Group Members:
Dan Le, Kaitlyn Tom, Madison Long

Topic: QoE Prediction For Streaming

- **Motivation**

  Video streaming services such as Netflix, YouTube, and Twitch have grown exponentially over the past decade. User experience has become the main metric used to evaluate the performance of these streaming services. Traditional metrics like Quality of Service (QoS) give insight into system performance but do not accurately evaluate user satisfaction. Predicting Quality of Experience (QoE) is important for improving adaptive stream algorithms, optimizing network resource allocation, and identifying when users have poor streaming experiences. QoE prediction on encrypted data packets would bridge the gaps between network metrics and human perception, allowing for more user-centric streaming services.

- **Specifications**

  The goal of this project is to predict a user's QoE during a video streaming session using packet data of SSIM, format, and rebuffering events as input. Our dataset that we obtained from Jaber will have 1000 data points with 30 second windows. The main objective is to develop machine learning models that can accurately calculate or classify for the three main metrics for QoE using raw packet data as input with puffer's logs as a comparative ground truth.

- **Existing Approaches**

  Some existing approaches include the usage of Machine Learning models using Random Forests for prediction of the QoE [1]. They also use chunk-based traffic analysis to extract lightweight features from encrypted flows, enabling real-time inference of video playback metrics such as buffering, video state, and resolution. Tools like NetMicroscope [2] already predict QoE from basic network features, and models like Pensieve [3] use reinforcement learning to optimize streaming QoE by controlling bitrate. Specifically, Pensieve trains a neural network to select bitrates based on real-time network and buffer observations [3]. However, many existing approaches focus on specific platforms or settings and do not always offer generalizable or explainable models.

- **Existing Datasets**

  Some existing datasets that we can work on are from the Puffer project (Stanford's open streaming platform) [4] and netMicroscope (network telemetry datasets for QoE prediction) [2]. Additionally, if we were to use netReplica [5], we can collect or simulate new datasets under varied network conditions to expand our training data.

- **Novelty**

  As we have discussed in class, a common limitation for many network machine learning tends to be a lack of replicability and generalization that occurs once the model leaves the lab specific conditions on which it was trained. Our project intends to address this issue by focusing exclusively on predicting QoE on Stanford’s Puffer project streaming platform. By including a variety of data from varying network conditions, we hope to produce a model which is not truly generalizable but far more robust than existing approaches. Additionally, we will integrate model explainability techniques to increase transparency by potentially utilizing TRUSTEE.

- **Implementation**

  Our approach is to retrieve log data and packet data from Puffer in 30 second increments; specifically we would have 1000 streaming sessions. From there, we will map the packet data to the corresponding log data as a preprocessing step. We will then isolate a 10 second chunk from each of the 30 second chunks in order to train our three separate machine learning models that correspond to the three metrics we are trying to predict i.e. format, SSIM, and rebuffering events. The models we will pick will be dependent on the variables and whichever model has the highest accuracy. Next, we will verify these models do not have any inductive biases with TRUSTEE. Once we establish our models, we will then test them with the remaining preprocessed data; to determine whether our model can predict the logs given the corresponding packets. At the end, we plan to assemble our variables (format, SSIM, and rebuffering events) and create a function to determine the QoE of the streaming. This will conclude as classifications of “excellent”,” good”, “okay” , and “bad” QoE.

References:

[1] Craig Gutterman, Katherine Guo, Sarthak Arora, Xiaoyang Wang, Les Wu, Ethan Katz-Bassett, and Gil Zussman. 2019. Requet: real-time QoE detection for encrypted YouTube traffic. In Proceedings of the 10th ACM Multimedia Systems Conference (MMSys '19). Association for Computing Machinery, New York, NY, USA, 48–59. https://doi.org/10.1145/3304109.3306226

[2] NetMicroscope Team. *NetMicroscope: Real-Time QoE Detection from Encrypted Traffic*. [Online]. Available: https://www.netmicroscope.com/.

[3] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17). Association for Computing Machinery, New York, NY, USA, 197–210. https://doi.org/10.1145/3098822.3098843

[4] Francis Y. Yan, Hudson Ayers (Stanford University), Chenzhi Zhu (Tsinghua University), Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein (Stanford University). *Learning in situ: a randomized experiment in video streaming*. In *Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI '21)*, 705–720. USENIX Association, 2021. https://puffer.stanford.edu/static/puffer/documents/puffer-paper.pdf

[5] NetReplica. *Network Replica: Predicting Quality of Experience from Encrypted Traffic*. GitHub repository. Available at: https://github.com/netreplica.