MIRA: Proactive Music Video Caching using ConvNet-based Classification and Multivariate Popularity Prediction

Christian Koch, Stefan Werner, Amr Rizk and Ralf Steinmetz
Multimedia Communications Lab, Technische Universität Darmstadt, Germany
Email: {Christian.Koch | Amr.Rizk | Ralf.Steinmetz}@kom.tu-darmstadt.de, stefan.werner@stud.tu-darmstadt.de

***Abstract*—Music belongs to one of the most popular content categories overall, and it is nowadays mainly consumed using online streaming services. With YouTube being the largest source of traffic in most networks about half of all YouTube requests address music videos. To cope with the continuously growing demand for content and thus increasing network traffic, YouTube operates its own CDN, a globally distributed network of caches. This allows serving content from locations close to the users, which circumvents potential network bottlenecks and increases the user-perceived QoE due to reduced latency. Recently, proactive caching and prefetching has shown superior performance results compared with traditional reactive caching schemes such as LRU and LFU. Due to the substantial footprint of music videos on today's Internet, we propose a novel proactive caching strategy specifically for music videos. This strategy incorporates two key observations: i) Music genre and mood popularity varies over the course of the day and ii) A video's past views are predictive for its future popularity development. For the classification task, we use a Convolutional Neural Network while investigating several predictive models for the popularity estimation. The proposed caching system can increase the cache hit rate up to 4.5% which is substantial for caching systems.**

I. Introduction

Nowadays, users access music predominantly using web services like Spotify and Google Music. Already 71% of the Internet users utilized at least one music streaming service in 2016 [1]. Even on YouTube, being one of the largest traffic sources in most networks [2], [3], music videos constitute the most popular content category with 82% of its users watching music videos. Furthermore, music is responsible for the largest share of YouTube requests ranging between 37% and 42.16% [4], [5]. This share is predicted to grow further together with the overall amount of Internet traffic [6]. To cope with this high traffic demand, YouTube operates a globally distributed cache network, i.e. the Google Global Cache (GGC), to serve content from locations close to the users. Thereby, potential transmission bottlenecks and long video startup stalling are prevented. Furthermore, the user satisfaction is likely to be higher since low delays correlate with higher QoE [7] and higher user engagement [8].

Increasing caching efficiency has gained a lot of research attention leading to sophisticated reactive caching approaches. Though, proactive caching has shown to outperform reactive caching [4], [9], [10]. In this paper, we focus on proactive caching of the largest category of YouTube traffic, which are music videos. We aim to show first results of a novel proactive caching system using a convolution neural network (ConvNet or CNN) to determine mood and genre of audio tracks of music videos. While mood and genre can be determined for, e.g., movies as well, for music, these features are the key characteristics and can be accurately determined using the audio signal only. Next, we leverage the varying diurnal genre and mood popularity pattern to proactively cache popular content, which protects it from being evicted. In this paper, we provide three contributions. First, we discuss the design of our proposed novel proactive caching system. This covers music feature extraction, classification, and diurnal pattern analysis. Second, we investigate the popularity prediction of music videos, e.g., using univariate and multivariate prediction models. Here we estimate the future popularity of videos based on the number of views received in the past. Third, we evaluate our system in a simulative environment using a real-world trace of YouTube requests collected within a large European ISP and covering two weeks.

The remainder of this paper is structured as follows. Section II discusses the related work. In Section III, we introduce our system design and evaluate it in Section IV. Section V concludes the paper and explains directions for future work.

II. Related Work

The related work comprises three parts. First, we discuss relevant works in the area of music classification. Second, we briefly discuss related proactive caching approaches, and third, we present popularity prediction approaches.

A. Music Classification

Music classification aims to assign a label from a pre-specified label set to a music track referring to, e.g., the track's genre or mood. For many tracks, such labels can be retrieved from online music databases, e.g., from last.fm[1]. However, for very recent and less popular tracks, this information is unlikely to be available as these databases rely on tags assigned to the

[1]https://www.last.fm/en/api [Accessed: July 26, 2018]

tracks by users. Hence, automated classification is preferable as it can also assign labels to new tracks. To this end, we use a set of music tracks and their known labels to train a classifier that can estimate the respective labels. Most existing research in the area of music information retrieval aims at genre or mood classification. The features on which the classifier is trained are traditionally characteristic audio features such as the Mel Frequency Cepstral Coefficients (MFCC). Wang et al. [11] evaluate the impact of different acoustic feature sets regarding classification accuracy. Therefore, they extracted features from, both, the signal processing and musical dimension using MIRtoolbox [12]. Next, they train Support Vector Machines (SVMs) on different feature subsets. A combination of, e.g., Rhythm, Timbre, and Tonality achieves 79.5% accuracy using a labeled dataset of 1,000 tracks [13]. Note that the classification accuracy strongly depends on the experts who annotated the training dataset and its size. This is an explanation of the spectrum for mood classification accuracy ranging between 25% [14] and 90.44% [15]. Recent works use neural networks to derive music features and show a more stable and higher performance compared to traditional approaches. This way, Costa et al. [16] achieve 87.4% accuracy. Choi et al. [17] present a transfer learning approach for music classification using ConvNets. Their neural networks take the music's Mel spectrograms as an input, which efficiently approximates human auditory perception. The training uses several publicly available datasets and aims at music tagging. However, the resulting 160-dimensional feature vector of their network is useful for any music information retrieval task. The authors use this feature vector as an input for an SVM trained to classify a music track's genre and, thereby, achieve 89.8% accuracy. Hence, we conduct that neural networks are in general more powerful to retrieve expressive music features.

B. Proactive Caching

Proactive caching has demonstrated to outperform reactive policies like LRU. Hasslinger et al. [10] propose score-gated LRU. Here, each item is associated with a score based on its popularity. If a recently requested item has a lower popularity score than the items in the cache, it is not cached. Thereby, the cached items are not evicted as long as the content popularity distribution does not change. By implementing a variant of score-gated LRU, the authors achieve about 10% hit rate increase compared with LRU. This approach is proactive in its eviction policy but still reactive by measuring the content's past popularity. Gouta et al. [9] present a proactive caching scheme based on collaborative filtering. The idea behind this is that users who request similar items are predictive for each other and content consumed by similar users is likely to be requested by each of them. Hence, this approach needs a user-item matrix containing past user requests to determine which content is likely to be requested in the near future. However, in their experiments, they exclude music videos as they show a different and more persistent popularity pattern compared with other video categories and, hence, are likely to decrease the performance of their proposed caching system. We argue that music videos are the largest category of YouTube videos measured regarding requests and should not be ignored. Koch et al. [18] demonstrate that proactive caching is superior to LRU. However, we do not address music content specifically and do not incorporate machine learning.

C. Popularity Prediction

In the following, we introduce and discuss popularity prediction approaches for YouTube videos. Szabo et al. [19] present an approach that predicts a content's long-term popularity based on early view count measurements from YouTube and Digg. In the case of YouTube, they collect view count time series of about 7k videos. They found that early and late popularity strongly correlate. Hence, the authors propose a univariate linear model for popularity prediction. In their evaluation, the error decreased with the video's age from 20% after five days to 10% after ten days. Pinto et al. [20] present a multivariate linear model that takes regular popularity measurements as input. Next, they evaluate the model using between 30 and 100 popularity measurements of about 20k YouTube videos. Their model outperforms the univariate linear model with up to 20% decrease in mean relative squared error (mRSE). However, if almost the entire video popularity history is available, it is likely to have predictions with low error but at a late point in the video's lifetime.

III. System Design

We design a novel proactive caching system, denoted as **MIRA** (**M**usic Video **I**nformation **R**etrieval for c**A**ching). It is a domain-specific system leveraging music content features and popularity characteristics and, thereby, differs from related work, as we will detail in this section. Figure 1 gives an overview of the system components. On the left side, we see the *Popularity Predictor* module, which estimates the video popularity based on view count time series of the past. As introduced in Section II-C, univariate and multivariate models are suitable for this task. Although a multivariate model tends to be more flexible and, hence, shows better results in the related work, we assess both models since the related work uses a different dataset for evaluation and, hence, comparability is limited. Note that MIRA can use either a univariate or a multivariate model. The second module that provides MIRA with information is the *Music Feature Extractor*. It uses a pre-trained ConvNet that extracts music features form

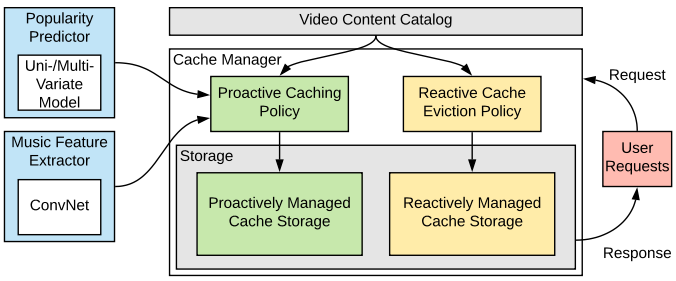


Fig. 1: MIRA system architecture

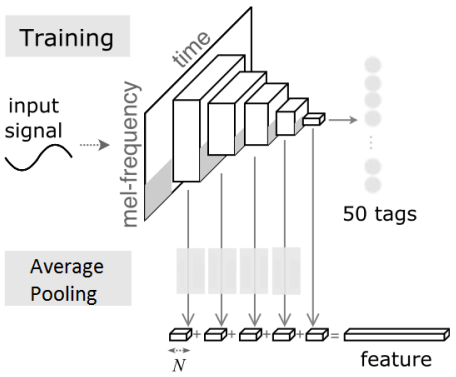


Fig. 2: ConvNet architecture by Choi et al. [17]

the video's audio signal. Both modules serve as an input for the *Proactive Caching Policy*. Here, the videos predicted to be most popular within the next proactive caching interval I_{t+1} are determined. While the video view count is computed on a daily basis, the genre and mood distribution vary during the hours of the day [4]. The caching interval is a tunable system parameter that we set to one hour in the course of this paper. The *Cache Manager* conducts the video selection using a proactive caching policy. The selected videos are stored in the proactively managed cache storage S_p.

Overall, we divide the cache of size S into two parts: the proactively managed storage S_p and the reactively managed cache storage S_r. We denote the storage proportions as r_p and r_r. Videos that should be stored into S_p but are already present in S_r are moved to S_p to avoid redundancy. Keeping r_p and r_r variable, allows for flexible evaluation of the impact of proactive caching. We will examine the impact of iteratively increasing r_p. Note that $r_p = 0\%$ results in pure reactive caching. Since LRU is the most popular reactive caching policy we choose it to allow for better comparison with the related work. If a user requests a video segment, the *Cache Manager* checks if it is contained, first, in S_p and, second, in S_r. In case the segment is contained in one of the storage areas, it is delivered as a reply to the requesting client. A cache miss causes a content request to the source, which holds the entire content catalog. The retrieved video segment is stored in S_r and is delivered to the requesting client. If S_r is full, the reactive eviction policy determines which video segment(s) to delete so that enough free space becomes available to allow storing the requested video segment.

A. Music Feature Extractor

MIRA uses mood and genre information provided by the *Music Feature Extractor* as an input. In the following, we discuss the design of the required music genre and mood classifier. This can be formulated as a supervised machine learning task where we give a pre-labeled dataset as an input to allow learning a generalized model. Such a model is able to determine the respective class for previously unseen audio tracks. To train the required two classifiers, we need to provide a labeled training set of <**x**, y> pairs where **x** is a feature vector and y is either the mood or the genre label. As discussed in the related work, ConvNets are superior in music feature extraction and, therefore, ideal for training our classifiers. To this end, we decide to rely on the publicly available[2] work of Choi et al. [17]. Figure 2 depicts their proposed ConvNet architecture. Here, the input is a 30s audio track starting from playback time 30s to discard the often not representative intro. The resulting feature vector of this ConvNet is taken as an input for an SVM, which showed the highest classification accuracy for both cases: mood and genre classification (ref. Section II-A). Note that using an SVM in combination with a DNN usually improves the performance compared with a classification that directly uses a DNN, e.g., by using a softmax layer [21]. We train the SVM using 10-fold cross-validation and a grid search for hyperparameter optimization. For the training of the music genre classification model, we use 9,029 annotated tracks annotated with 14 different genres, e.g., pop, rock, hiphop, and electronic similar to a related work [4]. For the music emotion classification, we use 966 annotated tracks together with their labels: angry, happy, sad, and relaxed. Thereby, we cover all four quadrants of the common mood model from Thayer [22]. Interestingly, we found that only about 1% of the music videos requested belong to angry music. For the rest of each emotion category, we observe roughly one-third of all requests. Analyzing the diurnal mood and genre distribution, we found that there is almost no change in emotion distribution. However, genres show a dynamic distribution over the hours of the day. Next, we show the evaluation of our classification accuracy in Table I. The performance of the mood classifier is substantially higher than the performance of a random classifier where the accuracy is $\frac{1}{\#\text{classes}}$ and shows even better results than the ones achieved by the related work in the area of music emotion classification [4], [23]–[25] and similar results in music genre classification [26], [27].

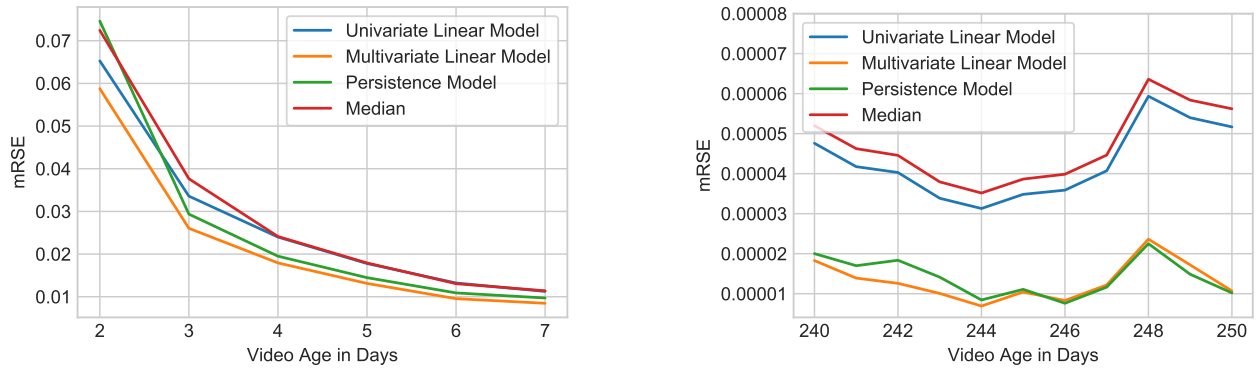
B. Popularity Predictor

The video popularity is commonly measured by the view count increase over a specific time interval, e.g., per day in the case of YouTube. We choose to incorporate four models in our system to allow for comparison. The first model is a multivariate linear model (ML) as it was shown to achieve diminishing relative errors when predicting the view counts of the near future [19], [20]. Additionally, it shows low errors when used for predicting view counts for music videos compared to other YouTube categories [20]. The second model is a univariate linear model (UL). Both, the ML and UL model predict the cumulative view count of a video for a given

[2]https://goo.gl/FCYtsJ [Accessed: July 26, 2018]

TABLE I: Parameters and accuracy for classifiers

Classification	Kernel	C	γ	Accuracy
Genre	RBF	$2^{1.75}$	$2^{0.25}$	57.34%
Emotion	RBF	$2^{1.5}$	2^{-1}	69.67%



(a) Videos younger than eight days

(b) Videos aging between 240 and 250 days

Fig. 3: Prediction mean relative squared error (mRSE) for videos of different age

reference date t_r and prediction target date t_p denoted in days since the video upload. In our scenario, we predict the next day's view count, i.e., $t_p = t_r + 1$. Since we want to predict the view count increase of the target day and not the cumulative view count, we compute the view count of a video v at t as follows. Let $\hat{y}_i(v)$ be the predicted view count of a video v on the i-th day since its upload. For a prediction on $i = t_p$ it follows: $\hat{y}_{t_p}(v) = \hat{N}(v, t_r, t_p) - N(v, t_r)$ where $\hat{y}_{t_p}(v)$ denotes the predicted view count for the date t_p. $\hat{N}(v, t_r, t_p)$ is the predicted cumulative view count from either the UL or ML model and $N(v, t_r)$ the cumulative view count at t_r.

The third model we use is a persistence model. It takes the view count from the last day as the prediction, i.e., $\hat{y}_i(v) = y_{i-1}(v)$. The median of the daily video view counts since its upload is used as the fourth model, i.e., $\hat{y}_i(v) = \text{median}((y_1(v), y_2(v), ..., y_i(v))$.

Before each proactive caching, we apply these models to predict the view count for each video in the content catalog for the upcoming time interval till the next proactive caching is triggered. In a last step, we rank the videos according to the view count prediction in descending order. This list determines the content used for proactive caching. Beginning with the first list entry, we load the specified content from the origin, if the video is not already present in S_r. This is repeated until the proactive cache share, i.e., S_p is filled. Videos from the last period which are ranked too low in the prediction list, i.e., are not considered anymore for proactive caching are deleted from the cache. This procedure is repeated each hour.

C. Proactive Caching Policies

Proactive caching prefetches content speculatively to serve future requests. Hence, an estimate of the future content popularity is required to determine which content to prefetch. To this end, we present two proactive caching policies:

1) Caching by Predicted Popularity: Content that is being predicted to be most popular in the next proactive caching interval, i.e., the time interval defining when the cache is filled proactively regularly, is placed at the cache. We take the previous video view counts as inputs for the prediction models considered, i.e., the UL and ML model (ref. Section III-B).

2) Caching by Music Category Distribution: This is an extension to the policy presented before. Instead of caching the videos which are predicted to become most popular, we consider a music-specific category, i.e., genre or mood. In case of the genre, the ratios of music that originates from each genre category are determined for the next hour of the day. Next, the proactive cache space is divided proportionally to these ratios and the content that is predicted to be most popular per category is stored into these divisions. This procedure is done regularly, e.g., once per hour. Thereby, we consider the dynamic and changing request behavior of users which might be more likely to listen to, e.g., pop during the day and jazz in the evening as reported by Gillhofer and Scheld [14].

D. Overhead

Creating the information basis needed for MIRA requires to derive the Mel spectrogram and process it by the ConvNet introduced in Section III-A. Deriving the Mel spectrogram is computationally inexpensive as is the utilization of the ConvNet. Note that neural networks are computationally expensive in training but fast when using them for classification. We assume that this information is computed by the content owner, e.g., YouTube anyway to use it as an input for their recommender systems. MIRA's UL and ML view count models have a linear complexity and only need to be recomputed on a daily basis because the YouTube view count is available on a daily basis as well. When used by a CDN cache that serves a specific region, MIRA's performance can be further increased by using the local view counts instead of the global ones. In this scenario, the CDN needs either to request the content features from the source or compute them by its own. This needs to be done before the next proactive caching interval starts and only for new content, which keeps the computation and communication effort limited.

IV. Evaluation

To evaluate MIRA's performance with varying parameters and policies, we designed a discrete event simulation that is

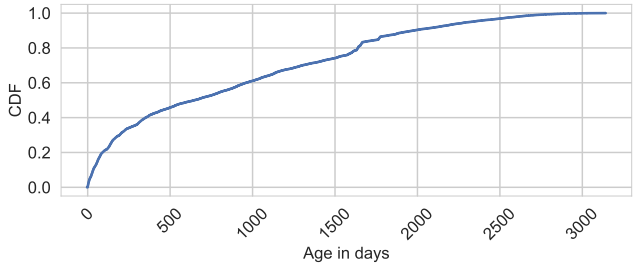


Fig. 4: CDF of the video age per request

based on Simpy [28]. Thereby, we simulate a CDN cache using real YouTube traces as an input. As we need the daily view counts for each video, we crawled this information and enhanced the trace data with it. Note, that the daily view counts are not available for every video since the uploader can decide to make this information public or not. This leaves us with 1,222,198 requests from 192,508 different users towards 16,022 music videos. The simulation has three main components: i) the clients, ii) the cache server(s), and iii) the content origin. The number of cache servers is a configurable parameter. In our simulations, a number of clients is assigned uniformly at random to each cache to reflect the affinity between clients and cache servers based on geographical closeness. The clients request video segments according to the trace. We consider video segments and not entire videos because most videos are only watched partially [29]. If multiple clients request the same video segment from a cache within a short time interval, the cache fetches the video segment just once and delivers it to all requesting clients.

A. Popularity Prediction

We use four models to predict the video view count of the next day: i) univariate linear model (UL), ii) multivariate linear model (ML), iii) persistence model, and iv) the median of the past daily views. To this end, we retrieved the daily view counts from YouTube using the ytcrawl web crawler [30]. The daily view counts are not available for all videos since the uploaded can decide to make this information available or not. This leaves us with a dataset of 16,022 videos for which we crawled the daily view counts from the day of the video's upload till the last video request in our dataset.

The UL and ML models are evaluated by a 10-fold cross-validation. The mean relative squared error (mRSE) is used as a metric to assess their performance, as it is robust against view count values of strongly varying size, which are present for videos of different age and popularity. Figure 3a and 3b depict the results. We intentionally left out the first day, which shows an mRSE of 0.2, in Figure 3a for the sake of figure clarity. We see that the mRSE decays quickly with increasing video age. Hence, the fresher a video is, the more error-prone are the UL and ML models. The persistence model and the median show even higher errors as the video popularity is likely to vary amongst the first days strongly. We empirically observe that the mRSE diminishes after seven days. Hence, we can predict the view count very accurately for videos older than seven days. This can be explained by, both, fewer fluctuations in the view counts and more data at hand that can serve as an input to the UL and ML model leading to a form of a central limit behavior. The ML model shows the lowest mRSE for videos of age below 246 days. For older videos, the persistence model shows the lowest mRSE, as changes in view counts are diminishing with increasing video age. However, both models' mRSE curves are very similar. Note that for music videos, an age of 200 days is not a long time, as they tend to be popular over many weeks, as shown by Figure 4. Here, we see that 50% of the music video requests address videos older than 500 days. Concluding, we predict the view count increase by a hybrid approach using the ML model for views younger than a 246 days cutoff and the persistence model for older videos.

B. Proactive Caching

For the caching simulation, we use the request trace as an input and measure the cache performance regarding cache hit rate (CHR). The CHR is the ratio of video segments delivered from the cache to the total number of requests N, i.e., $\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{h_i}$ with h_i evaluates to `true` if request i is a hit and `false` otherwise. This metric correlates with the content source offload as a higher CHR implies less requests being forwarded to the content origin. Furthermore, video segments delivered from a nearby cache can increase the user-perceived QoE because of a lower transmission delay and circumvention of potential bottlenecks. The cache server stores video segments. However, for the sake of simplicity, we denote the cache size in the number of videos between 100 and 10,000 videos, assuming a video to consist on average of 36 segments of 5 seconds length [5].

1) Pure Proactive Caching: In this section, we compare the two reactive policies LRU and Random Replacement caching (RR) with the proactive policies proposed in this paper. In this experiment, the entire cache is managed proactively or reactively. Figure 5 depicts the results. The dashed blue line indicates the results for the optimal prediction, i.e., storing the most popular content using information from the future. We can see that LRU shows the highest CHR for any cache size considered followed by RR, Genre, Mood, and Popularity. We explain the low performance of completely proactive manage caches by the inability to cache new content that is uploaded within a proactive cache time interval.

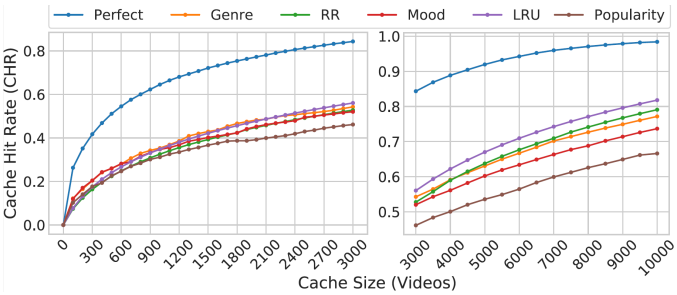


Fig. 5: Comparison with a perfect prediction of view counts

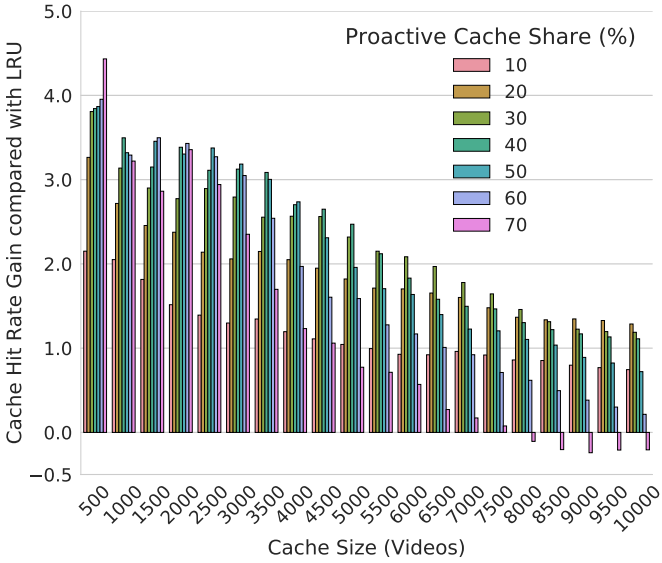


Fig. 6: CHR of the Genre policy compared with LRU

2) Proactive Caching Combined with LRU: In the next experiments, we vary the share of proactively managed cache space between 0% and 100%. The remainder of the cache is managed by LRU. We only consider the genre policy because other proactive caching policies already showed lower performance. Figure 6 depicts the CHR increase compared with LRU of different combinations of reactive and proactive cache storage distribution for different cache sizes. For small cache sizes, up to 2,000 videos, we can see that proactive caching results in the highest CHR increase. For the smallest cache size, i.e., 500 videos, 70% proactive share results in the highest CHR. We intentionally left out higher proactive cache shares for the sake of figure clarity. The optimal proactive share decreases with the cache size, shown in Table II. We conclude that LRU with about 50% proactively managed cache storage achieves the highest CHR for cache sizes $\leq$4,000 videos. For larger cache sizes, 30% is optimal for cache sizes until 8,000 videos and 20% for sizes till 10,000 videos. Hence, proactive caching can increase the cache performance, especially for small cache sizes while larger caches can benefit as well. We also investigated popularity and mood policy instead of the genre policy but could not achieve better results than shown by the presented experiments. Additionally, we evaluated a cache topology of five caches with one-fifth of the size of the large cache. Also in this scenario, LRU with genre policy outperforms the other policies. For the sake of shortness, we refrain from detailing these results in this paper.

TABLE II: Optimal proactive cache share per size

Size(k)	1	2	3	4	5	6	7	8	9	10
%	40	60	50	50	40	30	30	30	20	20

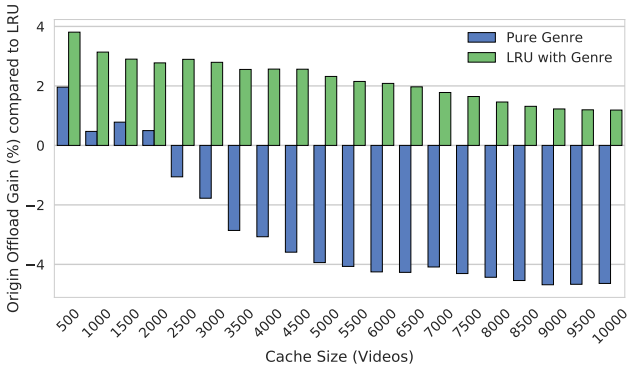


Fig. 7: Origin offload compared to LRU

C. Origin Offload

We demonstrated that proactive caching increases the CHR, especially for small cache sizes. When evaluating classical reactive caching policies, the CHR correlates with the origin offload. However, for proactive caching this correlation is assumed to be weaker as the proactive content fetching for every proactive caching time interval introduces an additional overhead. Figure 7 depicts the origin load reduction (gain) compared with LRU when using i) the pure proactive genre policy and ii) LRU combined with genre. In the latter case, we always chose the best proactive cache size, which we detailed in the previous section. We can see that the origin offload is severely higher than for pure LRU or pure proactive caching and the overhead introduced by proactive caching does not negatively impact the origin offload.

V. Conclusion and Future Work

This paper presents a proactive caching system for music video content, which is one of the largest sources of traffic in today's Internet. Therefore, we use a Convolutional Neural Network (ConvNet) to derive music features to train a genre and a mood classifier. This classification is used as an input for two proactive caching policies used to determine content that is likely to be popular within the next time. We evaluated the performance of three proactive caching policies: Genre, Mood, and Popularity. The genre policy showed the highest gain measured by the cache hit rate (CHR) with 4.5% compared to LRU. Furthermore, we evaluated a set of predictive models used for video popularity prediction and designed a hybrid model which performs best for videos of any age. We demonstrated that proactive caching can increase the CHR, especially for small cache sizes. Furthermore, we evaluate the origin offload, which is 1.5%–4% lower compared to LRU. Hence, we conclude that MIRA is showing promising results and is worth investigating further. We plan to extend our work by formulating proactive caching as a reinforcement learning task. Thereby, we leave the policy design to a neural network.

Acknowledgment

This work has been funded in parts by the DFG as part of the CRC 1053 MAKI.

REFERENCES

[1] I. Connect, "Music Consumer Insight Report," 2016.

[2] A. Gember, A. Anand, and A. Akella, "A Comparative Study of Handheld and Non-handheld Traffic in Campus Wi-Fi Networks," in *Passive and Active Measurement*. Springer, 2011, pp. 173–183.

[3] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update," Cisco, Tech. Rep., 2017.

[4] C. Koch, G. Krupii, and D. Hausheer, "Proactive Caching of Music Videos based on Audio Features, Mood, and Genre," in *ACM Multimedia Systems (MMSys)*, 2017, pp. 100–111.

[5] J. Li, A. Aurelius, M. Du, H. Wang, A. Arvidsson, and M. Kihl, "YouTube Traffic Content Analysis in the Perspective of Clip Category and Duration," in *IEEE International Conference on the Network of the Future*, 2013, pp. 1–5.

[6] "Ericsson Mobility Report," 2017.

[7] D. Stohr, A. Frömmgen, J. Fornoff, M. Zink, A. Buchmann, and W. Effelsberg, "QoE Analysis of DASH Cross-Layer Dependencies by Extensive Network Emulation," in *ACM Workshop on QoE-based Analysis and Management of Data Communication Networks*, 2016, pp. 25–30.

[8] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the Impact of Video Quality on User Engagement," in *ACM SIGCOMM Computer Communication Review*, Vol. 41, No. 4, 2011, pp. 362–373.

[9] A. Gouta, D. Hausheer, A.-M. Kermarrec, C. Koch, Y. Lelouedec, and J. Rückert, "CPSys: A System for Mobile Video Prefetching," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2015, pp. 188–197.

[10] G. Hasslinger, K. Ntougias, and F. Hasslinger, "A New Class of Web Caching Strategies for Content Delivery," in *IEEE International Telecommunications Network Strategy and Planning Symposium (Networks)*, 2014, pp. 1–7.

[11] Z. Wang, J. Xia, and B. Luo, "The Analysis and Comparison of Vital Acoustic Features in Content-Based Classification of Music Genre," in *IEEE International Conference on Information Technology and Applications (ITA)*, 2013, pp. 404–408.

[12] O. Lartillot, P. Toiviainen, and T. Eerola, "A Matlab Toolbox for Music Information Retrieval," in *Data Analysis, Machine Learning and Applications*. Springer, 2008, pp. 261–268.

[13] B. L. Sturm, "The GTZAN Dataset: Its Contents, its Faults, their Effects on Evaluation, and its Future Use," *arXiv preprint arXiv:1306.1461*, 2013.

[14] M. Gillhofer and M. Schedl, "Iron Maiden while Jogging, Debussy for Dinner?" in *International Conference on Multimedia Modeling*. Springer, 2015, pp. 380–391.

[15] C. Laurier, O. Meyers, J. Serrà, M. Blech, P. Herrera, and X. Serra, "Indexing Music by Mood: Design and Integration of an Automatic Content-based Annotator," *Multimedia Tools and Applications*, Vol. 48, No. 1, pp. 161–184, 2010.

[16] Y. M. Costa, L. S. Oliveira, and C. N. Silla, "An Evaluation of Convolutional Neural Networks for Music Classification using Spectrograms," *Applied Soft Computing*, Vol. 52, pp. 28–38, 2017.

[17] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer Learning for Music Classification and Regression Tasks," in *International Society of Music Information Retrieval (ISMIR)*, 2017.

[18] C. Koch, J. Pfannmüller, A. Rizk, D. Hausheer, and R. Steinmetz, "Category-aware Hierarchical Caching for Video-on-demand Content on YouTube," in *ACM Multimedia Systems Conference (MMSys)*, 2018, pp. 89–100.

[19] G. Szabo and B. A. Huberman, "Predicting the Popularity of Online Content," *Communications of the ACM*, Vol. 53, No. 8, pp. 80–88, 2010.

[20] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using Early View Patterns to Predict the Popularity of YouTube Videos," in *ACM International Conference on Web Search and Data Mining*, 2013, pp. 365–374.

[21] Y. Tang, "Deep Learning using Linear Support Vector Machines," in *In ICML*, 2013.

[22] R. E. Thayer, *The Biopsychology of Mood and Arousal*. Oxford University Press, 1990.

[23] J. Bai, J. Peng, J. Shi, D. Tang, Y. Wu, J. Li, and K. Luo, "Dimensional Music Emotion Recognition by Valence-arousal Regression," in *IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 2016, pp. 42–49.

[24] Y. Song, S. Dixon, and M. Pearce, "Evaluation of Musical Features for Emotion Classification," in *International Society for Music Information Retrial (ISMIR)*, 2012, pp. 523–528.

[25] H. Shahmansouri and J. Z. Zhang, "An Empirical Study on Mood Classification in Music through Computational Approaches," in *IEEE International Conference on Systems and Informatics (ICSAI)*, 2016, pp. 1050–1055.

[26] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson, "FMA: A Dataset For Music Analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[27] C. N. Silla Jr, C. A. Kaestner, and A. L. Koerich, "Automatic Music Genre Classification using Ensemble of Classifiers," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 1687–1692.

[28] K. Müller and T. Vignaux, "Simpy: Simulating Systems in Python," *ONLamp.com Python Devcenter*, 2003.

[29] C. Koch and D. Hausheer, "Optimizing Mobile Prefetching by Leveraging Usage Patterns and Social Information," in *IEEE International Conference on Network Protocols (ICNP)*, 2014, pp. 293–295.

[30] H. Yu, L. Xie, and S. Sanner, "Twitter-driven YouTube Views: Beyond Individual Influencers," in *ACM International Conference on Multimedia*, 2014, pp. 869–872.