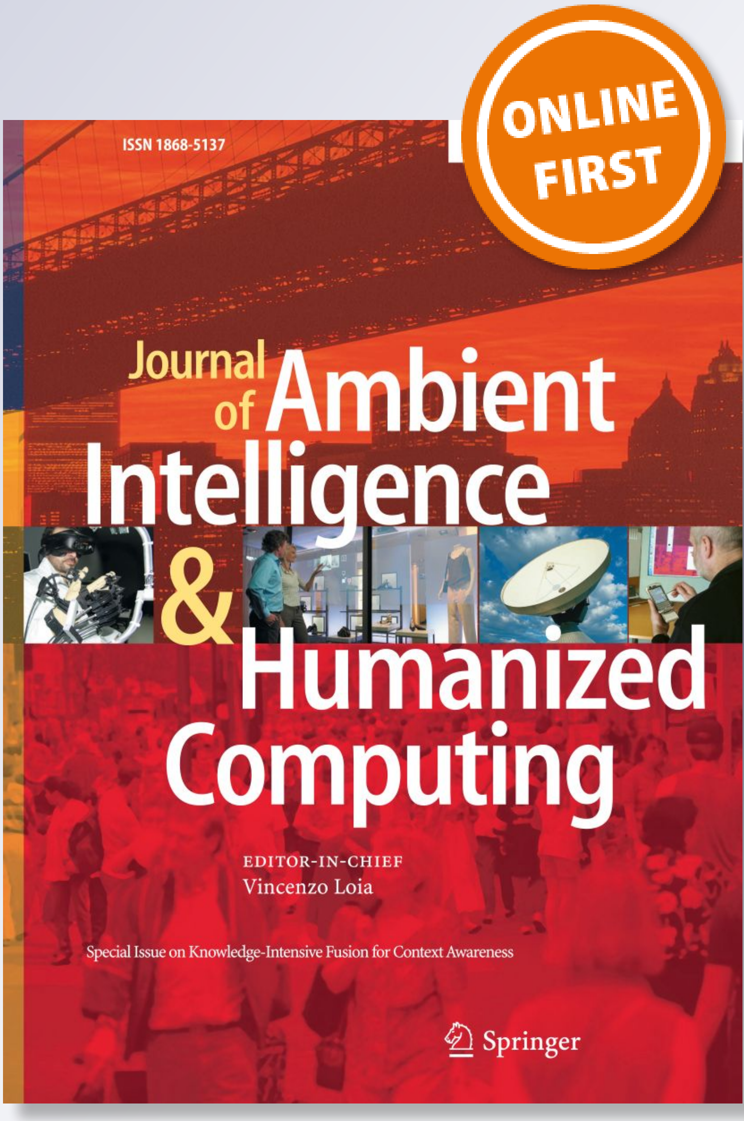
A Context-aware adaptive algorithm for ambient intelligence DASH at mobile edge computing

Jinsul Kim, Yonggwan Won, Changwoo Yoon, Jin-Young Kim, Sangho Park, JaeCheol Ryou & Linh Van Ma

Journal of Ambient Intelligence and Humanized Computing

ISSN 1868-5137

J Ambient Intell Human Comput
DOI 10.1007/s12652-018-1049-z



Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-018-1049-z

ORIGINAL RESEARCH

A Context-aware adaptive algorithm for ambient intelligence DASH at mobile edge computing

Jinsul Kim[1] · Yonggwan Won[1] · Changwoo Yoon[2] · Jin-Young Kim[1] · Sangho Park[3] · JaeCheol Ryou[4] · Linh Van Ma[1]

Received: 7 May 2018 / Accepted: 16 September 2018

Abstract

Adaptive streaming has recently emerged as a technology enabling high-quality streaming at various bitrates. One of the video streaming challenges remains in research topic nowadays that is choosing optimal segment base on network characteristics and streaming devices, such as network bandwidth, latency, the computational capacities of devices. Researchers have proposed many algorithms to overcome such issues within their predefined conditions. However, those proposed methods do not perform efficiently in the heterogeneous network today. Consequently, in this article, we present research on a context-aware adaptive algorithm for ambient intelligence dynamic adaptive employing mobile edge computing (MEC). Specifically, we apply deep learning in the adaptive algorithm which is installed at the MEC to assist clients in choosing the optimal streaming segments as well as reduce network latency. Furthermore, we apply the multilayer perceptron classifier with data obtained from various experiments of adaptive streaming algorithms then combine them in a general algorithm. In the analysis, we use network simulator NS3 as a tool to carry out the verification of our proposed method. As a result, the proposed research reduces network latency as well as improve quality streaming compared to existing approaches.

Keywords Adaptive streaming · DASH · Deep learning · Mobile edge computing · Adaptive algorithm

1 Introduction

Ultra-high-definition streaming is becoming the standard for video streaming in the next decade due to the constant need for high or ultra-high quality video streaming from users. Eventually, it becomes a hot research topic in the network transmission field. Researchers from all over the world have applied many techniques to transmit multimedia over the Internet efficiently (Van Ma et al. 2016). For example, one of the recent technology which has been recently emerged, adaptive streaming, has been deploying over the Internet streaming system widely. It transcodes original video raw files into different videos with different quality. Subsequently, it segments those transcoded videos into small segments which has a duration ranging from 2 to 10 s. The technique employs HTTP (Hypertext Transfer Protocol) to transmit those video segments. Technically, it gains benefits from HTTP where packets can easily traverse firewall and NAT (Network Address Translation) devices. In addition, HTTP does not require to maintain the connection as traditional server-driven video streaming such as RTSP (Real Time Streaming Protocol). Hence, it reduces persistent requirement between client application and server.

Since the adaptive bitrate streaming (ABS) emergence, several versions adaptive streaming system has been made that are employed by Microsoft, Apple, Adobe Systems, etc. For example, Adobe HTTP Dynamic Streaming was implemented in the latest version of Flash Player Media. Apple HTTP Live Streaming (HSL) is a part of iOS and Quick Time supporting video on demand and live streaming.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12652-018-1049-z) contains supplementary material, which is available to authorized users.

✉ Linh Van Ma
linh.mavan@gmail.com

1 School of Electronics and Computer Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 500-757, South Korea

2 Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, South Korea

3 JiranSoft Researh Center, 87, Daehak-ro, Yuseong-gu, Daejeon, South Korea

4 Department of Computer Engineering, Chungnam National University, 99, Daehak-ro, Yuseong-gu, Daejeon, South Korea

Published online: 21 September 2018

Furthermore, researchers have been employing artificial neural networks so-called reinforcement learning cooperating with ABS to improve quality of experience (QoE) fairness in recent years.

The crucial component of the technique is adaptive algorithms implemented at the client. It allows a user to adapt streaming quality seamlessly to network condition such as bandwidth, latency, and computational resource. Basically, we can implement the algorithms by adjusting video quality based on bandwidth fluctuation. For example, if the bandwidth is low around 300 Kbps, the client might request for low video quality at 360 p (Progressive scanning). If the bandwidth is much higher around 21 Mbps, the server might response the client with 2160 p (4K) video. In addition, we can implement the algorithms based on client buffer state. Most of the research in adaptive streaming have carried out to improve these adaptive algorithms. They either consider bandwidth or buffer state as criteria to optimally choose streaming segments in their algorithm (Feng et al. 2010; Mok et al. 2012; Vergados et al. 2014a, b, 2016). Some other researches (Kim et al. 2018; Lee and Guan 2012; Ma et al. 2017) combine the two criteria with mathematics techniques such as optimization to improve video streaming quality as well as reduce video segment response latency. Each proposed algorithm has their characteristic to deal with a particular network state or many network conditions. Consequently, combining all the characteristics of outstanding algorithms in the adaptive algorithm field can lead to generalizing adaptive algorithms into one efficient streaming algorithm.

In this research, we carry out a machine learning technique which is MLP (Chauhan et al. 2017; Huang et al. 2017; Tang et al. 2016) (Multilayer Perceptron) in the adaptive algorithm of ambient intelligence DASH (Dynamic Adaptive Streaming over HTTP). More specifically, we apply multilayer perceptron into various proposed adaptive algorithms based on a result obtained from a network simulation tool (NS3). First, we run the proposed algorithms in NS3 in different network condition to build learning model with test data and training data. Secondly, we apply the model back to the simulation and compare result outcome with other based algorithms. Furthermore, we exploit mobile edge computing (MEC) (Orsini et al. 2018; Roman et al. 2018) in our learning algorithm. More detail, we reduce computational resource at the client by locating the algorithm in the MEC where it can detect and deduce next quality segments of the client based on the network condition.

The rest of this paper is organized as follows. Section 2 provides related works with the latest proposed adaptive algorithms in the research literature as well as relevant research in MEC. Section 3 offers the logical of the learning algorithm and streaming system overview. Subsequently, we simulate our proposed algorithm in NS3 and present a discussion about the archived result. Finally, we conclude our research in Sect. 5 with the outcome and future research.

2 Related work

This section firstly introduces recent research in the field of adaptive streaming. Subsequently, we move to describe some techniques in deep learning and multilayer perceptron.

One research on receiver-driven approach, AAASH (Adaptation algorithm for adaptive streaming over HTTP) (Miller et al. 2012) described that their adaptive algorithm could avoid unnecessary video quality fluctuations while maximizing the QoE. Suppose that the algorithm is performed at time t instantly after a segment has completely downloaded, the algorithm has two input arguments. The first one is buffer level at a particular time between 0 and t, and the second one is the past information about available bandwidth between 0 and t. The output of the algorithm is the selected representation for the next segment and the condition to start downloading the segment which is the minimum buffer level in seconds of playback.

Another approach employing fuzzy logic in the adaptive algorithm, FDASH (Fuzzy-Based MPEG-DASH Adaptation Algorithm) (Vergados et al. 2016), they apply a fuzzy technique to deliver undisrupted video playback and control buffering time while playing back at the client. In the algorithm, the input variables are the buffering time which is the difference of the last buffering time from the previous one, and the previously received segment waits at the client until it starts playing. The output represents an increase/decrease factor of the resolution of the next segment. They model the two inputs into linguistic variables before processing in the fuzzy system. Subsequently, they simulated their proposed method and compared with other approaches. As a result, their approach optimally offers video streaming with rarely under buffer flows and low video resolution fluctuations.

Oppose to the two above research, OSMF (QoE-aware DASH System) (Mok et al. 2012) takes into account of measuring the available network bandwidth. They exploited an available bandwidth measurement method but it needs at least several seconds to achieve the estimation, and it does not adaptively adjust to network conditions. Consequently, they proposed a subjective QoE quantization method based on the fact that video representation in DASH has only a few numbers of values. As a result, their proposed method demonstrated that users do not prefer an abrupt switching compared to a gradual quality change between the worst and best quality levels.

In the field of CDN (Content Delivery Network) research, SFTM (Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network) (Liu et al. 2012) and RAAHS (Rate adaptation for adaptive HTTP streaming)

(Liu et al. 2011), addressed a problem in adaptive segment fetching with a conventional and parallel approach for DASH. In the research, they used multi-step switch-down and step-wise switch-up method based on a detection of the spare network's capacity and congestion. By doing a simulation, their proposed algorithms outperformed in selecting bitrate of DASH.

Last but not least, SVAA (Towards agile and smooth video adaptation in dynamic HTTP streaming) (Tian and Liu 2012), proposed video rate control algorithms balancing smoothness of video representation. More detail, they have one crucial adjustment factor which is a product of three essential elements. First, buffer size adjustment is an increasing function of buffer size deviation. Secondly, buffer trend adjustment is a is a rising function of buffer size growth. Thirdly, video chunk size adjustment is a decreasing function of the previous video rate. On the basis of extensive experiments, their method presented a highly efficient video streaming in the real network environment.

In the field of MEC cooperating with adaptive streaming, the authors (Xu et al. 2017) proposed a MEC-based adaptive bitrate streaming (ABS), which integrates ABS technology and content caching. Explicitly, the MEC server considers as a caching component as well as flexibly adjust bitrate between clients and the server. As a result, their proposed method improves both system throughput and video cache compared to other existing research.

3 Adaptive MEC streaming system and context-awareness algorithm

Initially, the DASH streaming system has one streaming server which has available videos for a request coming from clients which support adaptive streaming. However, the server itself does not guarantee consistent network connection as well as high quality of service (QoS). Consequently, researchers derived the system to MEC-based whereby it reduces network latency as well as workload for the server. The overview is depicted in Fig. 1 which is our based system structure. The main server is supposed to have DASH video content, MEC acts as an intermedia between local devices and the server. Technically, we install the multiplayer perceptron in MEC where it could recognize streaming requests and predictive the next request based on current network condition and some other factors which can affect the quality of service such as computational resource.

In the MEC as shown in Fig. 2, we implement two functions. First, one function supports adaptive streaming (DASH), and another one is a learning function which employs multilayer perceptron in deep learning discipline. After receiving a request from a client, the server responses the request via MEC where we can implement a content-awareness component (deep learning algorithms). The component can deduce future requests coming from clients based on past processed requests. More specifically, the learning function learns the behavior obtained from some proposed algorithms recently, such as AAASH (Miller et al. 2012), OSMF (Mok et al. 2012), RAAHS (Liu et al. 2011), SVAA (Tian and Liu 2012), FDASH (Vergados et al. 2016), SFTM (Liu et al. 2012). Furthermore, we apply the MLP in the eight adaptive algorithms and combine them in one general

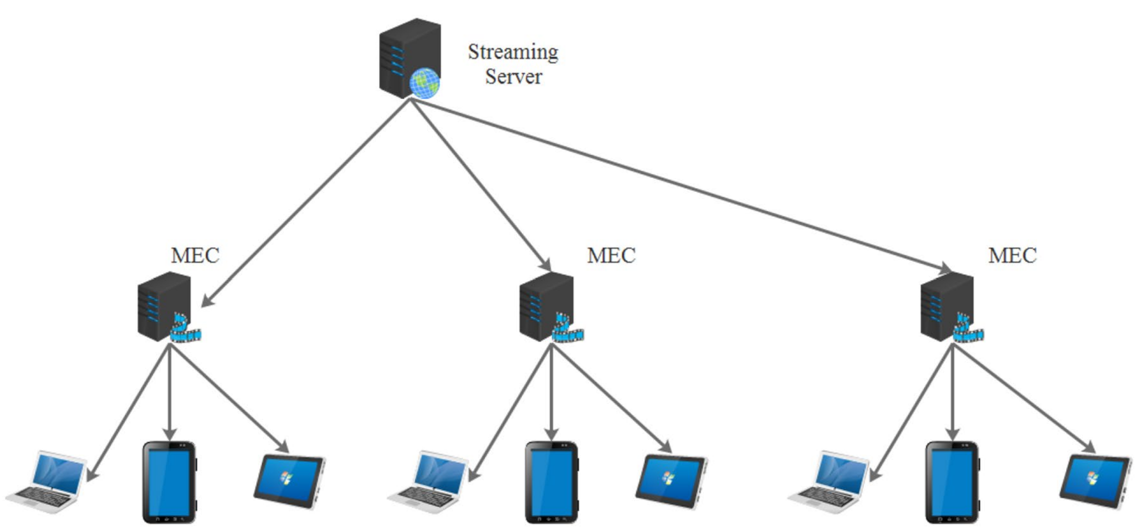


Fig. 1 Overview of MEC-based DASH System

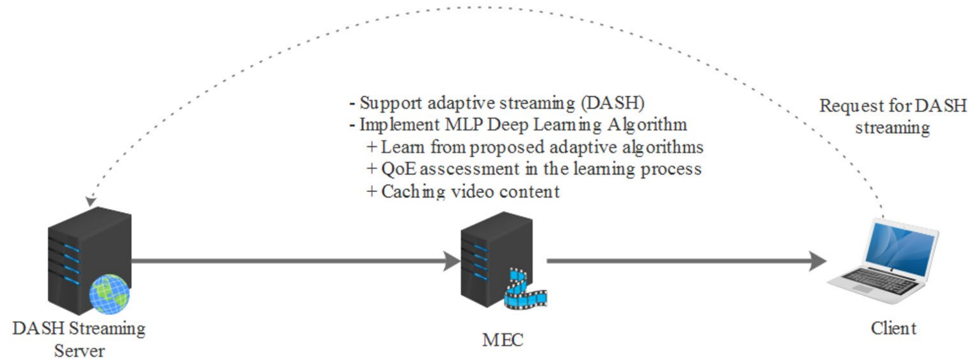


Fig. 2 Description of MEC functions in the role of intermediate between a server and a client

algorithm to find out how the general algorithm processes a request under a specific network condition.

In the context of ambient intelligence, each adaptive algorithm performs to select output bitrate based on different factors such as bandwidth, streaming buffer at the client, the differential bandwidth between previous and current estimation, and the differential buffer between previous and current estimation. Consequently, we choose the input layer of MLP differently based on the input of an adaptive algorithm. We use the list of algorithms shown in Table 1 which includes algorithms described in the related works section, the input and output of algorithms.

An MLP is a class of feedforward artificial neural network, which consists of three minimum layers nodes. The first two crucial nodes are input and output layer and at least one hidden layer. Each node considers as a neutron that employs a nonlinear activation function. Besides, MLP belongs to a set of supervised learning technique which utilizes back-propagation for training the process. Regarding an algorithm for adaptive streaming, MLP can distinguish data that is not linearly separable. Consequently, we apply MPL for each algorithm and adjust a number of neurons in the input layer and output layer. The process of MLP algorithm is shown in Fig. 3 where the input layer is bandwidth, buffer level, and bandwidth differential. The output of the algorithm is a selected bitrate for the next request of a client.

The detail process of the MLP algorithm is described in Fig. 4. We first simulate six selected algorithms in NS3 in various network conditions within a sufficient amount of time. Subsequently, we collect the achieved data after finishing the simulation and separate the data into two parts. The first part (around 90% of the record data) is used to train the MLP model, and the second part (10% of the record) is used to test the trained model. During the training and testing process, we adjust a number of hidden layers, a number of neutrons, learning rate and momentum to get the highest recognized pattern in the testing set.

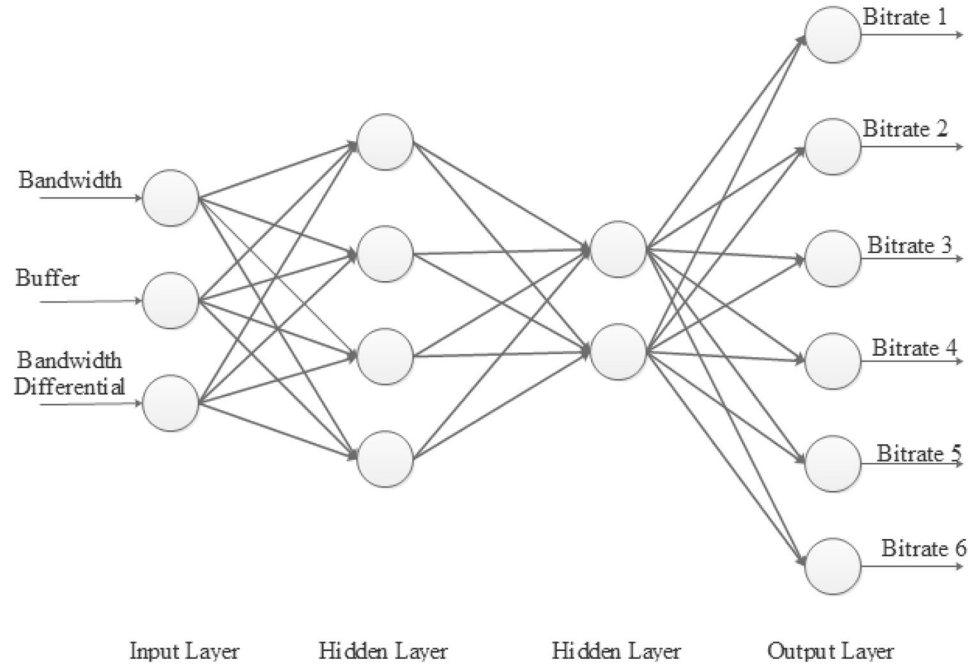
Support that we have m MECs denoted as $M_1, M_2, \dots, M_m$. Each MEC (M_i) represents an incoming/outgoing streaming request for n_i ($n_i \neq n_j$, $\forall i, j \in N^+$) clients. We consider each streaming request coming from a client to a MEC as load including bandwidth, latency, and computation. The load can be labeled as l_{ij} for a client *ith* in the MEC *jth*. The total load of a MEC (TE_i) can be calculated as Eq. (1). Besides, the total load of the streaming server (L) can be calculated as Eq. (2).

$$TE_i = \sum_{j=1,n_i} l_{ij} \tag{1}$$

Table 1 List of adaptive algorithms in DASH with input and output

Algorithm	Input	Output
AAASH	+ Buffer level + Past information about available throughput	+ Selected representation for the next segment + Minimum buffer level to start downloading segment
FDASH	+ Buffering time + Buffer Differential	+ Selected representation for the next segment
OSMF	+ Current quality level + Bandwidth	+ Selected representation for the next segment
RAAHS SFTM	+ Current DASH segment + Bandwidth	+ Selected representation for the next segment
SVAA	+ Buffer trend + Buffer size + Video chunk size	+ Selected representation for the next segment

Fig. 3 Illustration of MLP in the adaptive streaming algorithm



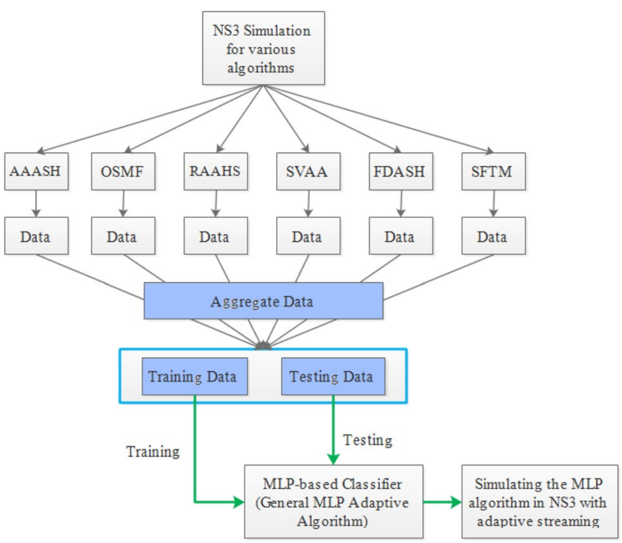


Fig. 4 Overview of MLP general algorithm for adaptive streaming

$$L = \sum_{i=1,m} \sum_{j=1,n_i} l_{ij}. \tag{2}$$

The Eqs. (1) and (2) are used to determine MEC load and server load, which avails for allocate network resource and computational resources in a network function virtualization (NFV) environment. The NFV is described in detail in the experiment section including implementation and usage.

4 Experiment results

In this section, we carry out several experiments in NS3 and real network environment built on our laboratory facilities. More detail, we first simulate and visualize with one existing algorithm, FDASH to demonstrate how our proposed MLP algorithm work. Consequently, we move to integrate all the mentioned algorithms as well as illustrate the result.

Regarding FDASH, we utilize Waikato Environment for Knowledge Analysis (WEKA), which is a suite of machine learning software written in Java, to illustrate MLP algorithm and its learning processes. In the demonstration, we set learning rate equal to 0.3, momentum to 0.2, and epoch to 300. The input of the is BufferEstimate and BufferDifferental as described in the problem above. The output is the set of video representation {45,000, 89,000, 131,000, 178,000, 221,000, 263,000, 334,000, 396,000, 522,000, 595,000, 91,000, 1,033,000} (bps) in which we can only choose one representation at once time.

In the first simulation, we run the program with one hidden layer and three perceptrons as shown in Fig. 5, and the result is shown in Table 2. After training, the model is evaluated using testing data which has 232,684 instances. It correctly classified almost 91% of the total instances with root mean squared error is 0.1015. Secondly, we run the program with one hidden layer and eight perceptrons as shown in Fig. 6a. After training, the model is applied to test data which has 232,684 instances. It correctly classified almost 91% of the total instances with root mean squared error is 0.1025. Thirdly, we run the program with two hidden layers (3, 2) perceptrons as shown in Fig. 6b

Fig. 5 One hidden layer with three perceptrons

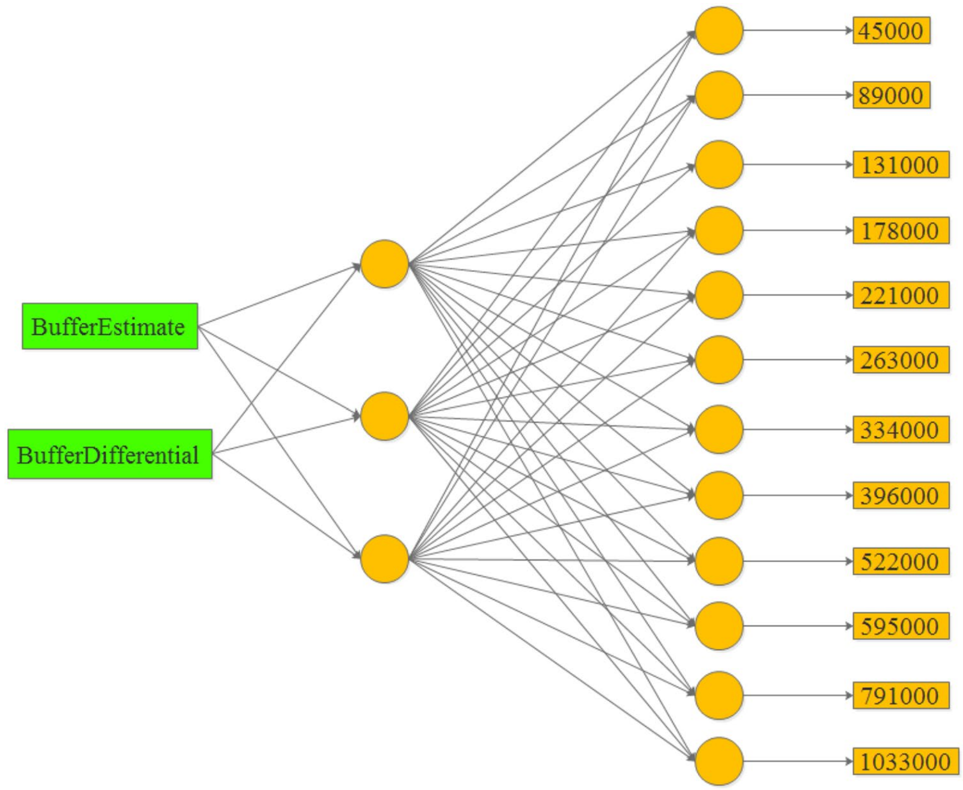


Table 2 Result of classification with one hidden layer and three perceptron

Correctly classified instances	210,890	90.6336%
Incorrectly classified instances	21,794	9.3664%
Root mean squared error	0.1015	
Total number of instances	232,684	

and the result is shown in Table 3. After training, the model is applied to test data which has 232,684 instances. It correctly classified almost 91% of the total instances with root mean squared error is 0.1019.

The three-layer perceptron networks have a similar result. That is, the correct classification of the testing set is about 91% although it has one more layer and fewer number of perceptrons compared to the first MLP network. In addition, root mean squared error (RMSE) comparison between the three is shown in Fig. 7. The error does not change after 50 epochs, but the stable value differs for each test. With the same learning rate and momentum, the one has one hidden layer, and eight perceptron has lowest RMSE. Besides, we set different learning rates and execute the classification program and got the comparison of convergence of the MLP shown in Fig. 8. It shows that the learning rate (0.2) leading to faster convergence than the other two.

We then move to simulate the general algorithm which is implemented on the client side. First, we set up 50 clients located in 100 Mbps network bandwidth capacity. The highest resolution is up to 4K bitrate which causes clients to compete with other to have their best streaming quality. We compare our proposed algorithm with other previous research FDASH and OSMF, which are the outstanding adaptive algorithms in the recent years. The result of streaming bitrate comparison is shown in Fig. 9. It shows that the MLP-based algorithm has a higher streaming quality (about 17%) compared to FDASH and 62% compared to OSMF.

Regarding real network experiment, we build a MEC in laboratory network environment. First, we implement an NFV system into five super physical computer running on Ubuntu 16.04. The streaming system and MEC are implemented based on the virtual network system. Then, we simulate the general algorithm with and without MEC. In this experiment, we consider latency as a metric to compare our proposed algorithm with and without MEC. As a result, the latency of MEC-based MLP algorithm lower than the one without MEC about 20% as shown in Fig. 10.

The general algorithm inherits many characteristics from other previous research since we train the algorithm using retrieved simulation result data. It is a reason explaining that the algorithm has better result compared to the based studies. Recall that, this research mainly focus on MEC where a client does not concern about network

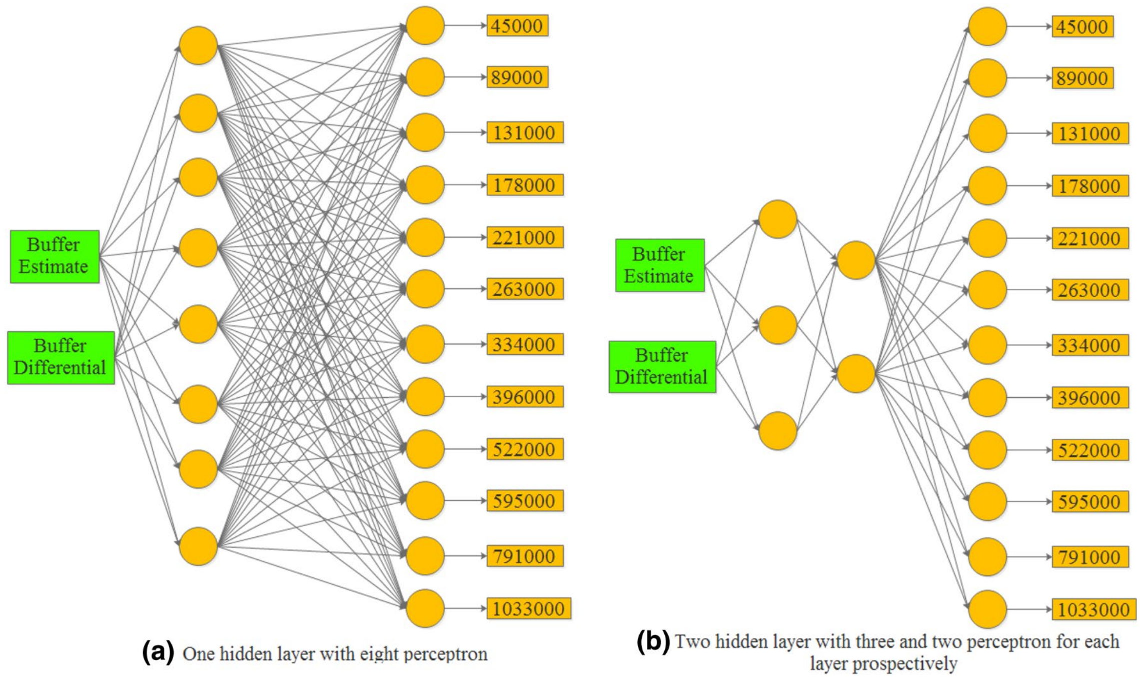


Fig. 6 Simulation with one and two hidden layers

Table 3 Result of (3.8) two hidden layers

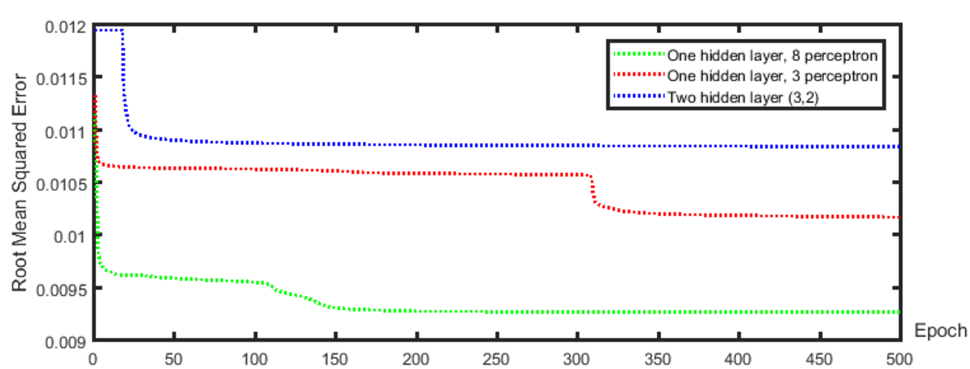
Correctly classified instances	210,948	90.6586%
Incorrectly classified instances	21,736	9.3414%
Root mean squared error	0.1019	
Total number of instances	232,684	

condition, all of the adaptive strategies are located on MEC server. On the one hand, it assists clients to reduce computing resource since it must calculate the next adaptive DASH segment under a particular network condition. On another hand, it reduces network latency since MEC is located nearest to the client location.

5 Conclusions

In this paper, we presented research on intelligent and context-aware adaptive algorithm employing MEC. More detail, we applied multilayer perceptron concept in the adaptive algorithm which is installed at the MEC to assist streaming clients in choosing the optimal streaming segment as well as reducing network latency. In the analysis, we simulated our proposed algorithm using network simulator NS3. As a result, our proposed method improved quality streaming as well as reduced network latency compared to existing approaches up to 5%. In the future research, we intend to dig deeper into the adaptive algorithm with more complicated neuron network structure. Regarding ambient intelligence MEC, we are going to carry out an intensive experiment

Fig. 7 Comparison of RMSE between the three MLP



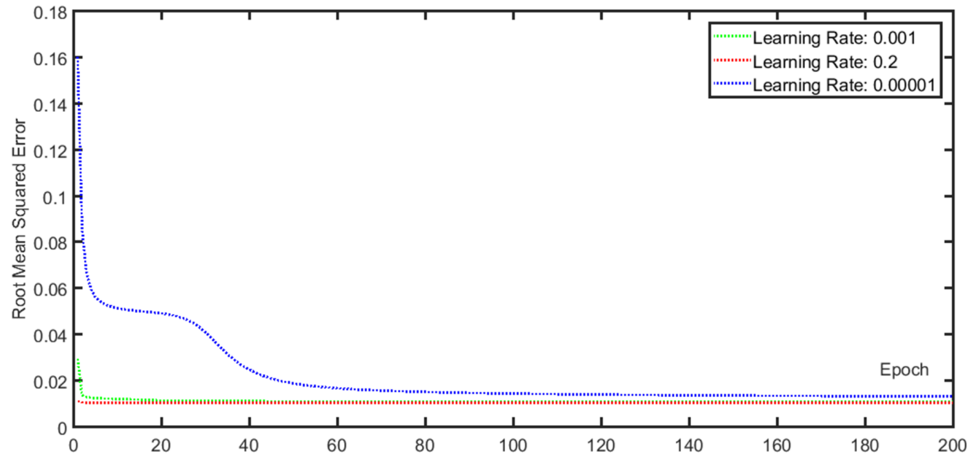


Fig. 8 Comparison of convergence between different learning rates

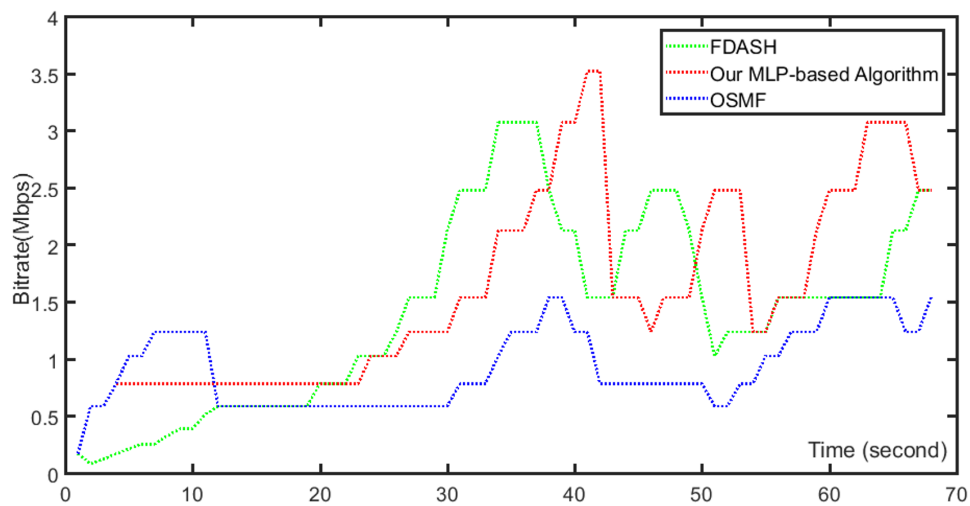


Fig. 9 Streaming bitrate comparison between our MLP-based algorithm with FDASH and OSMF

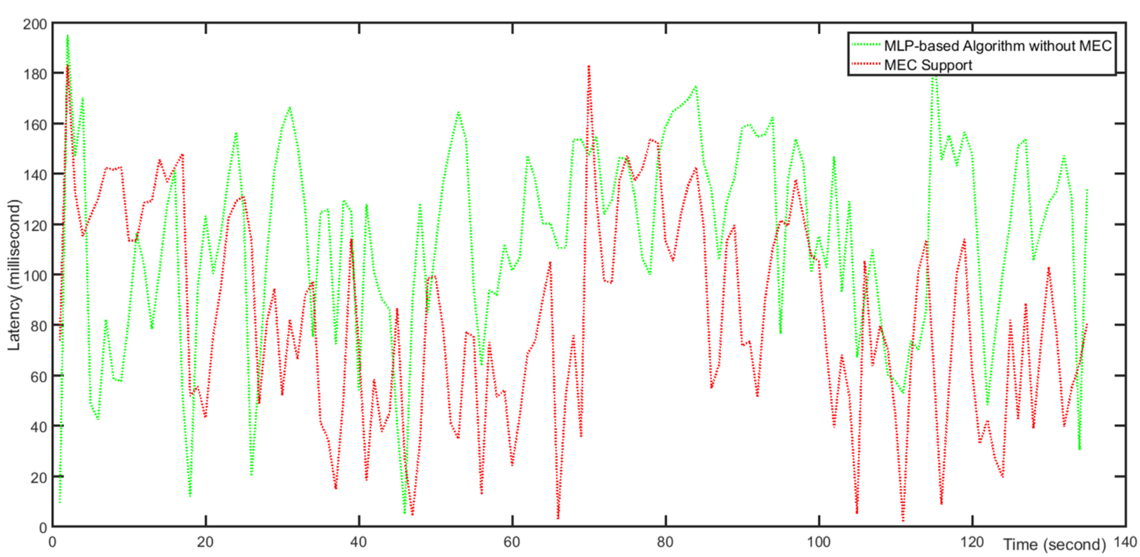


Fig. 10 Latency comparison between the general algorithm with and without the support of MEC

with network simulation tool to demonstrate and highlight its advantage in supporting context-awareness ultra-high-definition adaptive streaming. Furthermore, we plan to extend our previous research (Van Ma et al. 2017) on multimedia transcoding based on MLP to reduce computational resources.

Acknowledgements This research was supported by the IT R&D program of MSIT(Ministry of Science and ICT), Korea / NIPA(National IT Industry Promotion Agency) 12221-14-1001, Next Generation Network Computing Platform Testbed. Besides, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST)(Grant No. NRF-2017R1D1A1B03034429). Furthermore, this research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). Lastly, this work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(R0190-17-2009,Development of endpoint protection technology using white list and context-aware).

References

Chauhan R, Kaur H, Chang V (2017) Advancement and applicability of classifiers for variant exponential model to optimize the accuracy for deep learning. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-017-0561-x

Feng J, Xuefen C, Li P, Yining W, Guan L, Wang X (2010) Adaptive FEC algorithm based on prediction of video quality and bandwidth utilization ratio. J Ambient Intell Humaniz Comput 1:309–318

Huang R, Feng W, Fan M, Guo Q, Sun J (2017) Learning multi-path CNN for mural deterioration detection. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-017-0656-4

Kim E, Kim J, Lee C (2018) Efficient neighbor selection through connection switching for P2P live streaming. J Ambient Intel Humaniz Comput. https://doi.org/10.1007/s12652-018-0691-9

Lee I, Guan L (2012) Inter-frame dependency in multiview multi-description video streaming. J Ambient Intell Humaniz Comput 3:87–94

Liu C, Bouazizi I, Gabbouj M (2011) Rate adaptation for adaptive HTTP streaming. In: Proceedings of the second annual ACM conference on multimedia systems (2011) ACM, pp 169–174

Liu C, Bouazizi I, Hannuksela MM, Gabbouj M (2012) Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network signal processing. Image Commun 27:288–311

Ma LV, Park J, Nam J, Ryu H, Kim J (2017) A fuzzy-based adaptive streaming algorithm for reducing entropy rate of DASH bitrate fluctuation to improve mobile. Qual Serv Entropy 19:477

Miller K, Quacchio E, Gennari G, Wolisz A (2012) Adaptation algorithm for adaptive streaming over HTTP. In: Packet video workshop (PV), 2012 19th international (2012) IEEE, pp 173–178

Mok RK, Luo X, Chan EW, Chang RK (2012) QDASH: a QoE-aware DASH system. In: Proceedings of the 3rd multimedia systems conference (2012) ACM, pp 11–22

Orsini G, Bade D, Lamersdorf W (2018) Generic context adaptation for mobile cloud computing environments. J Ambient Intell Humaniz Comput 9:61–71

Roman R, Lopez J, Mambo M (2018) Mobile edge computing. fog et al.: a survey analysis of security threats challenges Future Gen Comput Syst 78:680–698

Tang J, Deng C, Huang G-B (2016) Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst 27:809–821

Tian G, Liu Y (2012) Towards agile and smooth video adaptation in dynamic HTTP streaming. In: Proceedings of the 8th international conference on emerging networking experiments and technologies (2012) ACM, pp 109–120

Van Ma L, Kim J, Park S, Kim J, Jang J (2016) An efficient Session_Weight load balancing and scheduling methodology for high-quality telehealth care service based on WebRTC. J Supercomput 72:3909–3926

Van Ma L, Park J, Nam J, Jang J, Kim J (2017) An efficient scheduling multimedia transcoding method for DASH streaming in cloud environment. Cluster Comput, pp 1–11

Vergados DJ, Michalas A, Sgora A, Vergados DD (2014a) A control-based algorithm for rate adaption in MPEG-DASH. In: Information, intelligence, systems and applications, IISA 2014, The 5th international conference on (2014) IEEE, pp 438–442

Vergados DJ, Michalas A, Sgora A, Vergados DD (2014b) A fuzzy controller for rate adaptation in MPEG-DASH clients. In: Personal, indoor, and mobile radio communication (PIMRC), 2014 IEEE 25th annual international symposium on (2014) IEEE, pp 2008–2012

Vergados DJ, Michalas A, Sgora A, Vergados DD, Chatzimisios P (2016) Fdash: a fuzzy-based MPEG/DASH adaptation algorithm. IEEE Syst J 10:859–868

Xu X, Liu J, Tao X (2017) Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource. Alloc IEEE Access 5:16406–16415

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.