HIT SONG PREDICTION: LEVERAGING LOW- AND HIGH-LEVEL AUDIO FEATURES

Eva Zangerle, Ramona Huber, Michael Vötter
University of Innsbruck, Austria
firstname.lastname@uibk.ac.at

Yi-Hsuan Yang
Academia Sinica, Taipei, Taiwan
yang@citi.sinica.edu.tw

ABSTRACT

Assessing the potential success of a given song based on its acoustic characteristics is an important task in the music industry. This task has mostly been approached from an internal perspective, utilizing audio descriptors to predict the success of a given song, where either low- or high-level audio features have been utilized separately. In this work, we aim to jointly exploit low- and high-level audio features and model the prediction as a regression task. Particularly, we make use of a wide and deep neural network architecture that allows for jointly exploiting low- and high-level features. Furthermore, we enrich the set of features with information about the release year of tracks. We evaluate our approach based on the Million Song Dataset and characterize a song as a hit if it is contained in the Billboard Hot 100 at any point in time. Our findings suggest that the proposed approach is able to outperform baseline approaches as well as approaches utilizing low- or high-level features individually. Furthermore, we find that incorporating the release year as well as features describing the mood and vocals of a song improve prediction results.

1. INTRODUCTION

The task of predicting hit songs aims to infer the potential (commercial) success of a given song, possibly before the release of the song [18]. This is particularly interesting for the music industry as it allows to find potentially successful songs, promising songwriters and composers, to allocate budget for promotion, and to identify key elements that are pivotal for the success of a song. A natural next step would be the automatic generation of musical pieces which actually exhibit these features that have been shown to be crucial for success. To this end, the hit song prediction task has been tackled from two perspectives [12, 23]: an *internal* perspective, which relies solely on (musical) features extracted from the audio, and an *external* perspective, which models aspects of the musical ecosystem, for instance by incorporating social media or market data.

In this work, we take on the internal perspective, focusing on audio descriptors of a song to predict its success. While this might not capture all the aspects that are relevant for the musical success of songs (e.g., social media trends and events [25], psychological issues [18] or social influence [9, 20]), we believe that it is still an important problem that can be approached on its own and possibly enriched with external information at a later stage.

Approaches focusing on internal factors have mostly modeled hit prediction as a classification or regression problem solved by traditional approaches or, more recently, deep learning [23, 24]. The features used to characterize songs range from low-level Mel-Frequency Cepstral Coefficients (MFCC) [6], melodic features [8], temporal features [10], lyrics features [21] to high-level audio features describing e.g., the danceability of songs [7, 17]. While both of these feature types have been individually shown to contribute to hit prediction, they have yet to be exploited jointly for this task.

Recently, Demetriou *et al.* [5] have investigated the most influential features when it comes to users liking or disliking a song in a user study. They have shown that the most significant features of a song are its ability to evoke emotions, vocals of the singer, beat and rhythm and the lyrics. Along these lines, we are particularly interested in investigating whether these features are also influential in the task of hit song prediction. Interiano *et al.* [12] have shown that audio descriptors relevant for the success of a song change over time and that musical fashion is rather short-lived, rendering it hard to exploit past data to predict future trends. Despite approaches to predict the release year of songs based on acoustic features [2] and the use of temporally weighted regression methods to account for changing features over time [7], this fact has not yet been explicitly explored for hit song prediction. Incorporating release year information into our hit song prediction approach to reflect the dynamics of success on the music market is another distinguishing feature of this work.

Consequently, we shed light on the following two research questions (RQ) in this study:

RQ1: How can we predict hit songs based on acoustic features extracted from the song's audio in a deep learning scenario?

RQ2: Which role do individual features (or groups of features) and the release year of a song play in this task?

To answer these research questions, we model the prediction of hit songs as a regression task. We extract low-

and high-level features from the audio of each track and feed these into a deep neural network architecture, where low-level features are fed into the deep part of the network to distill dense representations thereof, whereas high-level features are fed into the wide part of the network to be utilized directly. This also allows feeding the release year of a song into the network as a high-level feature. Utilizing the dense computed dense representations of low-level features in combination with high-level features, we subsequently compute a regression task to predict a track's peak ranking position.

The contribution of this paper lies in the following aspects: (i) we present a novel regression approach towards hit song prediction using neural networks which combines wide (high-level) and deep (low-level) acoustic features; (ii) we show that mood and vocals (the features identified as being crucial when it comes to liking and disliking a song [5]) are also of high relevance for the hit prediction task; (iii) we show that adding the release year as a high-level feature allows for further improvements, implying that contextualizing the song temporally is important due to the short-lived trends in music [12]; (iv) this is the first work that utilizes solely data from the public domain for this task. For reproducibility and to encourage follow-up research along this line, we also make public the data underlying our experiments [1] .

The remainder of this paper is structured as follows. Section 2 provides information about the dataset underlying our analyses. Section 3 details our proposed approach towards the prediction of hit songs. Section 4 presents the experiments we conducted and the results obtained. Section 6 concludes this paper and discusses future work.

2. DATASET

We base our experiments on the widely used and freely available Million Song Dataset (MSD) [2], which contains one million songs that are representative for western commercial music released between 1922 and 2011. The dataset contains release year information for 515,576 of the MSD songs [2]. As we are interested in the impact of the release year information on hit song prediction quality, we constrain our dataset to those songs that we can obtain the release year information for. In contrast to previous studies on hit song prediction, our dataset fully stems from the public domain. Please refer to Table 1 for an overview of the datasets utilized in existing work and their availability.

To extract low- and high-level audio features for every song, we rely on representative 30 seconds samples for each of the songs in the Million Song Dataset. We make use of the Essentia framework [3] to extract low- and high-level features from the audio (cf. Section 3.1 for details) and dropped all songs in the MSD where we could not determine all those features.

Moreover, our approach requires distinguishing between hits and non-hits of musical success [15]. Along the lines of previous research [13, 21], we define a song as successful if it is featured in the weekly Billboard Hot 100 [2] at least once. Therefore, we crawl the Billboard Hot 100 from the according website for the years 1954 until 2018. To find songs in the Billboard Hot 100 matching the songs contained in the Million Song Dataset, we compare both the artist name and track title for each song pair in the two sets and only consider exact matches as hit songs. After that, we dropped duplicates (determined based on artist name and track title). This provides us with a set of 5,832 hit songs and hence, positive samples for which we extract their highest rank in the charts. For negative samples (and hence, non-hits) sampled from the MSD, it is important to ensure that they are not accidentally hits. Hence, we used to following procedure based on the set of songs for which we have release year information and Essentia features. Firstly, we compute a fuzzy matching ratio [3] between all MSD songs and the set of hit songs by concatenating the artist name and track title with a delimiter and selecting the best matching pairs thereof. Based on this matching procedure, we gather a pool of non-hits where the title fuzzy matching ratio is less or equal than 40. We determined this threshold by preliminary experiments and manually inspecting results. We only consider the title ratio here as it is possible that an artist has multiple further songs, that we nevertheless aim to include in our set of possible non-hits and hence, we do not include artist similarity in this computation. The resulting dataset contains a substantially higher number of non-hits (89,235) than hits (5,832), hence it is highly imbalanced (6.1% positive vs. 93.9% negative instances). To overcome this imbalance, we decided to randomly draw 5,832 samples from the pool of non-hits to get a balanced dataset for our experiments.

Paper	Data	PD	AV
[23, 24]	KKBOX listening data, audio	no	no
[6]	in-house audio database, UK, US, AUS charts	no	no
[8]	in-house audio database, UK charts	no	no
[21]	lyrics features, Billboard charts	no	no
[7, 17]	Echonest features, UK charts	yes	no
[18]	HiFind database of music	no	no
this	MSD dataset, Essentia features	yes	yes

Table 1. Datasets utilized for internal hit song prediction. Notation: PD—dataset stems from public domain, AV—dataset is publicly available.

3. HIT SONG PREDICTION

In this section, we detail our approach on hit-song prediction. We first present the features utilized to characterize songs and then detail the neural network-based approach.

3.1 Song Features

Previous research in the field of hit song prediction has relied on utilizing either low- or high-level features of songs.

[1] https://doi.org/10.5281/zenodo.3258042

[2] https://www.billboard.com/charts/hot-100

[3] The ratio of matching tokens between the two strings is based on Levenshtein distance as implemented by Python's fuzzywuzzy library.

Category	Features
mood	acoustic, aggressive, electronic, happy, party, relaxed, sad [14]; Hu and Downie's 5 clusters of mood [11]
genre	blues, classic, country, disco, hip-hop, jazz, metal, pop, reggae, rock [22]
voice	voice, instrumental, female voice, male voice
rhythm/beat	bpm, beats count, bpm histogram, beats loudness, beats loudness band ratio, onset rate, danceability
chords	chords strength, chords change rate, chords number rate, chords key, chords scale, harmonic pitch class profile, tuning strength and frequency

Table 2. Feature categories and the Essentia features each category contains.

Low-level features allow capturing acoustic descriptors like loudness, dynamics, and spectral shape of a signal, rhythm descriptors or tonal information [19]. In contrast, high-level features are computed from low-level feature models and capture abstract concepts such as mood, genres, vocals or music type [19]. In this work, we aim to combine those two types of features. The intuition here is that while low-level features allow for a detailed description of the acoustic characteristics of a song, high-level features complement this detailed view with abstract concepts such as mood or danceability, resulting in a more holistic description of a song.

Based on the dataset presented in Section 2, we propose to extract low- and high-level features based on a given MP3 file of a song containing a 30 seconds preview. Particularly, we make use of the Essentia toolkit [19], a well-established and widely used extraction library for audio descriptors. For the extraction of low-level features, we rely on Essentia's pre-compiled extractors [4], which provide a variety of spectral, time-domain, rhythm, and tonal descriptors. This provides us with 40 basic features (e.g., MFCCs, dissonance or silence rate), 11 rhythm features (e.g., beats per minute or onset-rate) and 13 tonal features (e.g., key or harmonic pitch class profiles) that serve as low-level input for our task.

For high-level features, we again rely on Essentia and utilize the provided pre-trained high-level classification models [5] to compute high-level features based on the low-level features previously extracted. These features include musical genre, mood, timbre, vocals/voice, or danceability.

In this work, we hypothesize that features identified as salient in users liking/disliking a song [5] are also relevant for the task of hit song prediction. To assess the relative importance of these different features, we rely on the categories of features proposed by Demetriou *et al.* [5] and perform a matching between Demetriou's categories of features and our dataset's features. As our approach is based on internal features only, we are not able to match all of Demetriou's categories (e.g., lyrics). We argue that this is still a valid approach as this work is focused on internal aspects of a song. We hence make use of the following feature categories: mood, genre, voice, rhythm/beat, and chords. The first three contain solely high-level features computed by Essentia, whereas the latter two stem from both Essentia's low- and high-level features. Table 2 shows the assignment of individual low- and high-level features to those categories. As previous research has shown that musical fashion and trends are highly dynamic and short-lived [12], we are also interested in the impact of information about the release year of a song. The idea here is that providing temporal context in terms of the release year of a song can contribute to improved prediction performance as the characterization and embedding of the song is improved. Hence, we extract the release year information for each song from the Million Song Dataset and treat it as a high-level feature.

3.2 Regression Wide and Deep Network

The core idea of our approach is to combine low- and high-level acoustic features to characterize tracks as those two types of features capture different aspects and characteristics of a track (on different levels of abstraction). Given the differences between these two feature types in terms of the amount of features, complexity, and diversity, we aim to reflect this in the architecture of the neural network used for hit prediction. Therefore, we utilize a network architecture inspired by the structural concept of the Wide and Deep network architecture by Cheng *et al.* [4]. While our proposed solution is in fact quite different from the original model [6], we believe that the distinction and notion of deep and wide features describes our scenario well. Hence, we will nevertheless use this notion of wide and deep features and the corresponding network parts.

Figure 1 presents an overview of the proposed network architecture. This architecture allows training a wide linear model alongside a deep neural network while distinguishing two types of features: wide features can be regarded as abstract, high-level features that can directly be used for further computation, whereas deep features in the deep part of the network are used to learn dense, lower-dimensional representations of input features. In our scenario, low-level features can be considered deep features, whereas high-level features are wide features. Based on the wide features and the computed dense representations of the deep features, we aim to perform a regression task for predicting the peak position a song will reach in the charts. This can also be used to distinguish hits and non-hits by using any position larger than 100 as a threshold value. As for the implementation of the deep part of the network, the goal here

[4] http://essentia.upf.edu/documentation/extractors_out_of_box.html, music 1.0 extractor of Essentia v.2.1.-beta2 was used.

[5] http://essentia.upf.edu/documentation/streaming_extractor_music.html

[6] The original wide and deep approach was designed for a recommendation scenario, where the wide part is used to model user-item co-occurrences and the deep part is used to learn low-dimensional latent descriptors of queries and items.

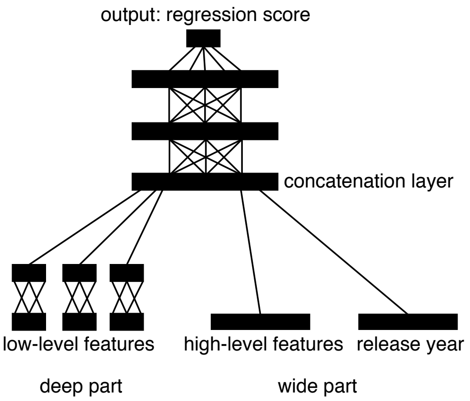

Figure 1. The wide and deep network architecture employed for hit song prediction.

is to use the sparse low-level features as input and to compute meaningful dense representations of audio descriptors to be processed further.

Low-level features comprise a variety of different feature formats: e.g., aggregations of frame-based features across the song or per-frame values for a set of frequency bands—rendering a sparse, complex set of feature representations, where also the computed individual values stem from a broad range. Practically speaking, our approach has to be able to cope with nested input features of varying size, complexity and value ranges. We chose to flatten these input arrays into one-dimensional arrays that can be fed into the deep part of the network.

The purpose of the deep part of the network is to aggregate feature vectors to a one-dimensional representation of the original input features, which are subsequently fed into the regression part of the network. We create multiple groups of low-level features corresponding to the feature categories and their components presented in Table 2. Each group is then fed into a single feature aggregation block (FAB) in the deep part of the network. We chose to model each FAB as a dense layer utilizing a sigmoid activation function as this has been shown to be effective for feature selection in deep neural networks [16]. The input size is chosen to fit the number of values in the feature group and the output size is one. This results in a single value per feature group which is the newly computed higher-level representation of this group. The resulting features computed by the FABs in the deep part of the network are subsequently merged with the features that we feed into the wide part of the network. This concatenation (merge) layer is followed by two dense layers with batch normalization and a ReLU activation function. These two dense layers have the same size as the concatenation layer. To ultimately compute the final result of the regression task, we add another dense layer with output size one and no activation function to ensure that the computed result is in the desired range of possible ranking positions (1–150).

Each high-level feature is represented by classes, where each class is assigned a probability value (range $[0, 1]$). In the special case of two complementing classes such as danceable and not danceable, we chose to only use one of these two probabilities (in the above example, the probability of a song being danceable) to model this feature. We use the resulting values as input for the wide part of the neural network. Feeding categorical values such as the tonal key into the network is realized by previously converting them to a one-hot encoded vector representation. Further, it should be mentioned that we normalized all input (feature) values to the range $[0, 1]$ using a min-max-scaler. Feeding the release year into the neural network is realized by adding another high-level feature (normalized to $[0, 1]$).

4. EXPERIMENTS AND RESULTS

Here, we first present the experimental setup used and secondly, we present and discuss the results obtained.

4.1 Experimental Setup

We base our experiments on the dataset presented in Section 2 and experiment with two different regression tasks: predicting the highest rank of a song in the Billboard Hot 100. For non-hits, we set the highest rank achieved to 150. We chose to use a ranking of 150 to describe non-hits to make the difference between hits and non-hits in terms of ranking more explicit based on preliminary experiments. Due to the high imbalance of hit and non-hit instances in the dataset, we chose to randomly downsample the negative class to achieve balanced classes [7] . Subsequently, we applied five fold cross validation on the remaining instances.

We trained the proposed network architectures with mean squared error (MSE) as loss function and a batch size of 32. The neural network was implemented based on Tensorflow [1], utilizing Keras. As optimizer, we used the adaptive learning rate optimization algorithm, Adam. As for the number of epochs used for training the network, we experimented with values between 10 and 200. All input data is scaled to $[0, 1]$. As we experimented with a wide variety of different setups, training epochs, etc., we utilized a grid search approach to determine the best configuration and present the best obtained results in Section 4.2. Naturally, the underlying network was trained and optimized individually for each input feature set. For the evaluation and comparison of the proposed regression approaches, we use root mean squared error (RMSE) and the mean absolute error (MAE). To also derive a measure of how well these approaches perform when it comes to actually predicting hit songs, we also present the accuracy values for each approach. These were computed based on the results of the regression computation and classifying all tracks with a predicted ranking of less than 100 as hits and a predicted ranking larger than 100 as non-hits. However, we consider this two-class classification an easier task than the regression task based on the actual ranking. Here, we argue that the accuracy evaluation allows us to get an intuition on how

[7] Manual inspection showed that the release year distribution of the test- and training datasets are comparable.

well the regression results may be used to generally distinguish hits from non-hits.

To assess the relative importance of feature classes, we base our evaluation on the following classes, combinations thereof and the combination of individual features stemming from those classes:

- **LL**: basic low-level acoustic features as presented in Section 3.1.
- **LL-filtered**: a subset of the low-level feature set that we have identified as highly relevant in our preliminary feature selection experiments [8] . We argue that pre-selecting a smaller feature set contributes to both runtime and performance (cf. Section 4.2 for results).
- **chords**: chords features as presented in Table 2, extracted from low-level Essentia features.
- **rhythm**: rhythm and beat features as presented in Table 2, extracted from low-level Essentia features.
- **HL**: all high-level acoustic features, information about the release year of the track, including the following sub-categories: **voice**, **mood**, **genre** and **release year** (cf. Table 2 for details on the contained features).

Please note that depending on the feature set utilized, we adapt the way we utilize the neural network accordingly—i.e., for the low-level feature set, we utilize the deep part of the proposed neural network only, whereas, for the high-level feature set, we utilize the wide part of the network only and for any combination of high- and low-level features, we exploit both parts of the full wide and deep network.

We propose to conduct two experiments to answer our research questions: Experiment 1 aims to assess the performance of the proposed wide and deep network architecture. Therefore, we utilize the proposed low-level features in the deep part of the network and the proposed high-level features in the wide part of the network aiming to show that the proposed architecture achieves superior results than (i) a linear regression baseline as well as (ii) utilizing the two parts of the network individually, relying solely on either low- or high-level features. Based on the results of Experiment 1, Experiment 2 aims to investigate the relative importance of individual feature subsets in the wide and deep neural network. To do so, we experiment with different feature sets (low- and high-level) and compare their prediction performance.

As baselines to compare our approach to, we chose to utilize traditional linear regression [9] , which we apply to the same feature sets.

Approach	**RMSE**	**MAE**	**Acc.**
HL (wide)	57.11	48.50	72.08%
LL-filtered+chords+rhythm (deep)	63.94	54.15	65.50%
LL, chords, rhythm (deep)	60.82	52.09	66.94%
HL+LL-filtered+chords+rhythm (wide + deep)	56.05	45.12	74.23%
HL+LL+chords+rhythm (wide + deep)	**55.45**	**43.84**	**75.04%**
HL (baseline)	58.10	50.38	71.01%
LL-filtered+chords+rhythm (baseline)	223.20	57.68	65.97%
LL+chords+rhythm (baseline)	8.54×10^{9}	7.92×10^{6}	68.47%
HL+LL-filtered+chords+rhythm (baseline)	504.90	52.98	72.56%
HL+LL+chords+rhythm (baseline)	6.4110^{9}	5.95×7^{9}	73.91%

Table 3. Results for highest rank prediction on full feature sets. Both the values of RMSE and MAE are the lower the better; the best results are printed in bold font.

4.2 Results and Discussion

In the following, we discuss the findings of the two experiments conducted.

Experiment 1 aimed to investigate the performance of the wide and deep parts of the network individually but also combined in a full wide and deep architecture. Here, we deliberately include all low- and high-level feature sets proposed. Table 3 depicts the results of this experiment. As can be seen, the proposed wide and deep network approach outperforms the baseline approaches across all evaluation measures. This approach achieves the lowest RMSE and MAE values of 55.45 and 43.84 when relying on all low-level features. Using the filtered set of low-level features reaches an RMSE of 56.05 and an MAE of 45.21. When inspecting the results of the network-based approaches that utilize solely either low- or high-level features (which we also consider as representative baseline methods), we observe that utilizing solely high-level features provides us with reasonable results, suggesting that high-level features indeed capture the abstract characteristics of songs well. In contrast, utilizing only low-level features achieves higher RMSE and MAE values. These observations our initial hypothesis as we find that the combination of low- and high-level features is indeed able to substantially outperform approaches utilizing these feature sets individually. The linear regression baseline approaches in the bottom half of the table achieve the best results when utilizing solely high-level features (MAE of 50.38).

When inspecting the accuracy evaluation, we can observe that the highest accuracy value of 75.04% is achieved by the proposed network approach, again utilizing both high- and low-level features. Interestingly, for the linear regression baselines, while RMSE and MAE values are substantially higher than our proposed approach, we can observe that accuracy values are within a reasonable margin, albeit still lower than the proposed wide and deep approach. We lead this discrepancy back to the fact that we assign non-hits a rank of 150. The observed error measures suggest that the predicted ranks computed by linear regression are very high, leading to such high error margins. This is particularly the case when utilizing the full set of low-level features, holding a substantially higher set of features and hence, posing a more complex regression task. However, given the reasonable accuracy results and

[8] Feature set comprises: dissonance, spectral features (centroid, spread, skewness, kurtosis, flatness db, flux, rolloff, decrease, energy), low energy ratio, avg. loudness, barkbands, erbbands, melbands, MFCCs and HFCs.

[9] We experimented with a number of linear regression algorithms (e.g., ridge, lasso or elastic net regularization), where linear regression obtained the best results.

the fact that the regression model indeed seems to capture the distinction between hit- and non-hit songs well (with a wide margin between predicted rankings for hits and non-hits) and hence, can be considered a reasonable baseline. To conclude and to answer RQ1, our experiments show the proposed wide and deep neural network-based approach combining low- and high-level features is a suitable approach towards hit song prediction.

Experiment 2 aimed to analyze the relative importance of individual features and classes thereof. Therefore, we evaluated different feature sets in the proposed wide and deep network. As the LL basic features have shown to outperform the filtered set of basic low-level features, we restrain the results presented here to the full low-level feature set. Please note that due to space constraints, we only list the best performing and informative configurations and their obtained results.

For low-level features (including chords and rhythm features), we hardly find differences in their performance (across all combinations with high-level features). Differences in RMSE and MAE between different feature variations are very subtle and do not show a clear pattern regarding best performing features. We conclude that neither chords nor rhythm features are particularly pivotal for hit song prediction. Hence, in the following, we restrain the presented results to the full set of low-level features (LL-filtered, chords, rhythm). Table 4 depicts the results of these analyses.

For high-level features, we can observe that year information profoundly contributes to the prediction performance, improving every experiment by 12–13%, when added to set of high-level features. This confirms our hypothesis that due to short-lived fashion and trends in the music industry, embedding songs in their temporal context by adding release year information allows modeling these dynamics efficiently for hit song prediction. Furthermore, we can observe that—along the lines of Demetriou *et al.* [5]—voice, mood, and genre features are also important for this task. Our experiments show that the combination of high-level features improves RMSE and MAE; the best results are obtained when utilizing low-level, rhythm, and chords features in combination with release year, voice, mood and genre features (hence, the full feature set). Inspecting the performance of single HL features (such as e.g., mood) in combination with low-level features shows that year has the highest impact on the evaluation measures, with genre, mood and voice leading to higher error measures. Combining those high-level features, however, allows to substantially increase performance in all evaluated measures. While the differences between these different feature sets are partly subtle, the patterns detected are stable across all our experiments. To answer RQ2, we find that the release year information is the most important high-level feature. Our experiments also show that voice and mood descriptors contribute to the hit prediction task, which is in line with previous findings regarding salient features in regards to whether people like or dislike a song.

Features LL	**Features HL**	**RMSE**	**MAE**	**Acc.**
LL, rhythm, chords	year, voice, mood, genre	**55.45**	**43.84**	**75.04%**
LL, rhythm, chords	year, genre	55.93	45.80	73.84%
LL, rhythm, chords	year, mood	57.12	45.66	73.55%
LL, rhythm, chords	year, voice	56.63	46.04	72.04%
LL, rhythm, chords	genre	64.14	52.84	65.11%
LL, rhythm, chords	mood	61.77	52.82	67.92%
LL, rhythm, chords	voice	61.18	52.50	68.00%
LL, rhythm, chords	year	57.51	46.35	72.29%
LL, rhythm, chords	year, mood, voice	56.22	45.53	74.46%
LL, rhythm, chords	year, genre, mood	57.35	45.38	73.63%
LL, rhythm, chords	year, genre, voice	56.06	45.66	73.60%

Table 4. Results for highest rank prediction on feature sets (best results are printed in bold font).

5. LIMITATIONS

We acknowledge that our dataset and our definition of a successful song are biased towards western, commercial music. While we believe that this approach is legitimate, it remains to be shown that our approach can be extended to other types of music and possibly other characterizations of success. However, we believe that due to using audio features, the approach taken is generalizable. Another limitation is the prevalent problem of class imbalance among hits and non-hits as the current setting does not reflect the real distribution of classes. We aim to experiment with unbalanced distributions between hits and non-hits as part of our future work to perform the evaluation in scenario that captures the real-world distribution better. Furthermore, our approach takes an internal perspective based on the audio signal to characterize the track and to predict its success. Here, we have to acknowledge that this model naturally does not include any external factors such as information about the artist (e.g., whether he/she has been on the charts before), marketing strategies of music labels or the relation with special events (e.g., songs being played at Super Bowl).

6. CONCLUSION

In this paper, we have presented a novel approach for the task of hit song prediction. Particularly, we propose to combine low- and high-level audio features of songs in a deep neural network that distinguishes low- and high-level features to account for their particularities. Our experiments on the Million Song Dataset suggest that the combination of these two types of features in the proposed network architecture can indeed improve the prediction performance. Furthermore, we find that incorporating the release year of songs into the wide part of the network allows for temporally contextualizing songs and hence, reflecting musical trends and fashions. In addition, we can show that mood and voice are salient features for this task. Future work includes experimenting with more complex network architectures to allow for improved feature selection and the computation of latent features within the network as well as analyzing and utilizing those features that distinguish hits from non-hits.

7. REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Proc. USENIX Symposium on Operating Systems Design and Implementation*, volume 16, pages 265–283, 2016.

[2] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proc. International Society for Music Information Retrieval Conference*, 2011.

[3] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Perfecto Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, and Xavier Serra. Essentia: An audio analysis library for music information retrieval. In *Proc. International Society for Music Information Retrieval Conference*, 2013.

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proc. ACM Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[5] Andrew Demetriou, Andreas Jansson, Aparna Kumar, and Rachel Bittner. Vocals in music matter: The relevance of vocals in the minds of listeners. In *Proc. International Society for Music Information Retrieval Conference*, 2018.

[6] Ruth Dhanaraj and Beth Logan. Automatic prediction of hit songs. In *Proc. International Society for Music Information Retrieval Conference*, 2005.

[7] Jianyu Fan and Michael Casey. Study of chinese and uk hit songs prediction. In *Proc. International Symposium on Computer Music Multidisciplinary Research*, pages 640–652, 2013.

[8] Klaus Frieler, K Jakubowski, and Daniel Müllensiefen. Is it the song and not the singer? Hit song prediction using structural features of melodies. *Yearbook of Music Psychology*, pages 41–54, 2015.

[9] Dorien Herremans and Tom Bergmans. Hit song prediction based on early adopter data and audio features. In *Late Breaking Demo at International Society for Music Information Retrieval Conference*, 2017.

[10] Dorien Herremans, David Martens, and Kenneth Sörensen. Dance hit song prediction. *Journal of New Music Research*, 43(3):291–302, 2014.

[11] Xiao Hu and J Stephen Downie. Exploring mood metadata: Relationships with genre, artist and usage metadata. In *Proc. International Society for Music Information Retrieval Conference*, pages 67–72, 2007.

[12] Myra Interiano, Kamyar Kazemi, Lijia Wang, Jienian Yang, Zhaoxia Yu, and Natalia L. Komarova. Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society Open Science*, 5(5), 2018.

[13] Yekyung Kim, Bongwon Suh, and Kyogu Lee. #nowplaying the future billboard: Mining music listening behaviors of twitter users for hit song prediction. In *Proc. International Workshop on Social Media Retrieval and Analysis*, pages 51–56. ACM, 2014.

[14] Cyril Laurier, Owen Meyers, Joan Serra, Martin Blech, and Perfecto Herrera. Music mood annotator design and integration. In *Proc. IEEE Workshop on Content-Based Multimedia Indexing*, pages 156–161, 2009.

[15] J. Lee and J. Lee. Music popularity: Metrics, characteristics, and audio-based prediction. *IEEE Transactions on Multimedia*, 20(11):3173–3182, 2018.

[16] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[17] Yizhao Ni, Raul Santos-Rodriguez, Matt Mcvicar, and Tijl De Bie. Hit song science once again a science? In *Proc. International Workshop on Machine Learning and Music*, pages 2–3, 2011.

[18] François Pachet. Hit Song Science. In *Music Data Mining*, pages 305–326. Chapman & Hall/CRC Press Boca Raton, FL, 1st edition, 2012.

[19] Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, and Xavier Serra. AcousticBrainz: a community platform for gathering music information obtained from audio. In *Proc. International Society for Music Information Retrieval Conference*, 2015.

[20] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006.

[21] Abhishek Singhi and Daniel G Brown. Hit song detection using lyric features alone. *Proc. International Society for Music Information Retrieval Conference*, 2014.

[22] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[23] Li-Chia Yang, Szu-Yu Chou, Jen-Yu Liu, Yi-Hsuan Yang, and Yi-An Chen. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. In *Proc. IEEE International Conference Acoustics, Speech and Signal Processing*, pages 621–625, 2017.

[24] Lang-Chi Yu, Yi-Hsuan Yang, Yun-Ning Hung, and Yi-An Chen. Hit song prediction for pop music by siamese cnn with ranking loss. *arXiv preprint arXiv:1710.10814*, 2017.

[25] Eva Zangerle, Martin Pichl, Benedikt Hupfauf, and Günther Specht. Can microblogs predict music charts? an analysis of the relationship between #nowplaying tweets and music charts. In *Proc. International Society for Music Information Retrieval Conference*, pages 365–371, 2016.