Adaptive FEC for Cloud Gaming

Abdulmueez Eniola, Ozioma Okonicha, Pavel Tishkin

Keywords

Adaptive Forward Error Correction (FEC), Cloud Gaming, Network Congestion, Quality of Service (QoS), Latency, Packet Loss, Video Streaming, Error Resilience, Error Correction Codes (ECC), Channel Coding, Multimedia Transmission, Real-time Gaming, Network Bandwidth, Network Jitter, Network Delay, Packet retransmission, Redundancy, Scalability, Video Quality, Data Loss Prevention.

Abstract procedure

- ✓ **Data Collection:** In order to train and test our models, we require a sufficient amount of data. Therefore, it is imperative that we carefully collect and curate datasets that are representative and comprehensive for effective machine learning.

- ✓ **Clustering**: Our objective is to train a model that can accurately detect whether the user's network connection is good or bad. Since we do not have any prior information about the clusters within the data, we first need to train a model that is capable of identifying such clusters.

- ✓ **Classification**: Once we have identified the clusters within the data, our next step is to train a model that can accurately predict which cluster any given data point belongs to.

- ✓ **Regression:** In addition, we must train a model that can forecast future data packet loss based on the prior history of data transfers.

- ✓ **Continual learning**: In order to maintain the performance of our models, it's necessary to implement a continual learning paradigm to prevent degradation of accuracy on previously seen data. This involves ongoing training using new data while ensuring that previously learned information is not forgotten.

Paper 1

Title: Deep Time-Series Clustering: A Review

Summary
The paper reviews the current state-of-the-art methods in time series clustering, including manual methods of feature extraction (e.g., principal component analysis) and learning features via deep convolutional autoencoders (DCAE). The authors also review several clustering algorithms, such as KMeans. Their study demonstrates that applying DCAE to feature learning enhances clustering.

Relevance
By utilizing the DCAE model proposed by the authors, we can identify clusters within our data, which can be useful for detecting good or bad network connections.

Paper 2

Title: DeepRS: Deep-learning Based Network-Adaptive FEC for Real-Time Video Communications

Summary
The paper proposes an adaptive Forward Error Correction (FEC) algorithm that adjusts the redundancy ratio of FEC encoders based on the prediction of packet loss. The authors also consider unpredictable network loss patterns and delayed feedback. To evaluate the efficacy of their model, the authors measure the recovery rate and redundancy ratio.

Relevance
The LSTM model proposed by the authors can be used to predict future packet loss during data transfers.

Paper 3

Title: RL-AFEC: Adaptive Forward Error Correction for Real-time Video Communication Based on Reinforcement Learning

Summary

The objective of the authors was to maintain the quality of videos transmitted over unreliable networks with low bandwidth consumption in FEC. So they proposed **RL-AFEC**, an adaptive FEC scheme that learns to *select proper redundancy rates* for all video frames in each Group of Pictures (GoP) using reinforcement learning *(without any domain specific predefined rules)*. Then, the redundant packets of each frame are added according to the selected redundancy rate.

Relevance

The model proposed shows how to make use of agent, states, actions and reward in order to chooses one of the 10 redundancy rates (10%, 20%, . . . , 100%) for each of the K critical frames individually and selects a redundancy rate for all the remaining non-critical frames

4.4 Model's Architecture

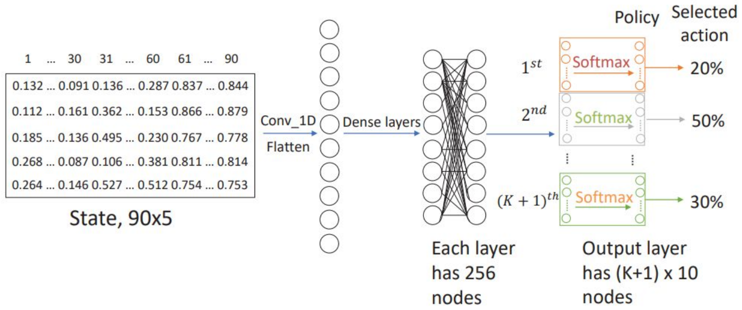


Figure 7: The architecture of RL-AFEC model.

*GF(256) is often used in the context of byte-oriented applications, such as digital communication and data storage, where each element of the field represents a byte (8 bits) of data. In such applications,
*GF(256) is used to encode the data for error correction and detection purposes, so that errors introduced during transmission or storage can be detected and corrected.

Paper 4

Title: LightFEC: Network Adaptive FEC with A Lightweight Deep-Learning Approach

Summary

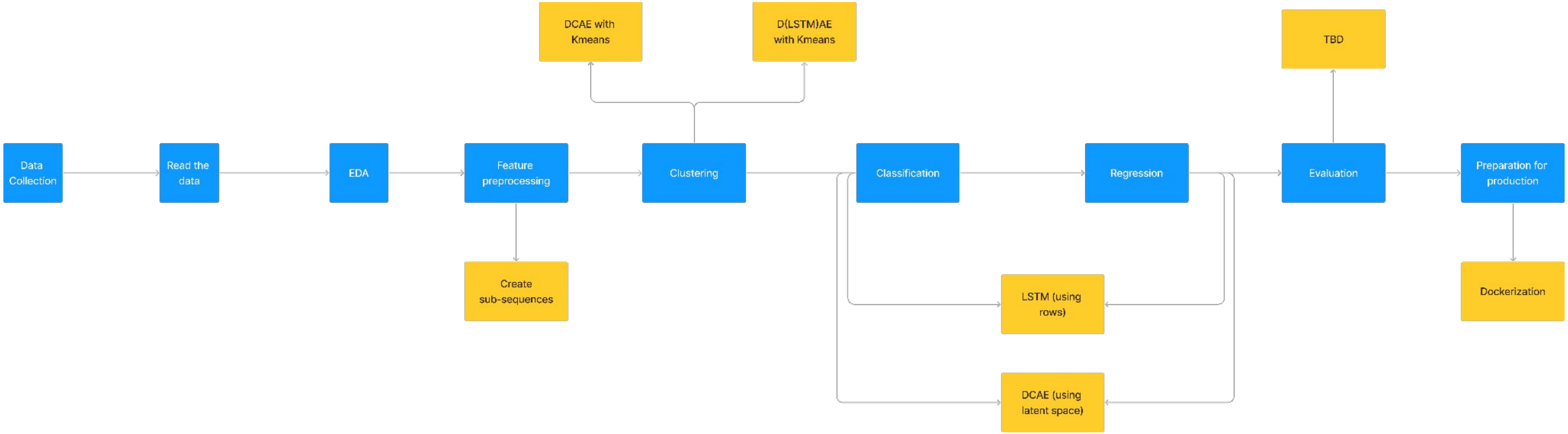
The paper reviews the historical means of Adaptive FEC, including algorithmic and proposes a lightweight solution for packet loss predicting. The authors demonstrate that DBScan classification, combined with multiple LSTMs for each cluster produces favourable results. To achieve time constraints, the researchers use weight pruning

Relevance

DBScan for data clustering to better determine network conditions + several LSTM models for better understanding of appropriate measures

Proposal

Flow diagram



Steps

1. To find the clusters in the dataset, we propose 2 approaches:
   a. Using a DCAE model with KNN to find clusters of the data.
   b. Using a deep lstm autoencoder model with knn .

Our work will evaluate both approaches and select the best approach afterwards.

2. To perform the classification task, We propose to either
   a. Use a many-to-many lstm model to perform data classification, With this approach we won’t need to learn embeddings of our data.
   b. Or perform classification on the embedding of the dataset. If eventually, this approach gives a better result, we will apply a 1 nearest neighbor search on our clusters and hence merge the classification with the clustering.

3. To predict data loss, we plan to explore both
   a. the lstm model
   b. And also learning with embeddings.