., Wang, Y., Sheng, Q. Z., Orgun, M., and Cao, L. (2020a). Intention Nets: Psychology-Inspired User Choice Behavior Modeling for Next-Basket Prediction, *Aaai*. In Proceedings of the AAAI Conference on Artificial

- Intelligence, New York, NY, USA, February 7-12, 2020. vol. 34, 6259–6266. doi:10.1609/aaai.v34i04.6093
- Wang, S., Hu, L., Wang, Y., Sheng, Q. Z., Orgun, M., and Cao, L. (2020b). Intention2basket: A Neural Intention-Driven Approach for Dynamic Next-Basket Planning. In *IJCAI*. 2333–2339. doi:10.24963/ijcai.2020/323
- Wang, S., Tian, H., Zhu, X., and Wu, Z. (2018). Explainable Matrix Factorization with Constraints on Neighborhood in the Latent Space. In *International Conference on Data Mining and Big Data*, 17–22 June, Shanghai, China. (Springer), 102–113. doi:10.1007/978-3-319-93803-5_10
- Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., and Xie, X. (2018). A Reinforcement Learning Framework for Explainable Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)* (IEEE, November 17–20, 2018 in Singapore), 587–596. doi:10.1109/icdm.2018.00074
- Wang, X., and Wang, Y. (2014). Improving Content-Based and Hybrid Music Recommendation Using Deep Learning. In *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando Florida USA November 3–7, 2014 (New York, NY, USA: ACM), MM '14, 627–636. doi:10.1145/2647868.2654940
- Wang, Z., Chen, C., Zhang, K., Lei, Y., and Li, W. (2018). Variational Recurrent Model for Session-Based Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1839–1842. doi:10.1145/3269206.3269302
- Wu, C.-Y., Ahmed, A., Beutel, A., and Smola, A. J. (2017a). Joint Training of Ratings and Reviews with Recurrent Recommender Networks. *ICLR 2017*
- Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J., and Jing, H. (2017b). Recurrent Recommender Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, Cambridge, UK, February 6–10 2017. (New York, NY, USA: ACM), WSDM '17, 495–503. doi:10.1145/3018661.3018689
- Wu, S., Ren, W., Yu, C., Chen, G., Zhang, D., and Zhu, J. (2016). Personal Recommendation Using Deep Recurrent Neural Networks in Netease. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, May 16 – 20, 2016, Helsinki, Finland), 1218–1229. doi:10.1109/icde.2016.7498326
- Yu, F., Liu, Q., Wu, S., Wang, L., and Tan, T. (2016). A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, July 17–21, 2016. 729–732. doi:10.1145/2911451.2914683
- Yuan, W., Wang, H., Yu, X., Liu, N., and Li, Z. (2020). Attention-based Context-Aware Sequential Recommendation Model. *Inf. Sci.* 510, 122–134. doi:10.1016/j.ins.2019.09.007
- Zhang, S., Yao, L., and Sun, A. (2017). Deep Learning Based Recommender System: A Survey and New Perspectives. *CoRR* abs/1707.07435
- Zhang, Y., and Chen, X. (2018). *Explainable Recommendation: A Survey and New Perspectives*. arXiv preprint arXiv:1804.11192
- Zhang, Y. (2015). Incorporating Phrase-Level Sentiment Analysis on Textual Reviews for Personalized Recommendation. In *Proceedings of the eighth ACM international conference on web search and data mining*, Shanghai, China, January 31–February 06, 2015. 435–440. doi:10.1145/2684822.2697033
- Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, Gold Coast Queensland Australia July 6–11, 2014. 83–92. doi:10.1145/2600428.2609579
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.
- Copyright* © 2021 Damak, Nasraoui and Sanders. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.