Intelligent Edge Learning for Personalized Crowdsourced Livecast: Challenges, Opportunities, and Solutions

Fangxin Wang, Jiangchuan Liu, Cong Zhang, Lifeng Sun, and Kai Hwang

Abstract

Recent years have witnessed the expeditious development of crowdsourced livecast, also referred to as crowdcast, breeding such industry upstarts as Youtube Live, Twitch Tv, Mixer, Douyu, and so on. Unlike traditional TV-based livecast that provides uniform services, in crowdcast, service providers are racking their brains to satisfy each viewer's personalized QoE demands since different viewers usually have quite diverse preferences. To achieve this goal, the key challenges lie in the accurate prediction of viewers' personalized QoE preferences and the cost-effective viewer serving at the network edge. We argue that the confluence of edge computing and recent advance in deep learning shed light on accommodating these challenges, enabling more intelligent personalized QoE provision. In this article, we first provide an overview of today's crowdcast solutions, highlighting the challenges and opportunities therein from uniform crowdcast to personalized crowdcast with a large-scale measurement. We then present an intelligent edge learning framework ELCast as a case study, which leverages convolutional neural network and deep reinforcement learning in an edge computing architecture for personalized crowdcast. Trace-driven experiments demonstrate the superiority of ELCast over state-of-the-art approaches.

Introduction

Recent years have witnessed the expeditious development of crowdsourced livecast, also referred to as crowdcast, breeding such industry upstarts as Youtube Live (https://www.youtube.com/live), Twitch Tv (https://www.twitch.tv/), and Douyu (https://www.douyu.com/), with billion dollars of revenue particularly from the younger generation every year. The huge dividends in the market as well as the prevalence of personal devices motivate a plethora of amateur broadcasters to conveniently contribute their own contents, such as sports, games, or even daily activities, further attracting a massive number of viewers. As reported by Twitch Tv, the monthly broadcasters achieved 5.76 million and the concurrent viewers even exceeded 1.95 million in 2020 (https://twitchtracker.com/statistics).

Unlike traditional TV-based livecast where viewers passively watch a few well-planned channels with uniform services, in crowdcast, viewers are free to select their preferred platforms and channels. In order to survive in the highly competitive market, recent service providers are racking their brains to diversify streaming services to meet each viewer's personalized QoE demands since different viewers usually have quite different preferences, such as the requirements for bit rate, streaming latency, and channel switching latency. For example, a viewer who frequently interacts with the broadcaster is usually highly sensitive to the network latency since the delayed messages will seriously affect the experience of all parties. In contrast, many silent viewers are more willing to enjoy high-definition (HD) watching experience even if they have to sacrifice network latency.

Following a typical two-layer cloud-end architecture, traditional crowdcast is confined to provide best effort but homogeneous streaming services, which is inflexible to satisfy the differentiated quality of experience (QoE) demands as all QoE metrics are optimized in a monolithic approach. The emerging edge computing points out a promising direction toward this goal by pushing the transcoding and viewer serving from the remote network core to the local network edge. The geo-distributed edge servers that are configured with different computation capacities, bandwidth, as well as network latencies, can provide differentiated streaming services for versatile QoE demands.

Even so, accommodating personalized QoE for viewers remains challenging. The first difficulty lies in profiling the personalized QoE preference of each viewer. It is unrealistic to survey each viewer's QoE preference since it can be highly dynamic over time; frequent inquiries will instead harm the experience. Thus, it is necessary to analyze viewers' implicit watching behavior patterns to profile the QoE preference. Second, with the numerous video contents and viewers as well as the limited edge resources, it is hard to achieve cost-effective video transcoding and viewer assignment at the edge servers, as we need to satisfy personalized QoEs to the maximum extent.

We argue that the recent breakthroughs on learning can facilitate the data processing capacities, enabling a thrilling development of edge

Fangxin Wang is with the Chinese University of Hong Hong, Shenzhen, and also with Simon Fraser University; Jiangchuan Liu is with Simon Fraser University; Cong Zhang is with the University of Science and Technology of China; Lifeng Sun is with Tsinghua University; Kai Hwang (corresponding author) is with the Chinese University of Hong Hong, Shenzhen.

Digital Object Identifier: 10.1109/MNET.011.2000281

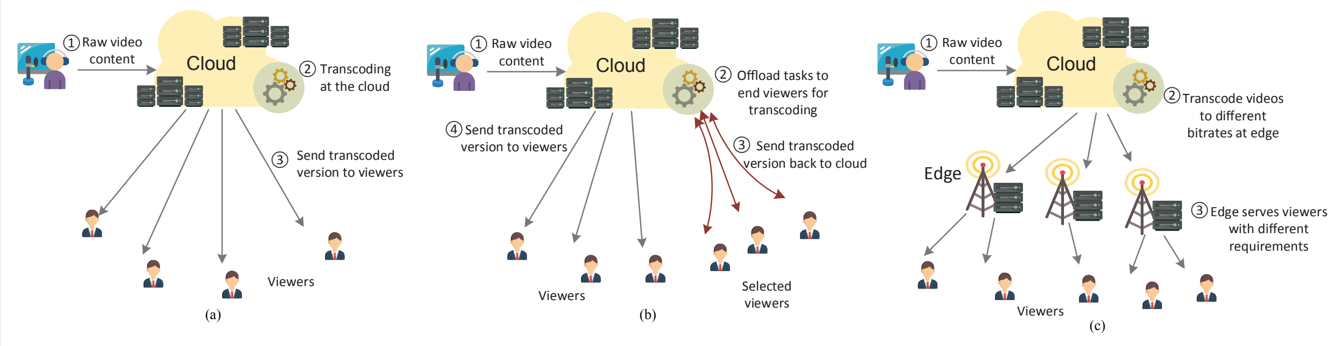

FIGURE 1. The system architecture of cloud-based approaches, end-assisted approaches, and edge-assisted approaches for crowdcast services: a) cloud-based architecture; b) end-assisted architecture; c) edge-assisted architecture.

computing services. The confluence of edge computing and deep learning is believed to bring new possibilities to achieve intelligent and personalized crowdcast services with cost-effective solutions.

In this article, we first provide a background overview of today's crowdcast solutions, highlighting the challenges and opportunities therein from uniform crowdcast to personalized crowdcast with a large-scale measurement. We then present an intelligent edge-learning-based framework ELCast as a case study, which integrates a convolutional neural network (CNN) and deep reinforcement learning (DRL) in the edge computing architecture for personalized crowdcast. Our trace-driven evaluation demonstrates the superiority of ELCast over state-of-the-art approaches. To our best knowledge, ELCast is the first to develop an integrated framework towards both the personalized QoE profiling and viewer assignment in crowdcast services.

From Uniform QoE to Personalized QoE: Background, Measurement, and Challenges

Crowdsourced Livecast: The Present Research

The booming development of crowdcast has drawn escalating attention from both industry and academia. The present research works mostly regard a crowd of viewers as an entity with uniform QoE demands and focused on optimizing such uniform QoE. According to the system architectures used, the current research can be divided into three categories — cloud-based approaches, end-assisted approaches, and edge-assisted approaches — as illustrated in Fig. 1. We summarize the present research works in Table 1.

Cloud-Based Approaches: Pioneer works rely on the resourceful cloud to serve the computation-intensive crowdcast services. In this architecture, the source videos from broadcasters are transcoded to multiple versions (bit rates) at the cloud to serve different viewers according to their network conditions. Given the expensive resource price and the unbalanced resource usage across different cloud servers, a dozen research works have explored the best workload scheduling to accommodate the dynamic task demands with cost-effective approaches [1, 2], so as to optimize the QoE such as latency and bit rate [3].

End-Assisted Approaches: Considering the requirement for huge concurrent computation resources as well as the high cost, existing research works have also explored the possibility of offloading the transcoding tasks to the devices of the massive amount of viewers. For example, He *et al*. [4] proposed to select highly interactive and stable viewers as candidates through a tree-based search algorithm and ask those candidates to contribute their computation capacities for joint transcoding. Following a similar idea, Liu *et al.* [5] outsourced the transcoding load to the crowdsourced end devices (or fog devices) to satisfy the proper bit rate requirements. Given the massive scale of terminal devices and the cheap computation resources, end-assisted approaches greatly save cost with carefully designed selection strategies. However, the uncertainty of working devices is likely to cause unstable transcoding services, which affects viewers' QoEs.

Edge-Assisted Approaches: The emerging paradigm of edge computing bridges the gap between satisfying QoE and reducing viewer serving cost. Edge servers can effectively under-

Approaches	Categories	Objectives	Technique summary
CALMS [1]	Cloud-based	Bit rate, cost	Adaptively lease and adjust cloud server resources to accommodate the dynamic task demands.
Cheng *et al.* [2]	Cloud-based	Latency, cost	Introduce a prediction-based scheduling algorithm to reduce the latency and resource cost.
CLGVS [3]	Cloud-based	Bit rate, latency, cost	Leverage the Lyapunov optimization for better latency, bit rate, and transcoding selection.
He *et al*. [4]	End-assisted	Bit rate, cost	Develop a tree-based search algorithm to select proper candidates for user-assisted transcoding.
Liu *et al*. [5]	End-assisted	Bit rate, cost	Formulate the end-assisted transcoding and viewer association as an integer programming problem.
LiveJack [6]	Edge-assisted	Bit rate, latency	Propose a transparent network service to seamlessly integrate edge and cloud for better QoE.
Ge *et al*. [7]	Edge-assisted	Bit rate, latency	Perform context-aware transient holding of video segments at the edge for high bit rate.
Zhang *et al.* [8]	Edge-assisted	Bit rate, latency, cost	Model the the viewer scheduling and transcoding selection as a Markov problem to solve.

TABLE 1. A summary of present research works on crowdcast with uniform QoE.

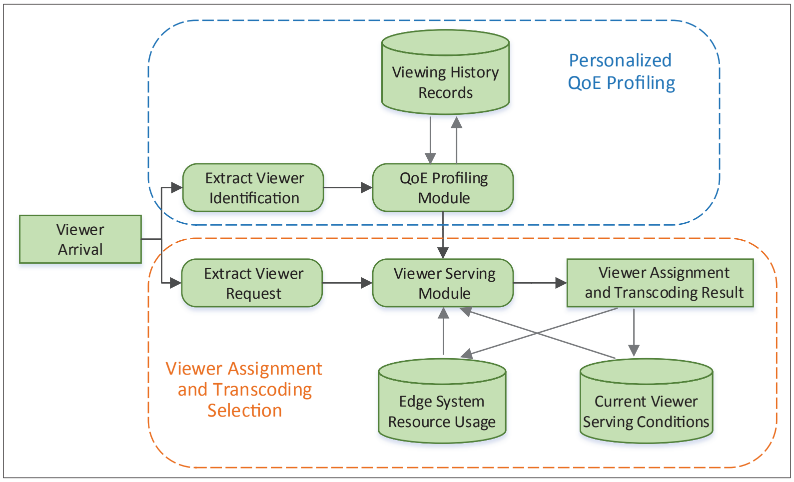

FIGURE 2. A typical system workflow of personalized crowdcast.

take the heavy burden of cloud servers, and provide differentiated streaming services to viewers directly. In this architecture, an edge server will transcode the local or remote video chunks to multiple bit rate versions, and serve viewing requests with different bit rates. Since different edge servers are featured with different computation capacities, network bandwidth, and latency, viewers with specific QoE preferences can be assigned to proper edge servers to optimize their QoEs. Existing works, however, have mostly focused on uniform QoE provision such as high bit rate [6] and low latency [7] through well-planned viewer scheduling and transcoding selection [8].

Personalized QoE in Crowdcast: Measurement and Challenges

In the crowdcast era, the core of content contributors has shifted from limited professionals to the massive crowd, enabling millions of amateur broadcasters to share their streaming content over the Internet. We next conduct a real-world measurement to understand the situations of the current crowdcast services.

We collected a dataset of viewers' watching logs from Inke.tv (a famous crowdcast platform in China), including 7.3 million records with viewer ID, channel ID, network type, location, and session start time and end time. To further study the interaction situations of viewers, we also collected a dataset from Twitch.tv, including viewers' arrival and departure logs, chatting messages, channel content types, and so on. According to our measurement, we have made the following three observations. First, the distribution of interaction messages is highly skewed for different viewers. We find that about 87 percent of viewers are silent viewers that never generate interaction messages during their session periods; more than 5 percent of viewers, however, are keen to communicate with the broadcasters and others, generating more than 10 chat messages per session. Second, a small portion of popular channels have attracted most viewers, revealing a long tail distribution. It is reported that only 2 percent of channels have more than 100 viewers on average. Third, different viewers have demonstrated quite heterogeneous watching durations for different channels. Based on our measurement, about 40 percent of viewers only watch each channel for 1 minute on average and then switch to other channels.

These observations indicate that it is no longer enough to offer uniform QoE in today's crowdcast services. On one hand, with limited computation and network resource, it is not practical and efficient to only provide uniform QoE in order to maximize viewers' QoE requirements. On the other hand, different viewers demonstrate quite diverse behaviors and thus have personalized QoE preferences, which calls for personalized QoE provision. In fact, the focus of crowdcast platforms has changed from optimizing the uniform QoE to offering personalized QoE to the massive amount of viewers.

However, it is not easy to achieve this goal. On one side, viewers have personalized QoE preferences in different aspects, for example, streaming latency, channel switching latency, as well as bit rate, and such preferences will even change dynamically with time, making the QoE profiling process extremely challenging. On the other side, accommodating personalized QoE demands in the edge-assisted architecture requires well-planned viewer scheduling and resource allocation for video transcoding and delivery. But the myriad video contents and various bit rate versions bring much burden and cost to the crowdcast platforms, making it difficult to achieve cost-effective service for personalized QoE. Existing model-based solutions cannot well address these two challenges effectively, which calls for a more integrated and intelligent framework for both accurate personalized QoE profiling and cost-effective viewer service.

Integrating Learning with Edge Computing toward Personalized QoE: The Opportunities

Recent breakthroughs in learning greatly facilitate the data processing and decision making capacity from intricate raw data, raising a technological revolution in many applications, such as video surveillance and transportation scheduling. The convergence of edge computing and deep learning brings more opportunities to empower more intelligent edge computing services, shedding light on personalized crowdcast services. The rich viewing behaviors and interaction patterns collected by the platform actually offer invaluable data resources that can be used for informed data-driven management.

A typical system workflow of personalized crowdcast to handle viewer requests is illustrated in Fig. 2. Generally, there are two main components in this system for personalized QoE provision, that is, personalized QoE profiling, and viewer assignment and transcoding selection. The first component is responsible for profiling each viewer's personalized QoE preferences through extensive analytics of his/her watching experience and interaction patterns. Given that many deep learning models, such as CNN [9] and long short-term memory (LSTM) [10], have been widely explored to mine data correlations and features with good effects, we try to use such advanced learning models for QoE preference prediction. The second component focuses on allocating the proper edge servers and transcoded video content to viewers to accommodate their personalized QoE demands with minimized system cost. We argue that DRL [11] reveals great potential in this area. It is able to automatically learn the allocation strategies in this context based on the past experience without hand-crafted feature selection.

Edge computing also plays an important role in this system. On one side, the geo-distributed edge servers deployed with heterogeneous configurations enable local viewer serving and video transcoding at the network edge. Such architecture can serve viewers with much lower streaming latency and channel switching latency as well as higher bandwidth provision compared to cloud-based architecture. On the other side, edge computing has provided specifically designed hardware foundations and platforms to better support deep learning running at the edge, which enables instant data processing at the network edge rather than transmitting all the data to remote cloud servers with large delay for decision.

The detailed working process of such a personalized crowdcast system is described as follows. When a viewer arrives, the system first extracts the identification information and the request information. With the identification information, the QoE profiling module searches the related history viewing records and leverages a deep learning model to predict the viewer's potential viewing behaviors, generating a personalized QoE preference. The viewer serving module takes as input the content request, personalized QoE preference, current system resource usage situation, and viewer serving situation to make a decision on viewer assignment and transcoding selection at edges. All system states will then update according to each viewer assignment decision. When a viewer leaves the crowdcast platform, the system will release the occupied resource, and system records will also update accordingly.

ELCast for Personalized QoE: A Case Study

In this section, we present the design of ELCast, a case of edge learning toward personalized QoE provision in crowdcast services.

Personalized QoE Profiling with Learning

We first introduce the QoE profiling module. Viewers' specific QoE preferences can be derived from their watching behaviors based on the following principles: viewers' interaction situations can reflect their sensitivity to the streaming latency; viewers who often browse many channels during a visit care more about channel switching latency; viewers who stay for a long time in a channel will prefer a high bit rate. Thus, we select interaction frequency, watching duration, and watching channel numbers as three indicators for personalized QoE profiling.

A deep learning model is designed to predict these indicators for each viewer, as presented in Fig. 3. We denote a viewer request as (u, loc, h, v), where u is the viewer ID, loc indicates the location, h is the channel ID, and v is the requested bit rate version. Then the input of our learning model can be represented as ($I_{u,h}$, $I_{u,all}$, $I_{all,h}$, $T_{u,h}$, $T_{u,all}$, $T_{all,h}$, N_u), where $I_{u,h}$, $I_{u,all}$, and $I_{all,h}$ indicate the interaction frequency of viewer u in channel

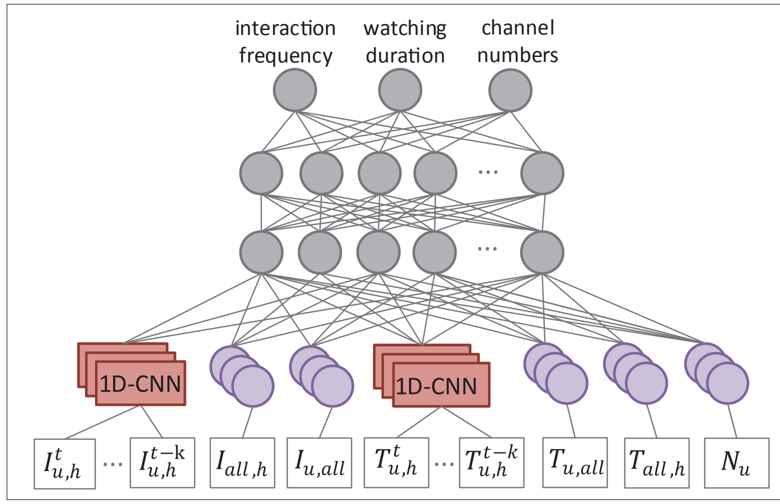

FIGURE 3. The deep learning model for personalized QoE prediction.

h, viewer u for all channels, and all viewers in channel h, respectively. $T_{u,h}$, $T_{u,all}$, and $T_{all,h}$ denote the similar meanings for watching durations. N_u denotes the average watching channel numbers per visit for viewer u. Note that for $I_{u,h}$ and $T_{u,h}$, the past k records are fed to 1D-CNN layers. We then concatenate all the processing results to multiple fully connected layers to predict the potential interaction frequency $\mathcal{I}^u$, watching duration $\mathcal{D}^u$, and watching channel numbers $\mathcal{N}^u$ for this request of viewer u. All the metrics in our model are normalized to the same scale before feeding into the neural networks.

After the prediction, the personalized QoE preferences are derived for each request as follows. $\overline{\mathcal{I}}$, $\overline{\mathcal{D}}$, and $\overline{\mathcal{N}}$ are the median value of interaction frequency, watching durations, and the watching channel numbers for all viewers, respectively. The QoE preference of viewer u for streaming latency can then be defined as $\alpha_1^u = \log(1 + (\mathcal{I}^u + 1)/\overline{\mathcal{I}} + 1))$, that for channel switching latency as $\alpha_2^u = \log(1 + \mathcal{N}^u/\overline{\mathcal{N}})$, and that for bitvrate as $\alpha_3^u = \log(1 + \mathcal{D}^u/\overline{\mathcal{D}})$. We adjust the three terms proportionally to make their sum be 10, so as to guarantee a constant sum of weight for every viewer. At last, we use the *QoE penalty* to represent the opposite number of the QoE level for a viewer u as $\mathcal{P}^u = \alpha_1^u \mathcal{L}s^u + \alpha_2^u \mathcal{L}w^u + \alpha_3^u \mathcal{B}^u$, where $\mathcal{L}s^u$, $\mathcal{L}w^u$, and $\mathcal{B}^u$ are the actual streaming latency, channel switching latency, and bit rate mismatch, respectively.

Deep Reinforcement Learning for Viewer Assignment and Transcoding Selection

We next present the design of a DRL-based learning model for viewer assignment and transcoding selection at edges [12], as shown in Fig. 4. An agent interacts with the system environment to make viewer assignment decisions. In each time step t, the agent observes a state s_t and can choose an action a_t to transit to the next state s_{t+1} with a received reward r_t. It will get accumulated rewards after every action until done. Our objective is to find the optimal policy that leads to the maximum accumulated reward. Note that we divide the time into small slots so that in each slot, DeepCast only processes one viewer request.

ELCast uses a state-of-the-art asynchronous advantage actor-critic (A3C) [11] model for learn-

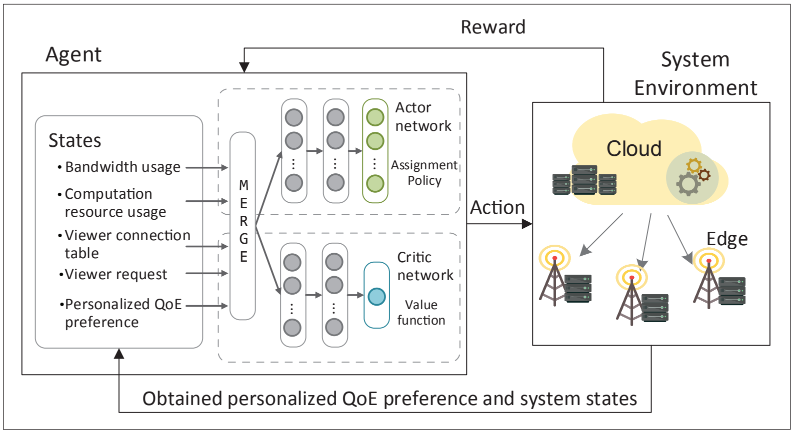

FIGURE 4. The deep reinforcement learning model that uses an actor-critic architecture for viewer assignment.

ing. It includes a policy (the actor network) and an estimate of the value function (the critic network). The actor network and the critic network share the previous part of network parameters except for the last several output layers. The model updates both the policy and the value function based on the returns of every t_{max} actions or until done.

Consider a practical online scenario where the viewers come and leave dynamically. The input state includes the system bandwidth usage, computation resource usage, the current viewer assignment situation, the current content request, and the profiled personalized QoE preference from the previous module. Based on the policy, the agent will select an action to decide whether to serve the viewer by an edge or the remote content delivery network (CDN). Once a viewer is assigned to an edge, the edge selects the bit rate version that leads to the maximum reward to serve the viewer, for example, transcoding from a higher local version or requesting it from the CDN. Note that the learning state will release the occupied resource if viewers leave the system. Thus, viewer departure will not affect the allocation process. Since the computation and bandwidth resources at each edge are limited, if the edge server's resources are not enough, the request will be redirected to the CDN for help.

We craft the reward to achieve the minimum overall *penalty*. Specifically, when we assign a viewer request to an edge (or the CDN), the viewer will obtain a personalized QoE and cause an extra cost for the system. The reward r_t can be defined as the opposite number of the overall penalty for the coming viewer u as $r_{u,t} = -\alpha Q^{u,t} - \beta(\mathcal{C}_T^{u,t} + \mathcal{C}_B^{u,t})$, where $\mathcal{C}_T^{u,t}$, $\mathcal{C}_B^{u,t}$ are the penalty value for transcoding cost and bandwidth cost for assigning viewer u, and α, β are weighted parameters to adjust the ratio between QoE and cost.

The training process follows the well-known policy gradient algorithm. The key idea is to estimate the parameter gradient direction toward the maximized total reward. In each step, we compute the gradient of the accumulated discounted reward regarding the parameters of the actor network and update them accordingly. To avoid the convergence to a suboptimal policy, we also add an entropy regularization term to the actor's update rule, which helps to encourage more exploration. The critic network can be trained following a temporal difference method. Once the model is well trained, we can select the viewer assignment action based on the output of the actor network.

EVALUATION

The performance of ELCast has been evaluated with trace-driven experiments on both the personalized QoE preference profiling and the viewer assignment. We collect multiple datasets for synthetic experiments, including the viewing session ID, viewer ID, and start time and end time from an Inke.tv dataset for 11 days with about 7.3 million records every day, the edge location from an iQiYi dataset with 2.8 million location records, the interaction situation of 300 popular channels from a Twitch.tv dataset with two-month records, and the network bandwidth situation from an FCC dataset. We select a rectangular area in Beijing from our location dataset as the target region (35 km × 21 km). For each viewing record, the bandwidth situation is randomly selected from the FCC dataset. In our training, we use the previous 80 percent data for training and the remaining 20 percent data for testing.

We first compare the personalized QoE preference model with four well-known baseline prediction methods, decision tree, random forest, support vector machine (SVM), and naive Bayes, all with the same input. Both root mean square error (RMSE) and mean absolute error (MAE) are used as indicators to evaluate their performance. Figures 5a to 5c show the prediction errors of interaction frequency (#/h), watching durations (h), and channel switching numbers (#), respectively. We can find that ELCast has the lowest prediction errors for both RMSE and MAE across all three metrics. Specifically, our deep learning model only has MAE of 1.16, 0,24, and 0.38 for the three metrics, which obtains 30, 32.5, and 49.2 percent lower MAE than the best baseline method (random forest), respectively.

We next examine the performance of the viewer assignment module. From our measurement, we have 6 kinds of streaming versions, for example, 1440p, 1080p, 720p, 480p, 360p, and 240p, with bit rate (megabits per second) of 4.3, 2.85, 1.85, 1.2, 0.75, and 0.3, respectively. To transcode from a higher version to the last 5 versions, the transcoding resource consumption (vCPU) is 3.3, 1.42, 0.82, 0.51, and 0.41, and the consumed time (second) is 0.27, 0.19, 0.16, 0.13, and 0.11, respectively. The streaming latency (second) is from 0.1 to 0.7 between end and CDN, from 0.02 to 0.1 between end and edge (or between edge and CDN). We consider three kinds of edge settings with different bandwidth and computation resources, for example, fat-edge (800 Mb/s and 64 vCPU), mid-edge (400 Mb/s and 36 vCPU), and thin-edge (200 Mb/s and 16 vCPU). The default QoE ratio setting is $\alpha = \beta = 0.5$.

We compare our DRL-based assignment model with three other models, including *cloud-cdn* that only uses cloud-based architecture, *online-heu* that follows a state-of-the-art online heuristic assignment approach [13], and *cost-only* that uses the same learning model but only considers the cost in the assignment. Figure 5d shows the CDF plot of the overall penalty by different approaches with

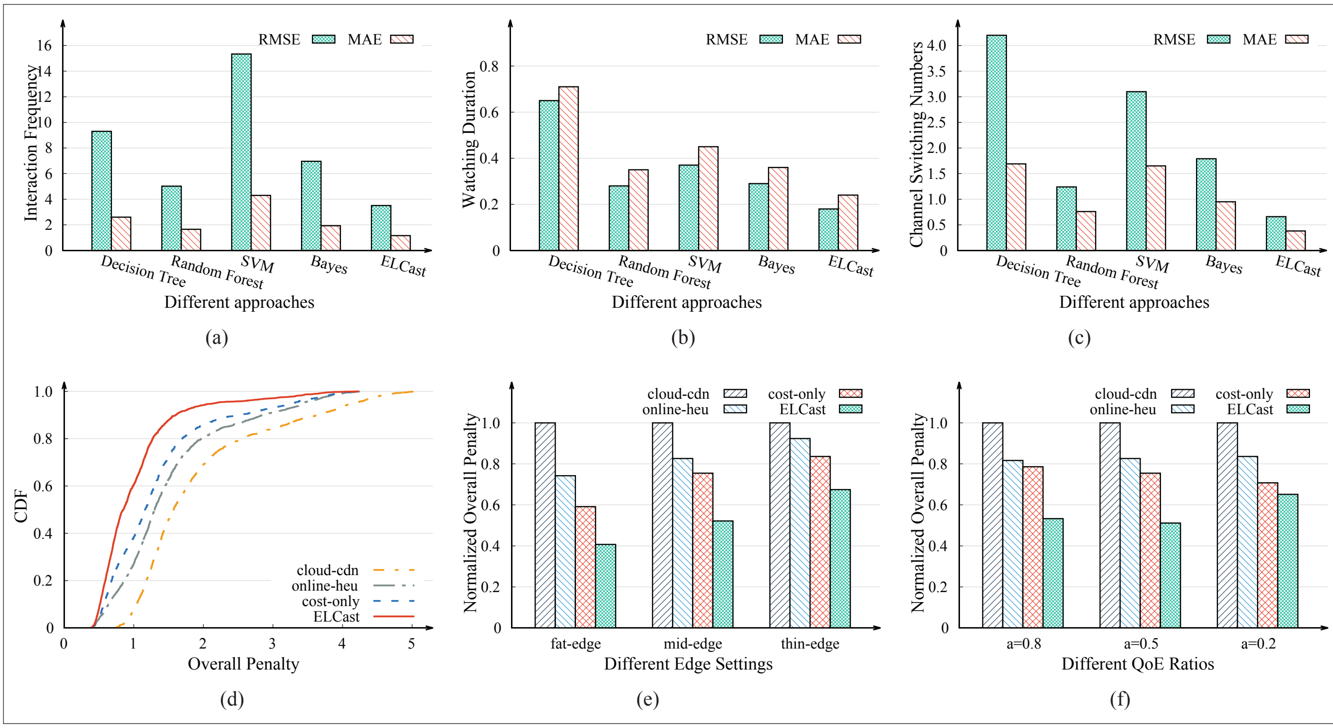

FIGURE 5. The evaluation performance for personalized QoE prediction and viewer assignment: a) prediction errors for interaction frequency; b) prediction errors for watching durations; c) prediction errors for channel switching numbers; d) the CDF plot of overall penalty by different approaches; e) the normalized overall penalty under different edge settings; f) the normalized overall penalty with different QoE ratios.

the mid-edge settings. We can see that ELCast outperforms all other approaches, with the penalty of more than 60 percent viewing requests less than one. Cloud-cdn has the worst performance since it cannot leverage the edge architectures to improve QoE provision. Figure 5e further presents the normalized overall penalty under different edge settings with cloud-cdn as the baseline. We can see that ELCast has the best performance, reducing by 59.3, 47.9, and 32.6 percent the penalty of cloud-cdn, and reducing by 45.1, 36.9, and 26.9 percent the penalty of online-heu for the three different edge settings, respectively.

Figure 5f considers the impact of different QoE ratios (a) on the assignment performance. Note that the sum of a and b is set as 1. Thus, a higher a means the crowdcast platform is more aggressive, caring more about viewers' QoE; a lower a means the crowdcast platform is more economical, with a high priority for saving cost. We can find that ELCast has the lowest normalized overall penalty of 0.532, 0.511, and 0.651 with different QoE ratios, which reduces by 34.8, 38.1, and 22.1 percent the penalty compared to online-heu. All these experiments demonstrate that ELCast can achieve better performance compared to all these baseline approaches.

Further Discussion

We have demonstrated the case of ELCast to integrate the advance of deep learning and edge computing for personalized crowdcast. The experiment results indicate that the deep-learning-based profiling is more capable of extracting the hidden features from viewers' watching history and achieving accurate prediction. Furthermore, they also reveal that ELCast can achieve more intelligent viewer assignment with better QoE provision while saving system cost, no matter whether provided with rich edge resources or rare edge resources. Such edge-learning-enabled personalized crowdcast can further be enhanced from the following perspectives.

Model Updating and Online Training

A deep-learning-based model for crowdcast relies on historical records, especially recent records, for accurate personalized QoE preference profiling and cost-effective viewer assignment. It thus requires periodic model updating to capture the possible viewing pattern changes. ELCast can well support model updating through an online training scheme. Note that the training time of the traditional methods is usually within tens of seconds while that of ELCast is about 4 minutes. However, this training overhead is negligible since we do not need frequent training. Re-training is required only when the system configurations (e.g., more edge servers are deployed to provide service) or the viewer access patterns are significantly changed, which is not frequent. In practical deployment, the learning models can be distributed at each regional server. Once a viewer is allocated, the target server will notify other servers to update their states accordingly.

Scalability and Multi-Agent DRL

ELCast employs a centralized DRL agent for viewer assignment and transcoding selection. However, when the number of edge nodes increases to a large scale, this centralized decision making method will no longer be efficient when faced with such massive choices. Fortunately, this framework can accommodate this scalability issue by extending to a distributed multi-agent DRL framework [14], where each edge node maintains a

DRL agent responsible for the local decision. These agents work collaboratively to well adapt to the distributed dynamics of viewer access patterns and achieve collective intelligence for scalable viewer assignments.

Privacy and Federated Learning

In personalized crowdcast, the personalized QoE prediction and the viewer assignment are expected to achieve better performance with more sufficient data of user behaviors, not only from the current platform, but also from other platforms or even other application sources. However, given the privacy and security issue, it is quite difficult to share the user data among different sources. The emerging federated learning [15] in recent years provides a promising solution. With federated learning, each data source maintains its own original data and only sends the model updates to other parties to achieve collaborative training. Such a learning scheme can greatly facilitate the cooperation of different data sources without worrying about the privacy and security issues. Thus, with richer data, the personalized crowdcast framework is expected to achieve better personalized QoE profiling and viewer serving.

Conclusion

In this article, we study the personalized crowdcast in the era of edge computing and deep learning. A review of existing approaches is presented. We then discuss the challenges and opportunities from uniform QoE to personalized QoE in crowdcast with a large-scale data-driven measurement. We present ELCast, a novel crowdcast framework, as a case study, which integrates the advances in both edge computing and deep learning to accommodate viewers' personalized QoE demands in today's crowdcast services. Trace-driven experiments demonstrated the superiority of ELCast compared to state-of-the-art approaches. We at last provide some promising directions that can be explored to enhance the edge learning framework.

Acknowledgments

This work is supported by a Canada NSERC Discovery Grant. The work of Cong Zhang is supported in part by the National Natural Science Foundation of China (NSFC No. 61902369). The work of Lifeng Sun is supported in part by NSFC (No. 61936011). The work of Kai Hwang is supported in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] F. Wang, J. Liu, and M. Chen, "Calms: Cloud-Assisted Live Media Streaming for Globalized Demands With Time/Region Diversities," *Proc. IEEE INFOCOM*, 2012, pp. 199–207.

[2] R. Cheng *et al.*, "A Cloud-Based Transcoding Framework for Real-Time Mobile Video Conferencing System," *Proc. IEEE MobileCloud*, 2014, pp. 236–45.

[3] Y. Zheng *et al.*, "Online Cloud Transcoding and Distribution for Crowdsourced Live Game Video Streaming," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 27, no. 8, 2016, pp. 1777–89.

[4] Q. He *et al.*, "Fog-Based Transcoding for Crowdsourced Video Livecast," *IEEE Commun. Mag.*, vol. 55, no. 4, Apr. 2017, pp. 28–33.

[5] X. Liu, M. Derakhshani, and S. Lambotharan, "Joint Transcoding Task Assignment and Association Control for Fog-Assisted Crowdsourced Live Streaming," *IEEE Commun. Letters*, vol. 23, no. 11, 2019, pp. 2036–2040.

[6] B. Yan *et al.*, "Livejack: Integrating CDNs and Edge Clouds for Live Content Broadcasting," *Proc. ACM Multimedia*, 2017, pp. 73–81.

[7] C. Ge *et al.*, "Qoe-Assured 4k Http Live Streaming via Transient Segment Holding at Mobile Edge," *IEEE JSAC*, vol. 36, no. 8, 2018, pp. 1816–30.

[8] Z. Zhang *et al.*, "QoS Aware Transcoding for Live Streaming in Edge-Clouds Aided Hetnets: An Enhanced Actor-Critic Approach," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 11, 2019, pp. 11,295–11,308.

[9] M. Chen and Y. Hao, "Label-Less Learning for Emotion Cognition," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 7, 2020, pp. 2430–40.

[10] R. Wang *et al.*, "Deepnetqoe: Self-Adaptive Qoe Optimization Framework of Deep Networks," 2020, arXiv:2007.10878.

[11] V. Mnih *et al.*, "Asynchronous Methods for Deep Reinforcement Learning," *Proc. ICML*, 2016, pp. 1928–37.

[12] F. Wang *et al.*, "Deepcast: Towards Personalized Qoe for Edge-Assisted Crowdcast With Deep Reinforcement Learning," *IEEE/ACM Trans. Networking*, vol. 28, no. 3, 2020, pp. 1255–68.

[13] Z. Wang *et al.*, "Joint Online Transcoding and Geo-Distributed Delivery for Dynamic Adaptive Streaming," *Proc. IEEE INFOCOM*, 2014.

[14] F. Wang *et al.*, "Intelligent Video Caching at Network Edge: A Multi-Agent Deep Reinforcement Learning Approach," *Proc. IEEE INFOCOM*, 2020.

[15] Q. Yang *et al.*, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intelligent Systems and Technology*, vol. 10, no. 2, 2019, pp. 1–19.

Biographies

Fangxin Wang [S'15, M'21] received his B.Eng. degree from Beijing University of Posts and Telecommunication, China, in 2013, his M.Eng. degree from Tsinghua University, China, in 2016, and his Ph.D. degree from Simon Fraser University, British Columbia, Canada, in 2020. He is currently an assistant professor at the Chinese University of Hong Kong, Shenzhen. His research interests include cloud and edge computing, multimedia systems and applications, networking optimization, and deep learning.

Jiangchuan Liu [S'01, M'03, SM'08, F'17] received his B.Eng. (Cum Laude) from Tsinghua University, Beijing, China, in 1999, and his Ph.D. from the Hong Kong University of Science and Technology in 2003, both in computer science. He is currently a full professor (with University Professorship) in the School of Computing Science at Simon Fraser University. He is a Fellow of the Canadian Academy of Engineering and an NSERC E.W.R. Steacie Memorial Fellow. He is a Steering Committee member of *IEEE Transactions on Mobile Computing*, Associate Editor of *IEEE/ACM Transactions on Networking*, and so on. He is a co-recipient of the Test of Time Paper Award of IEEE INFOCOM (2015), ACM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and ACM Multimedia Best Paper Award (2012).

Cong Zhang received his M.Sc. from Zhengzhou University, China, in 2012, and his Ph.D. from Simon Fraser University in 2018. Currently, he is an associate researcher in the School of Computer Science at the University of Science and Technology of China. His research interests include multimedia communications and cloud/edge computing.

Lifeng Sun [M'05] is currently a professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include video streaming, video coding, video analysis, and multimedia edge computing. He received the Best Paper Award for *IEEE Transactions on Circuits and Systems for Video Technology* in 2010, the Best Paper Award from *ACM Multimedia* 2012, and the Best Student Paper Award from MMM 2015, IEEE BigMM 2017, and NOSSDAV 2019.

Kai Hwang [LF] is a Presidential Chair Professor at the Chinese University of Hong Kong, Shenzhen. He also serves as the chief scientiest, Cloud Computing Center, Chinese Academy of Sciences. He received his Ph.D. in EECS from the University of California, Berkeley. He has published extensively in computer, Internet, and IT areas with a Google scholar citation of 18,600. His book, *Big Data Analytics for Cloud/IoT and Cognitive Computing* (Wiley, 2017), has been adopted by 20 universities including Stanford and Harvard. A CCF Foreign Council member, he received the Outstanding Achievement Award from the China Computer Federation (CCF, 2005) and the Lifetime Achievement Award from IEEE CloudCom 2012 for his pioneering work in parallel computing and distributed systems.