Quality of Experience Prediction Trends for Live Video Streaming Events

Submitted by:

Arseniou Christos, 02730

Kyriakou Georgia, 02901

Koutsoni Elpida-Myrto, 02900

Nikolaidis Alexandros, 02703

Tozakidis Dimitrios, 02744

Dept. of Electrical Engineering and Computer Science

University of Thessaly (UTH) Volos, Greece

2021 - 2022

ABSTRACT

Keywords -Deep Learning, QoE, buffer-ms, viewer-type , correlation, LSTM, RNN, evaluation

This document contains a summary of what has been implemented and validated both in the first and the second part of the Deep Learning Project ”Quality of Experience Prediction Trends for Live Video Streaming Events”.

Introduction

Live video streaming has become a mainstay as an essential communication solution for large enterprises. Fortune-500 companies organize live video streaming events for several purposes, such as training employees, announcing product releases, and so on. Accounting for the high value of live video streaming services, companies invest a significant amount of their annual budgets. Therefore, large enterprises expect the viewers that participated in the event to receive high quality video, while being fully engaged. However, in the real-world setting there are several factors hindering the video quality and user engagement, for example, the bandwidth limitations when several employees attend an event simultaneously at several offices, the time when the event is scheduled, and so on. In the first part of the project we perform data analysis to extract valuable insights from the dataset. In the second part we implement a deep learning architecture.

QoE trend over time (month) for each customer.

First for each customer we find everyone of his events to categorize them to the months that occurred. Then for each month we make a matrix called “QoE” where we save every QoE of each event of this month. Finally we take the average QoE of each month for each customer just by adding every element of the matrix “QoE” and divide it by its multitude. We selected the first 20 customers for our analysis because they represent the 95% of the data. From the figures that we produced we observe that in the most cases the QoE is high (0.950 - 1.000). We believe that the reason that the observed phenomenon occurs is the viewer’s fast and stable internet connection.

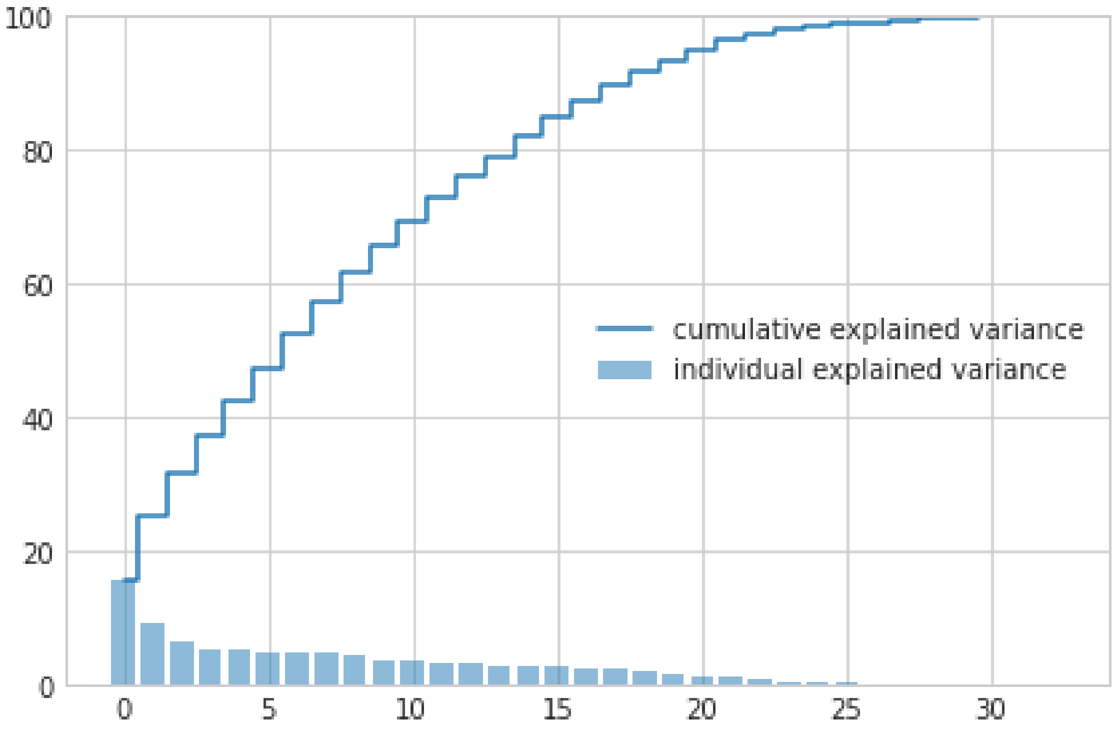

Figure 1: The first 20 customers for our analysis represent the 95% of the data

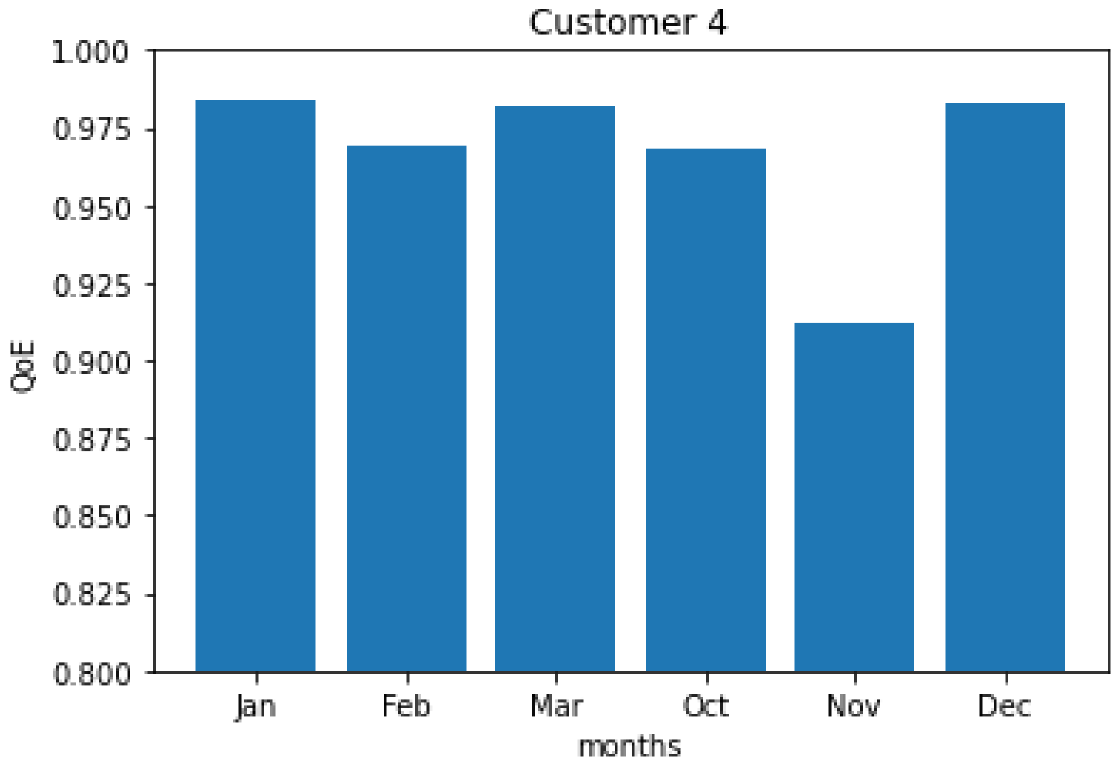

Figure 2: QoE trend over time for customer 4.

Buffering severity over time (month) for each customer.

For the second question we used scikit-learn to discretize ”buffer-ms” to 5 different levels (0 is Excellent, 1 is Good, 2 is Average, 3 is Poor and 4 is Bad). Then we calculate the average buffering severity of each customer for every month. Finally we plot the new discretized values (Buffering severity) over time (month) for each customer. For the x-axis we stored only the months that contain data. From the figures we observed that the value of the Discretize buffer is mainly 0 . That occurs because there were only a few entries with a high buffer-ms.

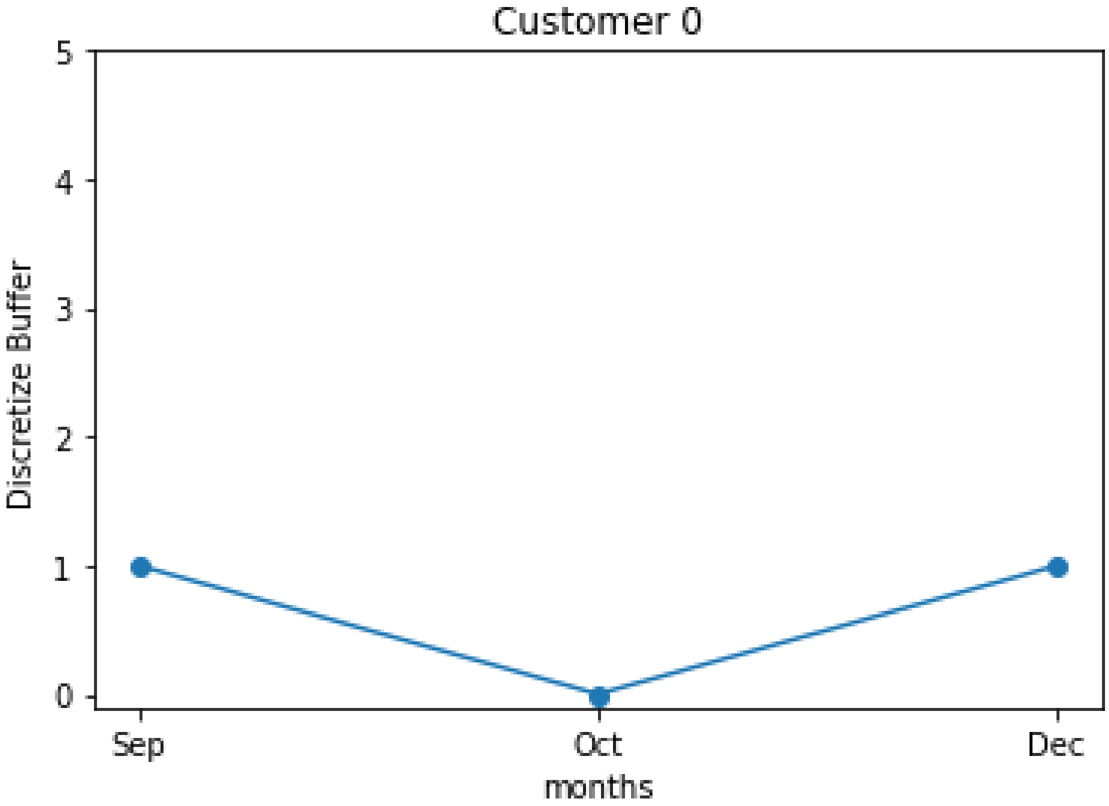

Figure 3: Buffering severity over time for customer 0

Number of viewers over time that experience the QoE level

Similar to above, we discretize the QoE values that we have obtained from the first question to 5 different levels (0 is Bad, 1 is Poor, 2 is Average, 3 is Good and 4 is Excellent). As expected, most viewers experience QoE level 4 (Excellent) which means the most viewers do not experience many broadcast delays, freezes and buffers.

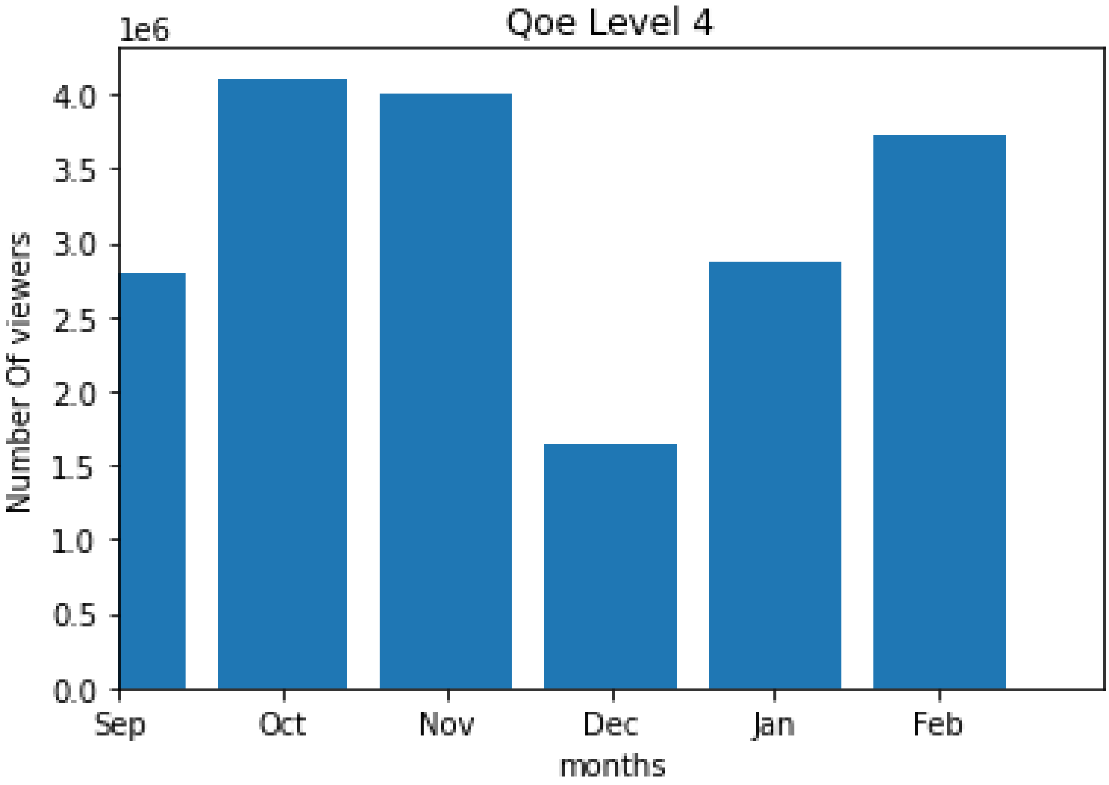

Figure 4: Number of viewers that experience QoE level 4

QoE trend over time for each customer and the viewers' location type

In this task, we plot the QoE trend over time for each customer and the viewers' location type. Green colour represents ”work from office” and the blue colour represents ”work from home”. After we created the figures we observed that there is no relationship between the two variables (”work from office”, ”work from home”). To the figure below we see the viewers' location type of customer 3.

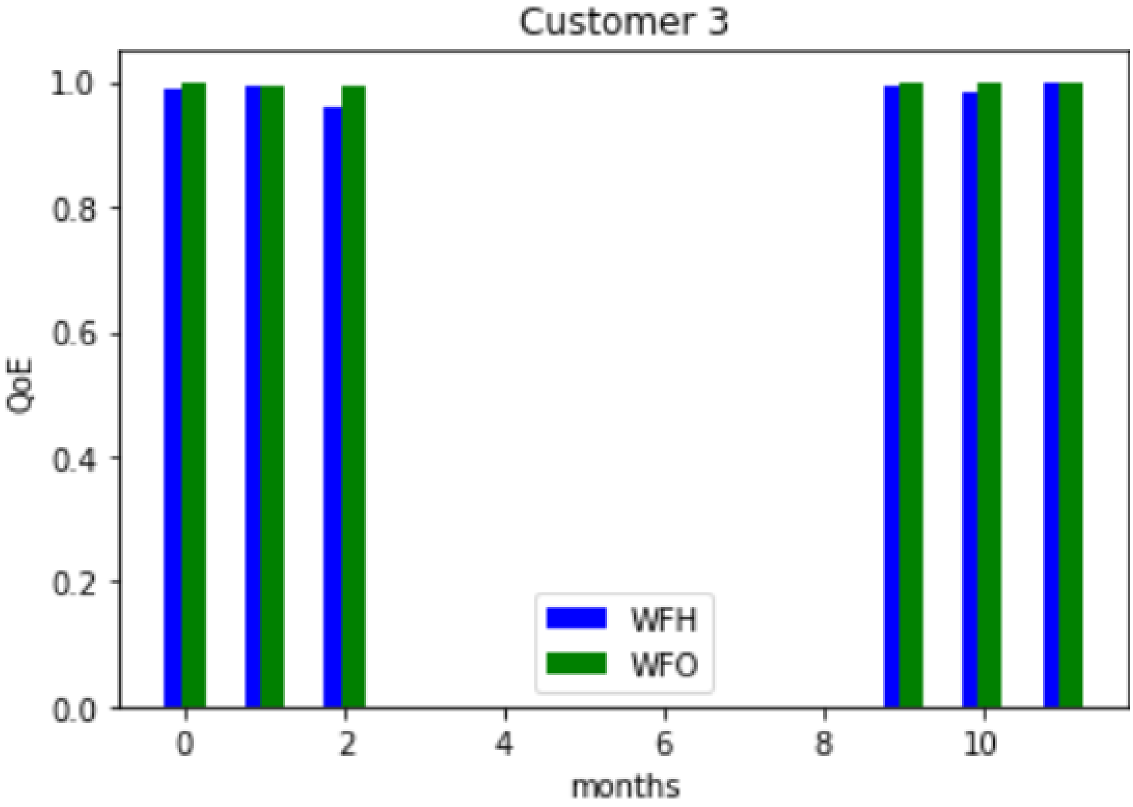

Figure 5: QoE trend over time for each customer and the viewers' location type. The numbers 0 - 11 represent the months January - December

Data points correlation

The Pearson product-moment correlation coefficient (or Pearson correlation coefficient, for short) is a measure of the strength of a linear association between two variables and is denoted by r. Basically, a Pearson product-moment correlation attempts to draw a line of best fit through the data of two variables, and the Pearson correlation coefficient, r, indicates how far away all these data points are to this line of best fit (i.e., how well the data points fit this new model/line of best fit). The Pearson correlation coefficient, r, can take a range of values from +1 to -1. A value of 0 indicates that there is no association between the two variables.

The Spearman's rank-order correlation is the nonparametric version of the Pearson product-moment correlation. Spearman's correlation coefficient, (, also signified by r_s) measures the strength and direction of association between two ranked variables.

Correlations between the data points

- Check for low engagement on Fridays:
  Fridays don't correlate with low viewer engagement score

- Check for correlation between qoe engagement (when qoe=1). We hypothesize that when qoe is max, engagement will be high:
  No obvious correlation between high qoe and high engagement

- Check for high engagement shen location is WFO (Work From Office):
  No obvious correlation between high engagement and WFO

- Check for low engagement shen location is WFH (Work From Home):
  No obvious correlation between low engagement and WFH

- Check for correlation between qoe and buffer_ms:
  Qoe and buffer_ms have almost perfect negative correlation (exact inverse linear relationship), the bigger the buffer_ms is the smaller the qoe is. Which is a logical result, as the more the user experiences buffering the worse the user's experience is.

- Check for correlation between engagement and buffer_ms:

Low buffer_ms doesn't necessarily mean that viewer engagement is high as that is shown in the data, so no obvious correlation here either.

Correlations between QoE and various factors

- Between QoE and number of viewers during the event:
  Pearson correlation coefficient is 0.064 so there is no strong correlation between the number of viewers and the event's QoE.

- Between QoE and day of the event:
  Pearson correlation coefficient is 0.058 so there is no strong correlation between the day the event happens and the event's QoE.

- Between QoE and duration of the event:
  Pearson correlation coefficient is 0.02 so there is no strong correlation between the event's duration and QoE.

- Between QoE and number of countries each customer reach:
  Pearson correlation coefficient is -0.004 so there is no strong correlation between the number of countries that the customer reaches and this customer's events' QoE.

- Between QoE and country:
  Pearson correlation coefficient is -0.458 so there is a fairly strong relationship between the country and the event's QoE. The negative symbol doesn't have any meaning as the countries' IDs don't represent anything.

- Between QoE and Viewers' retention:
  Pearson correlation coefficient is 0.012 so there is no strong correlation between viewers' retention and event's QoE.

Feature Selection

Feature selection is a fundamental step before initializing and training a model, since it is indicates to what extent each variable should participate in the model. It also helps in extracting those features that would lead to overfitting and thus to improve the performance. Among the methods of feature selection are some statistical metrics, such as Pearson correlation coefficient, chi-square test and others.

Based on the analysis of Part 1, we conclude that only few of the features are relevant to feed the model. According to the correlation results driven above, we would have to ignore some of the variables that are highly correlated to each other and therefore have the same effect on the dependent variable, which in this case is the Quality of Experience ('QoE'). The features we eventually used are listed below:

- Customer ('customer_id')
- Sequence of 4 items, the QoE values from the last 4 events in a month for one customer ('features')
- Number of countries that every customer reaches ('Number_of_countries')
- Average event duration ('average_event_duration')

The last two features are not highly correlated and therefore do not appear to improve the performance of the model as much as the other features.

Deep learning architecture and loss function

The Quality of Experience prediction process requires a memory of previous QoE values in certain events that have impact on the future QoE value. This can be accomplished by employing a model based on Long Short-Term Memory (LSTM), a type of Recurrent Neural Network that is proven to be effective for such dependencies either long or short-term. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

The architecture we proposed in this project is an LSTM model with one recurrent layer, input size set to 4*(sequence length) and hyperparameter hidden size equal to 2. This LSTM layer is followed by dropout, a linear layer and a sigmoid layer, in order to reach the vector size we need for the output of the model. A simple Recurrent Neural Network model was also put to the test, however it performed a little worse than the LSTM.

The loss function we use is the Mean Absolute Error, which calculates the average difference between the calculated values and actual values. It is used as evaluation metrics for regression models in machine learning.

Model training

We picked a fixed sequence length equal to 4. We could choose a custom length for each month to month prediction based on the number of events each month has, but this would lead to making new dataloaders for each length, increasing the complexity of the implementation. Therefore, we pad each sequence based on the beginning of it, when a customer doesn't have 4 events in a month. Then, we split the dataset to train, validation and test sets, creating dataloaders for each one of them, with the batch size equal to 1 to use stochastic gradient descent.

Evaluating the performance of the deep learning model

To evaluate the performance of the LSTM deep learning model we picked the mean absolute error metric but the mean squared error works just fine, since we want to calculate and evaluate the difference of the predicted QoE values of the next month events from the actual. Using the MAE metric we get a 0.022 mean difference from the actual output. Otherwise, the MSE metric presents a 0.0014 mean squared difference.

Conclusion

The mean absolute error with the LSTM model that is equal to 0.022, indicates that each prediction falls to the right category, since there are 5 categories (Bad, Poor, Average, Good, Excellent) after the discretization. The same evaluation metric for the RNN model is 0.027 which is slightly worse, but good in general. This happens because the lstm model is more sophisticated since it provides units that include a memory cell that can maintain information in memory for long periods of time. On the other hand, the RNN model is simplier and faster to train. As far as the evaluation metric, we end up choosing the mean absolute error instead of the mean squared error, because it is more immediate since we can direct evaluate our predicted values. Lastly, we picked Stochastic Gradient Descent instead of Gradient Descent to make our model learn a lot faster.