Exploiting device and audio data to tag music with User-Aware listening contexts

Karim M Ibrahim, Elena V. Epure, Geoffroy Peeters, Gael Richard

EXPLOITING DEVICE AND AUDIO DATA TO TAG MUSIC WITH USER-AWARE LISTENING CONTEXTS

Karim M. Ibrahim[1,2] **Elena V. Epure**[2] **Geoffroy Peeters**[1] **Gaël Richard**[1]
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris
[2] Deezer Research
karim.ibrahim@telecom-paris.fr

ABSTRACT

As music has become more available especially on music streaming platforms, people have started to have distinct preferences to fit to their varying listening situations, also known as context. Hence, there has been a growing interest in considering the user's situation when recommending music to users. Previous works have proposed user-aware autotaggers to infer situation-related tags from music content and user's global listening preferences. However, in a practical music retrieval system, the autotagger could be only used by assuming that the context class is explicitly provided by the user. In this work, for designing a fully automatised music retrieval system, we propose to disambiguate the user's listening information from their stream data. Namely, we propose a system which can generate a situational playlist for a user at a certain time 1) by leveraging user-aware music autotaggers, and 2) by automatically inferring the user's situation from stream data (e.g. device, network) and user's general profile information (e.g. age). Experiments show that such a context-aware personalized music retrieval system is feasible, but the performance decreases in the case of new users, new tracks or when the number of context classes increases.

1. INTRODUCTION

Since the invention of recorded music, people have been shifting from consuming music as a main activity in a live setting, to using music as a background activity as they go through the day. With the growing availability of music on streaming platforms, people developed distinct preferences for the varying listening situations, also known as context [1]. Consequently, there has been a growing interest in considering the user's situation when automatically recommending music to users.

Previous works have proposed user-aware autotaggers to infer situation-related tags from music content and user's global listening preferences [2]. However, in a practical music retrieval system, the autotagger could be only used by assuming that the context class is explicitly provided by the user. In this work, we perform a study to evaluate the feasibility of inferring the listening situation. The listening situation for our system is an activity, location, or time that is influencing the listener's preferences.

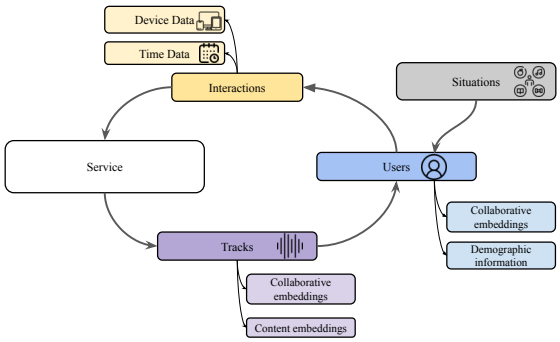

Figure 1. The available data to online music streaming services.

The process of music streaming from the perspective of our proposed approach can be found in Figure 1. We find that the music service is informed of the users, their track history, plus their past and current interactions with the service, i.e. the device and time data sent during an active session. However, the service is unaware of the influencing listening situation. Our goal is to utilize the available information for the service to infer the listening situation and the suitable tracks for the inferred situation. We propose an approach that infers the potential context from the user interactions in near real time, while the tagging of tracks with their potential listening situation happens in the background using autotaggers. Both systems are user-aware.

Our contributions in this paper are: 1) a large dataset of tracks, device data, and user embeddings labeled with their situational use through a rigorous labelling pipeline; 2) an extended evaluation of music autotaggers in predicting personalized situational tags in various scenarios; 3) a simple, yet effective model that ranks the potential listening situations for a given user based on the transmitted data from the device to the service.

2. RELATED WORK

Our proposed approach is related to two different problems: music autotagging with contextual tags, and instant

prediction of the user's listening situation. Previous work has already showed that listening situation (i.e. context) has a strong influence on the user's preferences [1, 3–5]. Hence, context has become an important factor for reaching a personalized user experience [6].

On one hand, music content is highly complex and is often challenging to be analyzed and described in human readable terms. This missing link between the content of the music and a set of semantic descriptors is referred to as the "semantic gap" [7]. One common way, which is often used when searching for or organising music, is the intended listening situation [8]. Unlike most tags that depend solely on the music content, certain tags depend also on the user [9, 10]. There has been a recent work on predicting personalized situation-related tags from music content and user embeddings [2], which we adopt here too.

On the other hand, the listening situation, e.g. activity or location, can change frequently, which leads to changes in user preferences. Explicitly inferring the listening situation is a challenging task that has only been studied on a small scale [11]. We aim at addressing this missing link by performing an extensive study on predicting the listening situation using available device data. In order to employ the personalized autotagging approach in an actual real-world setting, it is also important to be able to predict when a specific listening situation is being experienced.

3. OBJECTIVE AND PROPOSED APPROACH

A *session* consists of a sequence of audio-tracks a a given user u is listening to over time t on a music streaming service in a continuous time span [1]. A session is therefore defined as a sequence of *streams* which are each a tuple (audio-track a, user u, device data $\mathbf{d}_u^{(t)}$).

A *situational (or contextual) session* is a session resulting from listening to tracks in a certain situation (or context) c such as "gym". However, in our case, in order to gather a ground-truth dataset, we consider that a situational session can also result from listening to a playlist that contains a context-related keyword in the title [2]. A situational (or contextual) session is defined as a sequence of tuples (audio-track a, user u, device data $\mathbf{d}_u^{(t)}$, situation c).

Our objectives is to propose for a given user u a session (sequence of audio-tracks a) that fits their current situation c. However, since we don't know its current situation c we estimate it based on its device data $\mathbf{d}_u^{(t)}$ (such as the time of the day, day of the week, or type of network connections).

3.1 Proposed approach

To do so, we first estimate for each pair audio-track/user (a, u) its situation $\hat{c}_{a,u}$. In other words, we estimate in which situation c the user u would intend to use the track a. This is done using an Audio+User based Situation Prediction (`auSP`) trained to estimate situation tags c given as input a pair (audio-track a, user u). This is done offline on the server side and stored in a database.

We then estimate in real-time (with a lightweight model on the client side) for a given user u and the transmitted data from its device to the service $\mathbf{d}_u^{(t)}$, its potential current situation $\hat{c}_{d,u}$. This is done using a Device+User based Situation Prediction (`duSP`) trained to estimate situation tags c given as input a pair (device-data $\mathbf{d}_u^{(t)}$, user u). `duSP` provides us with a list of the most likely situations $\hat{c}_{d,u}$ (ranked from the most to the less likely).

Finally, to create the situational playlist, we simply select the audio tracks a for which situation $\hat{c}_{a,u}$ matches the most-likely current situations of the user $\hat{c}_{d,u}$.

Figure 2 indicates the overall architecture.

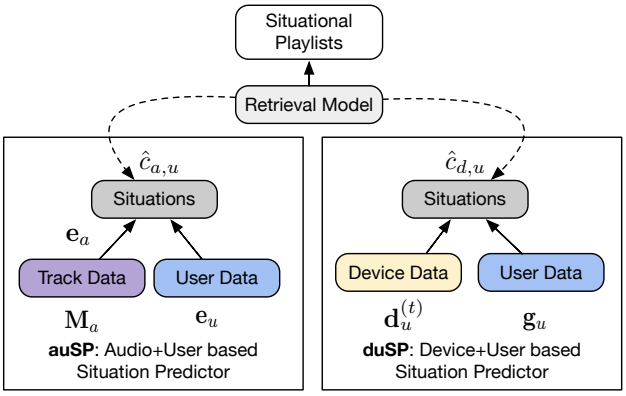

Figure 2. Overview of the system to generate a situational playlist. The left side (`auSP`) tags each track/user pair with a situational tag. The right side (`duSP`) ranks the potential situations for a device/user pair to be presented to the user.

3.2 Data description

We first describe what are exactly the data for the tracks a, the users u and the devices.

Track data $\mathbf{M}_a$. For each audio-track a, we retrieve its 30 s. snippet from the Deezer API. We represent a by its 96 Mel-bands $\times$ 646 frames matrix $\mathbf{M}_a$.

User data $\mathbf{e_u}$ and $\mathbf{g}_u$. Representing the users can be achieved through various versatile techniques. Consistent with our requirements (lightweight model and preserving privacy), we choose to represent the users using the basic data available during streaming. We use two different representations of the user that will be used for estimating $\hat{c}_{a,u}$ and $\hat{c}_{d,u}$ respectively.

For the `auSP` (estimation of $\hat{c}_{a,u}$) we use a user embedding $\mathbf{e_u}$. Similar to previous works on `auSP`, we used the users' listening history to derive user embeddings that encode their listening preferences. We compute these embeddings through matrix factorization of the user/track interactions matrix, leading to a 128-d embedding vector per user, which in commonly used to generate representations [13]. The constructed matrix uses all the tracks available in the catalogue to model the user preferences, i.e. it

[1] without any break longer than a pre-defined gap, defined here as 20 minutes as proposed by [5]

[2] The underlying assumption is that if the user started streaming a playlist with a certain title related to a situation (or context), most likely the intention of the user was to play something suitable for that situation (or context) [12].

Table 1. Summary of the notations

Symbol	Definition	Dimension
a	an audio track	
$\mathbf{M}_a$	Mel-spectrogram of a	$\mathbb{R}^{96\times646}$
$\mathbf{e}_a$	Embedding of a	$\mathbb{R}^{256}$
u	a user	
$\mathbf{e}_u$	Embedding of u	$\mathbb{R}^{128}$
$\mathbf{g}_u$	Demographics data of u	$\mathbb{R}^3$
$\mathbf{d}_u^{(t)}$	Device data of u at time t	$\mathbb{R}^8$
c	a situation (or context)	
s	a stream, a tuple $(a, u, \mathbf{d}_u^{(t)}, c)$	

is not computed exclusively with the tracks included in our dataset. The computed embeddings will be published with the dataset for reproducibility.

For the `duSP` (estimation of $\hat{c}_{d,u}$), we use the basic demographic data $\mathbf{g}_u$ of the user recorded during registration. This data is composed of: `|age,country,gender|`. While this data selection is prone to errors and short of fully representing the users, it is consistent with our requirements of using basic always-available data.

Device data $\mathbf{d}_u^{(t)}$. We collect only basic data sent by the device to the service and selected those that deemed relevant to the situation prediction. The data are: the time stamp (in local time), day of the week, device used and network used. Additionally, we extend the time/day data with circular representation of the time and day similar to the one used in [14]. The final feature vector representing device data is made of 8 features: `linear-time, linear-day, circular-time-X, circular-time-Y, circular-day-X, circular-day-Y, device-type, network-type`. The `device-type` can be: `mobile`, `desktop` (e.g. a laptop), or `tablet`. The `network-type` can be: `mobile` (a connection through cellular data), `wifi` (a WiFi connection), `LAN` (a connection through wired Ethernet), or `plane` (an offline stream from a device without a connection).

4. COLLECTING THE DATA

Pichl et al. and Ibrahim et al. proposed methods for labelling streaming sessions with a situational tag by leveraging playlist titles [2, 12]. Although sometimes prone to errors and false positives, playlist titles provide an appropriate proxy for labelling streams with tags [2, 12]. Users create playlists with a common theme according to their use [12]. One common theme of these playlists is the listening situation.

First, we collected a set of situational keywords used previously in the literature [1, 11, 15]. We extended these keywords by adding synonyms and hashtags that are frequently used on Twitter to refer to music listening. Afterwards, we retrieve from all public playlists from Deezer [3], an online music streaming service we were given access to, those playlists that include any of the keywords in their "stemmed" title. We then filtered out playlists that contained more than 100 tracks [4] or where a single artist or album represented more than 25% of the playlist, similar to [2].

From the extensive list of situational keywords and their corresponding playlists, we settled on three different subsets with an increasing number C of situational tags (4, 8, and 12): `work, gym, party, sleep | morning, run, night, dance | car, train, relax, club`.

These tags were selected by popularity [5]. We used these situations as independent tags without attempting to merge potentially similar activities and places (e.g. "party" and "dance"). Working with three situational tag sets (of increasing C) allowed us to observe how the system performs as the complexity of the problem increases.

We then focused on the users who actively listened to these playlists and retrieve the device data of those users while they were actively listening to the playlist. This resulted in a set of streams each described by an audio-track a, a user u, a playlist with a situational keyword c in the title, along with the device data $\mathbf{d}_u^{(t)}$ sent during this stream. Note that an audio-track/user/device triplet have a joint tag, none of them are tagged individually.

To ensure high quality data, we selected the month of August 2019 for inspection, because this period had more stable use patterns, before the Covid-19 pandemic. We had access to data from two locations: France and Brazil. These two locations were provided because they have the most active users in Deezer, while being in two distinctive time zones and seasons. This allowed us to perform our study on diverse data with different sources and patterns. We release the dataset [6] along with the code [7].

4.1 Dataset Analysis

As a sanity check on the collected data, we plot the distribution of the situations c across the different device-data. Figure 3 shows the ratio of the used `network-type` to connect to the service across situations c. We observe variations that correspond to what is expected from each situation, i.e. outdoors vs. indoors. However, we also find certain networks that do not match the expectations, e.g. LAN connections in a car situation, which represents noise in the dataset that can be a continuation of already existing sessions that moved indoors. Figure 4 represents the used `device-type` across situations c. We notice that most users overwhelmingly use mobile device in most cases, with small variations that also match expectations of indoor and outdoor situations. Finally, Figure 5 shows the distribution of all situations for each hour of the day. Similarly, we find predictable patterns for each situation ranging from night-related situation in the early hours that gradually progress as the time passes. These patterns support

[3] `www.deezer.com`

[4] to increase the chance that playlists reflect a selection of situation-related tracks, and not randomly added ones

[5] Estimated as the number of corresponding playlists in the service catalogue.

[6] `https://zenodo.org/record/5552288`

[7] `https://github.com/Karimmibrahim/Situational_Session_Generator`

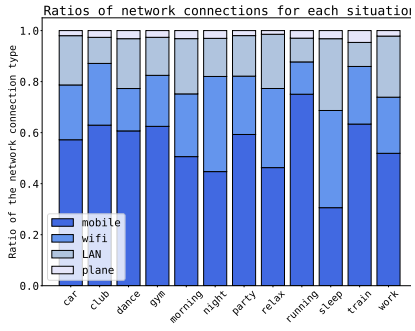

Figure 3. Network across situations c

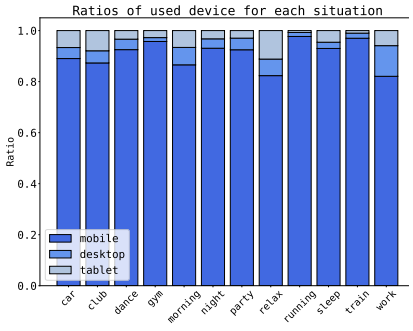

Figure 4. Device across situations c

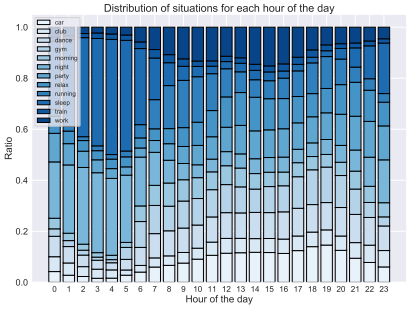

Figure 5. Distributions of situations c over hours of the day

the hypothesis of using playlist titles as proxy for inferring the actual listening situation.

5. DETAILED MODELS DESCRIPTION

5.1 Audio+User based Situation Prediction (auSP)

The auSP estimates the probability of each situation $c \in \{1 \dots C\}$ given a pair (track a represented by $\mathbf{M}_a$, user u represented by e_u): $P(c|\mathbf{M}_a, \mathbf{e_u})$. It is implemented as a Deep Neural Network $\hat{c}_{a,u} = f_\theta(\mathbf{M}_a, \mathbf{e}_u)$ with softmax output and trainable parameters θ. To train it we use the set of training streams represented as tuples $(\mathbf{M}_a, \mathbf{e}_u, c)$ We train it by minimizing the categorical cross-entropy $\mathcal{L}(\hat{\mathbf{c}}_{a,u}, \mathbf{c}, \theta)$ where $\hat{\mathbf{c}}_{a,u}$ is the estimated probability and $\mathbf{c}$ the one-hot-encoded ground-truth.

Practical implementation. The **audio input**, $\mathbf{M}_a$, is passed to a batch normalization layer then to 4 layers each made of a convolutional (CNN) and a Max-pooling operation. The CNNs have various numbers of filters (32, 64, 128, 256) but each with the same size (3x3). They are each followed by a ReLU. All Max-poolings are (2x2). The flattened output of the last CNN layer is passed to a fully connected (FC) layer with 256 units followed by a ReLU. The output of the audio branch $\mathbf{e}_a$ is a 256-d audio embedding vector $\mathbf{e_a}$. The **user input**, $\mathbf{e}_u$, is processed through 2 FC layers each with a ReLU. This output is then concatenated with $\mathbf{e}_a$ and passed to a FC layer with ReLU activation, and a dropout (with 0.3 ratio) for regularization. The final layer is made of C output units with a Softmax activation function, where C is the number of situations to be predicted. We train the model until convergence by minimizing the categorical cross entropy, optimized with Adam [16] and a learning rate initialized to 0.1 with an exponential decay every 1000 iterations.

5.2 Device+User based Situation Prediction (duSP)

The duSP estimates the probability of each situation $c \in \{1 \dots C\}$ given a pair (device-data d represented by $\mathbf{d}_u^{(t)}$, user u represented by $\mathbf{g}_u$): $P(c|\mathbf{d}_u^{(t)}, \mathbf{g}_u)$. It is implemented as a function $f_\gamma(\mathbf{d}_u^{(t)}, \mathbf{g_u})$ with Softmax output and trainable parameters γ. To train it we use the set of training streams represented as tuples $(\mathbf{g}_u, \mathbf{d}_u^{(t)}, c)$

Practical implementation. In choosing a real-time "light" duSP model, we prioritize the computational complexity requirements over accuracy. The low dimensional input features (11-d = 8 device features + 3 demographic features) already provide a strong case for the investigated models. For our implementation, we experimented with different classifiers: Decision Trees, K-Nearest Neighbors, and eXtreme Gradient Boosting (XGBoost) [17]. While all gave comparable results, we chose XGBoost for its consistent performance across splits and different evaluation scenarios. Similar to the autotagger model, the output predictions depend on the number C of situations in the dataset.

6. EVALUATION

We evaluate here the performance of our system which aims at proposing for a given user u a session (sequence of audio-tracks a) that fits their current situation c. For this, we first evaluate the performance of the two branches of our system (auSP and duSP) to correctly estimate the situation c. We evaluate this using various numbers of situations: $C \in \{4, 8, 12\}$. To evaluate the auSP, we use the common AUC (Area Under Roc Curve) and Accuracy performance measures. To evaluate the duSP, we use the Accuracy but also the Accuracy@K. This measures the capability of duSP to include the correct situation in the top K predictions. We then evaluate the global system by measuring the overlap of correct predictions between the auSP and duSP branches. This accuracy can be interpreted as the ratio of existing streams that would have occurred in these sessions if the playlists were generated with this system instead.

6.1 Scenario

We approach the evaluation of this system from two different perspectives: 1) evaluating the system on its capability of learning and generalizing, 2) evaluating the proposed system in a stable use-case with frequent users/tracks.

We simulate these scenarios through a different split criteria for the test-set. Let the full set of streams in our collected dataset S, where each stream s has a user u and a track a. We will be referring to the training-set as S_{train}, the test-set as S_{test}, the set of unique users in training and

testing as U_{train} and U_{test} respectively, and similarly the unique audio-tracks in the splits as A_{train} and A_{test}.

To evaluate the system intelligence and fit to the data, we restrict the evaluation splits to include either: 1) new users (cold-user case): $S_{test} = \{s|u \notin U_{train}, a \in A_{train}\}$, 2) new tracks (cold-track case): $S_{test} = \{s|u \in U_{train}, a \notin A_{train}\}$. We exclude the specific case of both new tracks and new users because splitting the data with only new user/track pairs in the testset is difficult and rare to find. Additionally, recommending a new track to a new user is not a common nor practical scenario to use for evaluating a system.

To evaluate how the system would perform in a regular use-case (warm case): $S_{test} = \{s|u \in U_{train}, a \in A_{train}, s \notin S_{train}\}$. The regular use-case does not restrict the system to neither new users nor tracks. However, the test-set contains exclusively new streams, i.e. (user/track) pairs, not present in the training-set. The evaluation of this regular use-case is relatively complex and includes several entwined evaluation criteria. The goal is to compare the overlap of generated sessions with groundtruth sessions.

6.1.1 auSP Evaluation

The results for the auSP can be found in Table 2.

As shown, the model can reach satisfying performance relative to the evaluation scenario. In terms of AUC, the model's fit for both new users and tracks in the cold user/track splits is not significantly impaired compared to the warm case. The performance decreases evidently as the problem gets harder with more situations C to tag, though in some cases it increases given the increase of dataset size from additional situations. In terms of accuracy, the model's performance in the intended use-case, i.e. warm case, is satisfying (Accuracy above 70). That is to say, the system can correctly tag around two thirds of the user/track listen streams with their correct situational use, when it has seen the user or the track before, but not jointly.

Note that this accuracy was computed by selecting the most probable situation from the predictions. While the high values of AUC (above 0.94) suggest a threshold optimization is needed for each class, in real use-case we do not necessarily need a threshold. The prediction probability could be used directly to retrieve tracks, e.g. by ranking tracks with the prediction probabilities and include top ranked tracks in the generated sessions. However, this max-probability threshold is needed for further evaluations with the situation predictor and with the sequential retrieval model.

Additionally, Figure 6 represents the confusion matrix obtained in the $C = 12$ and warm case.

6.1.2 duSP Evaluation

The results for the duSP can be found in Table 3. We find that predicting the situation for new users becomes noticeably harder. In the case of C=12 situations, the system was able to correctly predict the situation for only 25% of the streams. However, when the system is allowed to make multiple guesses (Accuracy@3), the accuracy evidently increases. In the case where the user is to make the last decision, the system is able to include the correct situation in the top 3 suggestions 96%, 80%, and 68% in the cases of C =4, 8, and 12 situations respectively. The choice of K, when evaluating with accuracy@K, can be obviously changed, and the performance will increase as K increases. We choose to display the results for K=3 since 3 is around the number of visible items in the carousels displayed by most streaming services on the suggestions screen on mobile devices.

Additionally, Figure 7 shows the confusion matrix obtained in the C=12 situations and warm case. We observe that the confusion is mostly coherent with the statistic shown earlier of the distribution of situations with the device data. Situations that are likely to originate with similar device data are harder to discriminate than the rest. For example, we observe a cluster of night-related situations including night, sleep, and relax situations. Similarly, outdoors situation are also often confused together. Discriminating those situations is hindered by the limited data available. However, the convenience of recommending top k situations provides as easy solution to further discriminate between these similar situations.

Table 2. Results of the auSP evaluated with AUC and Accuracy in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). The results are shown as mean(std.).

C	AUC		
	Cold User	Cold Track	Warm
4	0.889 (.009)	0.873 (.013)	0.959 (.013)
8	0.815 (.005)	0.866 (.007)	0.945 (.007)
12	0.852 (.004)	0.824 (.012)	0.941 (.012)
C	Accuracy		
	Cold User	Cold Track	Warm
4	69.72 (1.07)	63.77 (2.33)	83.75 (2.33)
8	47.56 (0.53)	52.44 (2.31)	70.81 (1.45)
12	52.68 (1.25)	37.61 (3.47)	69.14 (3.79)

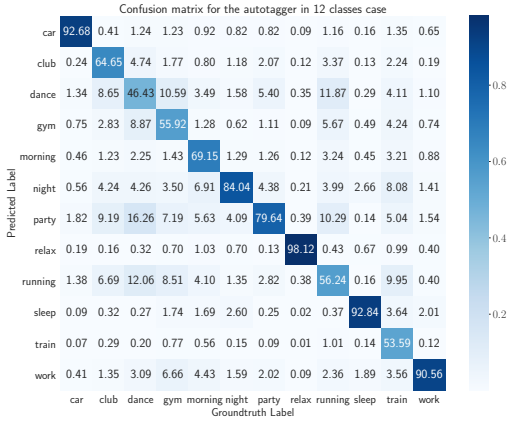

Figure 6. Confusion Matrix of the auSP in the case C=12 and warm case

Table 3. Results of the duSP evaluated with Accuracy and Accuracy@3 in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). The results are shown as mean(std.).

C	Accuracy		Accuracy @3	
	Cold User	Warm	Cold User	Warm
4	47.46 (0.98)	66.96 (0.39)	90.51 (0.31)	96.3 (0.1)
8	30.95 (0.89)	49.23 (0.16)	64.11 (1.42)	79.62 (0.13)
12	25.00 (0.29)	39.92 (0.13)	52.04 (0.61)	67.62 (0.21)

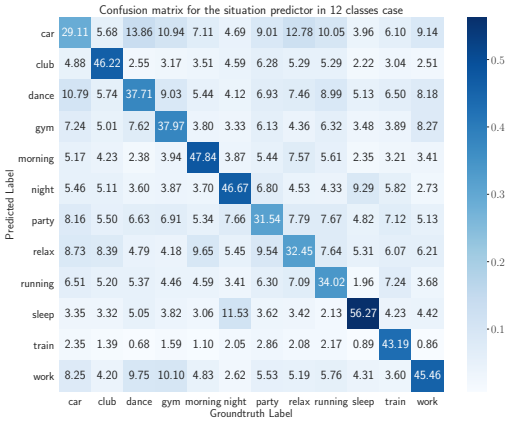

Figure 7. Confusion Matrix of the duSP with C=12 and warm case

Finally, to evaluate the challenge in classifying situations from multiple sources, we compare between the evaluation results in each location (France, Brazil) separately. We compare between two different cases: 1) a model trained globally on the data from both locations but tested locally, 2) a model trained locally on each location independently and tested on the corresponding location. Table 4 shows the results for this evaluation setting. We find that training the models locally slightly improves the results, but not significantly. This suggests that using a single unique model for all locations gives comparable results to using multiple local models. We also observe a clear distinction in the accuracy between the two locations, where Brazil scores higher than France in all cases. This is due to the larger number of users in our dataset who are in France, i.e. there are more users with more distinct patterns in the France case.

Table 4. Evaluation results of the globally and locally trained models for each of the two locations in our dataset, France and Brazil, evaluated with accuracy at each subset of situations in the warm case. The results are shown as mean(std.).

C	France		Brazil	
	Global	Local	Global	Local
4	53.4 (0.2)	55.1 (0.2)	59.2 (0.2)	61.7 (0.2)
8	35.8 (0.1)	36.8 (0.1)	45.9 (0.1)	48.2 (0.1)
12	27.6 (0.1)	28.2 (0.1)	39.6 (0.2)	41.8 (0.1)

Table 5. The joint evaluation results of the auSP and duSP and their overlapping predictions evaluated with Accuracy in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). The results are shown as mean(std.).

Model	Cold Users	Cold Tracks	Warm Case
	4 Situations		
auSP	69.73 (1.07)	63.78 (2.33)	83.75 (2.33)
duSP	47.46 (0.98)	66.81 (0.35)	67.20 (0.26)
Overlap	36.22 (1.27)	44.60 (1.01)	**58.92** (1.71)
	8 Situations		
auSP	47.56 (0.53)	52.44 (2.31)	70.81 (1.45)
duSP	30.95 (0.89)	49.13 (0.24)	49.35 (0.19)
Overlap	17.77 (0.49)	28.94 (1.24)	39.52 (1.27)
	12 Situations		
auSP	52.68 (1.25)	37.61 (3.47)	69.14 (3.79)
duSP	25.00 (0.29)	39.05 (0.31)	39.19 (0.14)
Overlap	16.19 (0.32)	18.75 (1.63)	31.26 (1.30)

6.1.3 Joint Evaluation

The results for the joint system can be found in Table 5. As we can see, each variable in our evaluation influences the performance of the system. The most influential parameter is the number C of potential situations. As the complexity increases, we find the accuracy of the model decreasing: from 58% in the case C=4 with no new users or tracks to 16% in the case C=12 with cold scenarios. Additionally, we find the expected variation in performance between the cold cases and the warm case of intended use. We observe how the drop in the performance of the auSP and duSP, on new users/tracks, negatively affects the joint system performance.

However, in the harder evaluation case of generating a situational playlists with only 1 guess allowed out of C=12, the proposed system would have been able to include at least a third of the actual listened tracks (31.26%) in those playlists, while pushing them to the user at the exact listened time.

7. CONCLUSION

In this study, we address the problem of the unobserved listening situation which influences the users' preferences. We proposed a two-branch framework to predict when a situation is being experienced based on the device data, while simultaneously autotagging the music tracks with their intended listening situation in a personalized manner. Through the proposed approach, users could access a set of predicted potential situations. These situations are also associated with a set of tracks "likely" to be listened to by the user. This likelihood is estimated using an autotagger trained on predicting the situational use of tracks, given a specific user and his/her listening history. We evaluated each of our system's blocks individually and combined. The evaluation results indicated that the system is capable of learning personalized patterns for users, which can be employed to provide contextual music recommendation.

8. REFERENCES

[1] A. C. North and D. J. Hargreaves, "Situational influences on reported musical preference." *Psychomusicology: A Journal of Research in Music Cognition*, vol. 15, no. 1-2, p. 30, 1996.

[2] K. Ibrahim, E. Epure, G. Peeters, and G. Richard, "Should we consider the users in contextual music auto-tagging models?" in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[3] A. E. Greasley and A. Lamont, "Exploring engagement with music in everyday life using experience sampling methodology," *Musicae Scientiae*, vol. 15, no. 1, pp. 45–71, 2011.

[4] M. Gorgoglione, U. Panniello, and A. Tuzhilin, "The effect of context-aware recommendations on customer purchasing behavior and trust," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 85–92.

[5] C. Hansen, C. Hansen, L. Maystre, R. Mehrotra, B. Brost, F. Tomasi, and M. Lalmas, "Contextual and sequential user embeddings for large-scale music recommendation," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 53–62.

[6] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challenges," *Computer Science Review*, vol. 6, no. 2-3, pp. 89–119, 2012.

[7] Ò. Celma and X. Serra, "Foafing the music: Bridging the semantic gap in music recommendation," *Journal of Web Semantics*, vol. 6, no. 4, pp. 250–256, 2008.

[8] K. Ibrahim, J. Royo-Letelier, E. Epure, G. Peeters, and G. Richard, "Audio-based auto-tagging with contextual tags for music," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[9] M. Schedl, A. Flexer, and J. Urbano, "The neglected user in music information retrieval research," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 523–539, 2013.

[10] F. Korzeniowski, O. Nieto, M. McCallum, M. Won, S. Oramas, and E. Schmidt, "Mood classification using listening data," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR*, 2020.

[11] X. Wang, D. Rosenblum, and Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 99–108.

[12] M. Pichl, E. Zangerle, and G. Specht, "Towards a context-aware music recommendation approach: What is hidden in the playlist name?" in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 1360–1365.

[13] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[14] P. Herrera, Z. Resa, and M. Sordo, "Rocking around the clock eight days a week: an exploration of temporal patterns of music listening," in *Proceedings of the 1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[15] M. Gillhofer and M. Schedl, "Iron maiden while jogging, debussy for dinner?" in *MultiMedia Modeling*, X. He, S. Luo, D. Tao, C. Xu, J. Yang, and M. A. Hasan, Eds. Cham: Springer International Publishing, 2015, pp. 380–391.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.