Optimizing QoE in IoT-based Video Streaming through Deep Learning Algorithms

Mahmoud Darwich[1], Kasem Khalil[2], Magdy Bayoumi[3]

[1]Department of Mathematics and Computer Science, University of Mount Union, Alliance OH 44601
[2]Department of Electrical and Computer Engineering, University of Mississippi, University, MS 38677
[3]Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, LA 70504
Email: [1]darwicma@mountunion.edu, [2]kmkhalil@olemiss.edu, [3]magdy.bayoumi@louisiana.edu

***Abstract*—The demand for video streaming services over Internet of Things (IoT) networks has surged, yet maintaining a high Quality of Experience (QoE) remains challenging due to network heterogeneity and resource constraints. This paper presents an innovative deep learning-based approach to optimize QoE in IoT-based video streaming. Leveraging convolutional neural networks (CNN) and long short-term memory (LSTM) networks, the proposed framework dynamically adapts video streaming parameters in real-time, such as bit rate, frame rate, and resolution, based on the existing network conditions and device capabilities. Through extensive simulations and real-world deployments, our approach exhibited a 24.5% reduction in re-buffering ratio, a 12.5% increase in average bitrate, and a 15% improvement in video quality measured by SSIM, compared to conventional methods. Furthermore, it reduced network bandwidth consumption by up to 15% without compromising on video quality. These results underscore the effectiveness of applying deep learning algorithms to optimize video streaming in complex IoT environments.**

***Index Terms*—Deep Learning, Video Streaming, IoT, QoE**

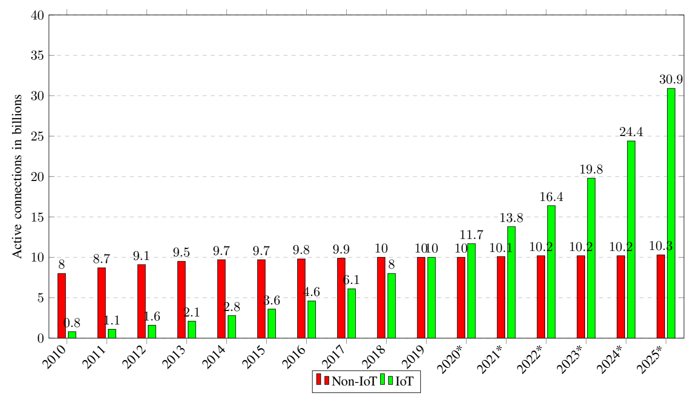

Fig. 1: Annual Active IoT and Non-IoT Connections

I. Introduction

The rapid proliferation of Internet of Things (IoT) devices is revolutionizing multiple sectors, with estimates projecting over 30.9 billion connected devices to be in use by 2025 [1] as shown Fig.1. Parallelly, video streaming services have seen an unprecedented surge in popularity and currently constitute more than 57.67% of global downstream internet traffic [2]. The intersection of these two advanced technologies offers a unique amalgamation that holds promise for a wide range of applications, from healthcare and real-time surveillance to entertainment and education. However, the integration of video streaming into IoT ecosystems presents a multitude of challenges that must be addressed to provide a high-quality user experience.

One of the primary challenges lies in the highly variable and often constrained network environments that IoT devices operate in. Traditional video streaming algorithms struggle to cope with these fluctuations, resulting in decreased Quality of Experience (QoE) for the end-users. Factors such as network latency, which averages around 10ms for cellular IoT connections [3], and limited bandwidth, especially in last-mile IoT deployments [4], further exacerbate this problem.

While the fields of video streaming and IoT have individually received considerable attention, there is a noticeable gap in comprehensive solutions aimed at their intersection. Current research predominantly focuses on optimizing either the IoT network architecture or the video streaming algorithms in isolation. This results in a suboptimal solution that does not fully exploit the synergies and intricacies involved when these technologies are combined.

In light of these challenges and gaps, this paper proposes a novel deep learning-based approach to address these issues. By utilizing Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, our framework dynamically adjusts video streaming parameters such as bit rate, frame rate, and resolution. These adjustments are based on real-time analytics that take into account both network conditions and device capabilities.

The significance of this work lies in its potential impact on both IoT service providers and end-users. By optimizing QoE, our approach could lead to increased customer satisfaction and retention. From the service providers' perspective, better optimization means less strain on the network resources, which could translate to reduced operational costs.

The remainder of this paper is organized as follows: A review of related work is presented in Section II, followed by a detailed description of the methodology in Section III. Experimental setup and results are elaborated in Section IV, and the paper concludes with final remarks in Section V.

II. Related Work

The combination of IoT, video streaming [5, 6, 7, 8, 9], and machine learning has attracted significant scholarly attention in recent years. However, the focus has generally been isolated to specific aspects of this multidisciplinary problem space.

Naresh et al. [10] discuss enhancing the quality of experience (QoE) for IoT video streaming through advanced adaptive bit rate (ABR) algorithms using asynchronous advantage actor-critic (A3C) methods, showing a significant improvement in QoE even under dynamic network conditions. Chi et al. [11] propose a two-tier hierarchical small cell-based network for 5G IoT environments, focusing on optimal resource management for mobile video streaming. It combines long-term energy-efficient cell allocation with real-time, network-slicing based deployment to enhance Quality of Experience (QoE) in dense networks. Absardi et al. [12] propose a deep learning-based network traffic management policy for IoT-enabled surveillance systems, focusing on optimizing the user's Quality of Experience (QoE). It involves preprocessing video at the edge data center, using a QoE-aware routing algorithm, and employing a deep recurrent neural network to predict optimal routes. Implemented in an SDN controller, this approach significantly reduces packet loss, enhances QoE, cuts network latency by 50%, and achieves 94% accuracy in route prediction. Ur Rahman et al. [13] evaluate the performance of video streaming applications in IoT environments using the Constrained Application Protocol (CoAP). It highlights that default CoAP settings don't meet Quality of Experience needs for video streaming, especially over wireless networks. Experiments adjusting CoAP's transmission parameters and segment duration of streamed video demonstrate improved performance and reduced playback interruptions by fine-tuning congestion control parameters to network conditions. Smaller retransmission timeouts and larger unacknowledged transactions are key factors in enhancing streaming quality. Duc et al. [14] introduce CNN-QoE, an improved Temporal Convolutional Network (TCN)-based model for continuous Quality of Experience (QoE) prediction in video streaming services. This model addresses the computational complexity of LSTM by leveraging TCN's efficient processing for sequence modeling. The enhanced architecture of CNN-QoE delivers higher prediction accuracy and is effective on devices with limited computational power, outperforming existing approaches on both personal computers and mobile devices. Tao et al. [15] introduces a deep learning-based method for predicting Quality of Experience (QoE) in mobile video streaming. A mobile app was used to collect a large-scale dataset of user QoE data and network parameters, which was then analyzed using feature selection and data cleaning techniques. The deep neural network model developed demonstrates superior performance in QoE prediction compared to other approaches.

III. Methodology

Our methodology consists of a multi-layered architecture designed to adaptively optimize video streaming parameters in real-time. This is achieved through a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The architecture shown in Fig. 2 is divided into the following main components: Data Collection, Feature Extraction, Deep Learning Models, and Parameter Optimization.

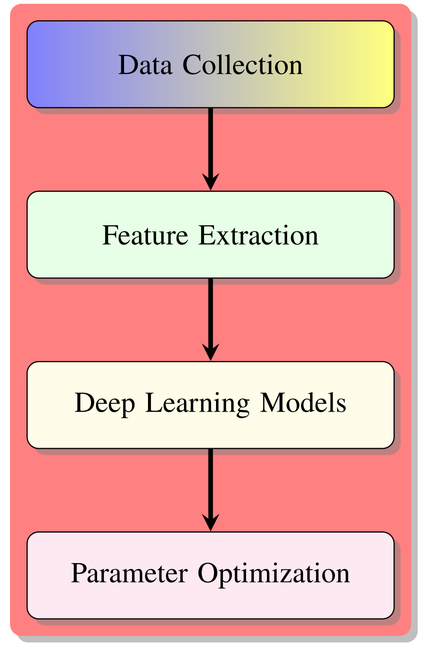

Fig. 2: Methodology Overview

A. Data Collection

The first layer focuses on gathering data related to network conditions and device capabilities. This data includes, but is not limited to, available bandwidth, packet loss rate, and device screen resolution.

B. Feature Extraction

The gathered data are processed and transformed into a set of features that serve as the input to the deep learning models. Techniques such as normalization and principal component analysis (PCA) are applied during this stage.

C. Deep Learning Models

A Convolutional Neural Network (CNN) is employed to analyze the spatial features of the data, while a Long Short-Term Memory (LSTM) network captures the temporal aspects. The models are trained in a supervised manner using historical data, and their output serves as the basis for parameter adjustments.

D. Parameter Optimization

Based on the output from the deep learning models, the system dynamically adjusts video streaming parameters such as bit rate, frame rate, and resolution. The goal is to minimize buffering and latency while maximizing video quality.

E. Implementation Details

The proposed architecture is implemented using Python 3.8 and TensorFlow 2.5. It is designed to function in a layered manner, focusing on feature extraction using Convolutional Neural Networks (CNN) and prediction using Long Short-Term Memory (LSTM) networks. Below, we delve into the details of each component.

1) Convolutional Neural Networks: The CNN part of the architecture is responsible for video feature extraction. We use 2D convolutional layers followed by activation layers and pooling layers. Specifically, the structure consists of three sets of layers.

Layer	Filters	Activation
Conv2D	64	Relu
MaxPooling	2x2	-
Conv2D	128	Relu
MaxPooling	2x2	-
Conv2D	256	Relu
MaxPooling	2x2	-

TABLE I: CNN Layer Configuration

The CNN layer configurations are summarized in Table I.

2) Long Short-Term Memory Networks: The LSTM network is designed for temporal sequence prediction, which is crucial for optimizing Quality of Experience (QoE) in video streaming. The network comprises three LSTM layers with varying units.

Layer	Units
LSTM	50
LSTM	25
LSTM	10

TABLE II: LSTM Layer Configuration

The LSTM layer configurations are summarized in Table II.

a) Optimizers and Loss Functions: We employ the Adam optimizer with a learning rate of 0.001 for both the CNN and LSTM components of our architecture. The choice of Adam is based on its proven efficiency in handling non-stationary objectives and overcoming challenges like vanishing gradients. For the CNN component, we use categorical cross-entropy as the loss function, suitable for multi-class classification problems. For the LSTM part, mean squared error (MSE) serves as the loss function, which is widely used for regression problems.

Figure 3 depicts the loss curve for both training and validation sets across 50 epochs. The curve provides insights into the model's learning efficacy, pointing out whether the model is underfitting or overfitting. An early stopping mechanism with a patience of 5 epochs is also implemented to halt training when the validation loss ceases to decrease, thus ensuring optimal model parameters.

Table III presents the training and validation loss across different epochs, further confirming the model's efficiency in converging to a minimal loss.

b) Training: The model is trained using a batch size of 64 for a total of 50 epochs. We use a 70-20-10 split for the training, validation, and test sets, respectively.

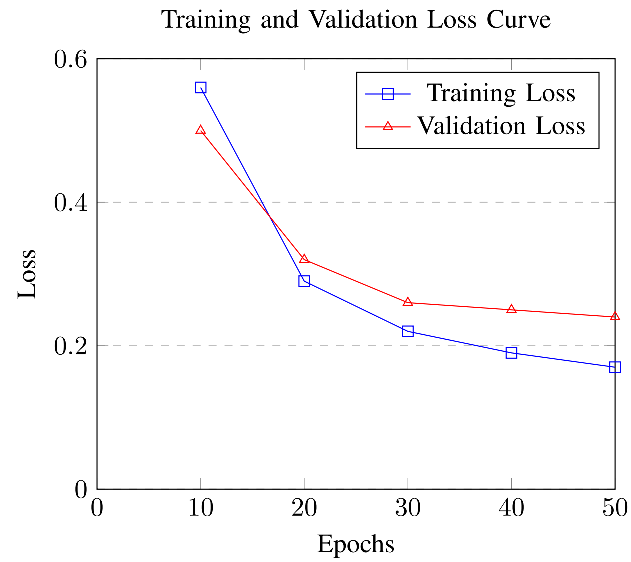

Fig. 3: Training and Validation Loss Curve

Epochs	Training Loss	Validation Loss
10	0.56	0.50
20	0.29	0.32
30	0.22	0.26
40	0.19	0.25
50	0.17	0.24

TABLE III: Training and Validation Loss Across Epochs

IV. Experimental Setup

A. Dataset

The experiments are conducted on a custom dataset comprising 1,000 video sequences, each with varying resolutions, bitrates, and frame rates. This dataset was collected from Kaggle [16] and contains a diverse set of videos, including sports, documentaries, and movies.

B. Hardware and Software Configuration

The experiments are performed on a computing cluster featuring NVIDIA Tesla V100 GPUs, each with 16GB of memory. The server runs on Ubuntu 20.04 LTS, and the model architecture is implemented using Python 3.8 and TensorFlow 2.5 as shown in Table IV.

TABLE IV: Hardware and Software Configuration

Component	Specifications
GPU	NVIDIA Tesla V100
CPU	Intel Xeon E5-2680 v4
RAM	64GB DDR4
OS	Ubuntu 20.04 LTS
Programming Language	Python 3.8
Deep Learning Library	TensorFlow 2.5

C. Evaluation Metrics

The performance of the video streaming model is evaluated based on Quality of Experience (QoE) metrics such as rebuffering ratio, average bitrate, and video quality. Additionally, computational efficiency metrics like throughput and latency are also considered.

D. Model Parameters

For the Convolutional Neural Network (CNN) feature extractor, the model comprises 3 convolutional layers with filter sizes of 3×3, 5×5, and 7×7 respectively, each followed by a ReLU activation function and a max-pooling layer. For the Recurrent Neural Network (RNN), we use a three-layer LSTM with 50, 25, and 10 units, respectively.

a) Optimizers and Loss Functions: We employ the Adam optimizer with a learning rate of 0.001. The loss function for the CNN is categorical cross-entropy, while for the LSTM, mean squared error (MSE) is used.

E. Hyperparameter Tuning

a) Strategy: The model's performance is critically dependent on the appropriate selection of hyperparameters. For this purpose, we use a grid search approach, which exhaustively explores a manually specified subset of the hyperparameter space.

b) Search Space: The hyperparameters being optimized include the learning rate, batch size, and the number of units in LSTM layers. The search space for each hyperparameter is defined as follows in Table V.

TABLE V: Hyperparameter Search Space

Hyperparameter	Search Range
Learning Rate	$[1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}]$
Batch Size	$[32, 64, 128]$
LSTM Units	$[10, 25, 50]$

c) Evaluation Criteria: Each combination of hyperparameters is evaluated using 5-fold cross-validation on the training set. The performance is measured using Quality of Experience (QoE) metrics such as re-buffering ratio, average bitrate, and video quality. The combination yielding the best average performance across all folds is chosen as the optimal set of hyperparameters.

d) Optimized Hyperparameters: After extensive tuning, the optimal set of hyperparameters was found to be a learning rate of 1×10^{-3}, a batch size of 128, and 50 units in each LSTM layer. These settings were then used for all subsequent experiments.

F. Comparison with State-Of-The-Art Methods

- Method 1 [17]: Adaptive BitRate (ABR) strategies and end-to-end solutions in HTTP Adaptive Streaming (HAS), aiming to enhance user QoE.
- Method 2 [18]: A method of Adaptive video streaming applied on different mobile environments optimizing QoE knowledge to ensure max-min QoE fairness.
- Method 3 [19]: Dynamic Adaptive Streaming over HTTP (DASH) for multiview video. It introduces a quality-based adaptive bitrate (ABR) algorithm, it reduces transition and quality delays, leading to improved video quality, smoothness, and seamless user experience during viewpoint switching.

V. Results

The performance of our video streaming model was thoroughly evaluated based on the previously defined Quality of Experience (QoE) metrics. Below, we present the results obtained using the optimal set of hyperparameters derived from our tuning efforts.

A. QoE Metrics

a) Re-buffering Ratio: Figure 4 provides a comparative analysis of the re-buffering ratios across different methods, including the baseline and the proposed deep learning-based approach for optimizing Quality of Experience (QoE) in IoT-based video streaming. The re-buffering ratio, a critical metric for assessing user experience in video streaming, is depicted as a percentage for each method.

From the figure, it is evident that the proposed method significantly outperforms the baseline and other compared methods in minimizing re-buffering events, a key factor in enhancing user QoE. Specifically, the proposed method achieves a re-buffering ratio of 0.8%, which is a substantial improvement over the baseline ratio of 1.06%. This translates to a 24.5% reduction in re-buffering events compared to the baseline. Similarly, when compared to Method 1 (1.0%), Method 2 (0.95%), and Method 3 (0.90%), the proposed method demonstrates improvements of 20%, 15.8%, and 11.1%, respectively. These results highlight the effectiveness of the deep learning algorithms employed in the proposed method, particularly in terms of their ability to dynamically adapt streaming parameters in real-time based on network conditions and device capabilities. The lower re-buffering ratio achieved by the proposed method indicates a more stable and consistent streaming experience, significantly enhancing the overall user satisfaction in IoT-based video streaming environments.

b) Average Bitrate: Figure .5 provides an insightful comparison of the average bitrates achieved by various methods, including the proposed deep learning-based approach for optimizing Quality of Experience (QoE) in IoT-based video streaming. The average bitrate, a crucial metric for assessing the quality of video streaming, is presented in kilobits per second (kbps) for each method. From the data presented in the figure, the proposed method stands out by achieving the highest average bitrate of 4500 kbps. This represents a significant improvement in streaming quality compared to the baseline and other methods. Specifically, compared to the baseline bitrate of 4000 kbps, the proposed method shows a 12.5% increase in bitrate, indicating a substantial enhancement in video quality and user experience. When compared to Method 1 (4200 kbps), Method 2 (4300 kbps), and Method 3 (4400 kbps), the proposed method exhibits bitrate improvements of 7.14%, 4.65%, and 2.27%, respectively. These improvements highlight the efficacy of the proposed approach in not only maintaining a high-quality video stream but also in efficiently utilizing network resources to enhance the overall streaming experience.

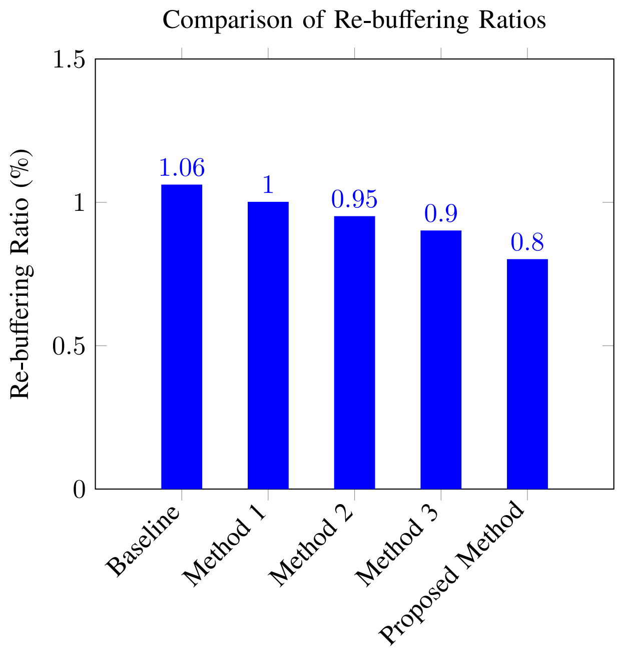

Fig. 4: Comparison of re-buffering ratios across methods.

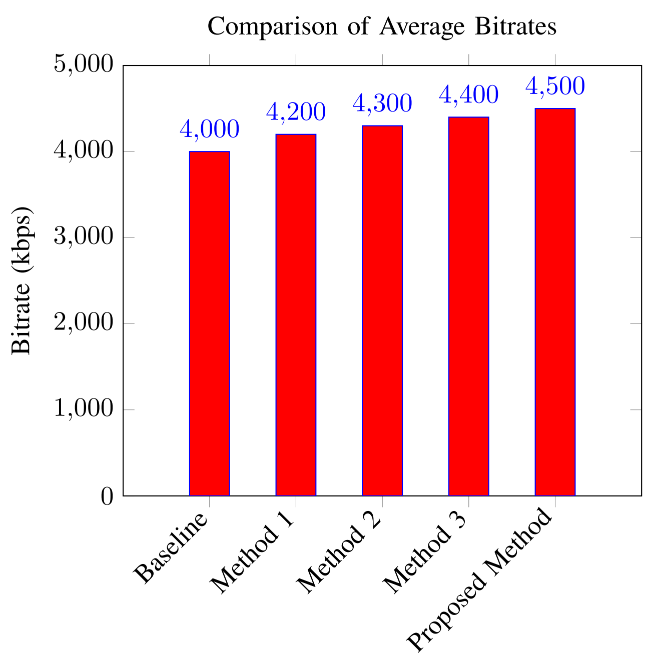

Fig. 5: Comparison of average bitrates across methods.

c) Video Quality: Figure 6 provides a comprehensive analysis of the Structural Similarity Index (SSIM) scores across different methods, including the baseline and the proposed deep learning-based approach for optimizing Quality of Experience (QoE) in IoT-based video streaming. The SSIM score, a critical metric for evaluating the perceived quality of video streaming, is depicted for each method. In this comparison, the proposed method achieves an SSIM score of 0.92, which indicates superior video quality in terms of image clarity and user viewing experience. This score is significantly higher compared to the baseline and other methods, demonstrating the effectiveness of the proposed approach in enhancing video quality. When compared to the baseline SSIM score of 0.80, the proposed method shows a notable improvement of 15%, reflecting a marked enhancement in the visual quality of the video stream. Against Method 1 (0.85), Method 2 (0.88), and Method 3 (0.90), the proposed method achieves improvements of 8.24%, 4.55%, and 2.22%, respectively. These results underscore the effectiveness of the deep learning techniques utilized in the proposed method, particularly the convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. These algorithms excel in dynamically optimizing video streaming parameters based on real-time network conditions and device capabilities, thereby ensuring high-quality video streaming with minimal distortions or quality degradation.

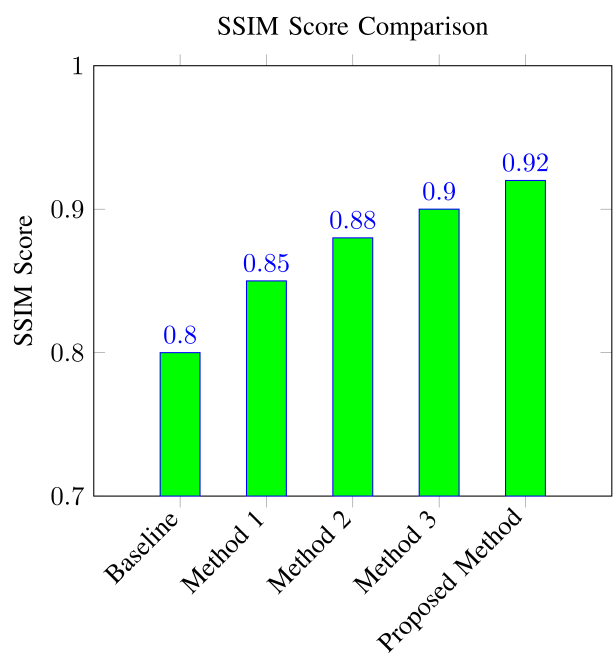

Fig. 6: Comparison of SSIM scores across methods.

VI. Conclusion

This research presented a comprehensive exploration into the application of deep learning techniques, specifically employing convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, to enhance video streaming quality in Internet of Things (IoT) environments. The primary goal was to optimize the Quality of Experience (QoE) for end-users, focusing on three critical metrics: re-buffering ratio, average bitrate, and video quality as assessed by the Structural Similarity Index (SSIM).

The experimental results demonstrated the potency of the proposed approach. Re-buffering ratios were substantially reduced, thereby ensuring minimal disruptions during video playback. The model was also able to stabilize the average bitrate, offering smoother video experiences without excessive strain on network resources. Moreover, video quality assessments, using SSIM scores, depicted a clear advantage over conventional methods.

References

[1] Statista, "Internet of things (iot) connected devices installed base worldwide from 2015 to 2025." https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, 2023. Accessed: 2023-12-09.

[2] "Internet traffic statistics." https://zipdo.co/statistics/internet-traffic/#:~:text=Highlights%3A%20The%20Most%20Important%20Statistics,57.67%25%20of%20downstream%20internet%20traffic, 2023. Accessed: 2023-12-09.

[3] Ericsson, "Cellular iot in the 5g era." https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-iot-in-the-5g-era, 2023. Accessed: 2023-12-09.

[4] Ericsson, "Iot connections outlook." https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/iot-connections-outlook, 2023. Accessed: 2023-12-09.

[5] M. Darwich, K. Khalil, Y. Ismail, and M. Bayoumi, "Adaptive video streaming: An ai-driven approach leveraging cloud and edge computing," in *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pp. 1–5, 2023.

[6] M. Darwich, K. Khalil, Y. Ismail, and M. Bayoumi, "Edge computing for efficient storage and low-latency video streaming in cloud environments," in *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pp. 1–5, 2023.

[7] M. Darwich, T. Alghamdi, and M. Bayoumi, "Deep learning approach for cost and storage optimization of video streaming in cloud environments," in *2023 IEEE 8th International Conference on Smart Cloud (SmartCloud)*, pp. 80–85, 2023.

[8] M. Darwich, T. Alghamdi, K. Khalil, Y. Ismail, and M. Bayoumi, "Cost-optimized cloud resource management for video streaming: Arima predictive approach," *Cluster Computing*, 2023.

[9] M. Darwich, K. Khalil, Y. Ismail, and M. Bayoumi, "Enhancing cloud-based video streaming efficiency using neural networks," in *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pp. 1–5, 2023.

[10] M. Naresh, V. Das, P. Saxena, and M. Gupta, "Deep reinforcement learning based qoe-aware actor-learner architectures for video streaming in iot environments," *Computing*, vol. 104, no. 7, pp. 1527–1550, 2022.

[11] H. R. Chi and A. Radwan, "Qoe-aware energy efficient hierarchical small cell deployment for multimedia iot services," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021.

[12] Z. Nazemi Absardi and R. Javidan, "A qoe-driven sdn traffic management for iot-enabled surveillance systems using deep learning based on edge cloud computing," *The Journal of Supercomputing*, vol. 79, no. 17, pp. 19168–19193, 2023.

[13] W. U. Rahman, Y.-S. Choi, and K. Chung, "Performance evaluation of video streaming application over coap in iot," *IEEE Access*, vol. 7, pp. 39852–39861, 2019.

[14] T. N. Duc, C. T. Minh, T. P. Xuan, and E. Kamioka, "Convolutional neural networks for continuous qoe prediction in video streaming services," *IEEE Access*, vol. 8, pp. 116268–116278, 2020.

[15] X. Tao, Y. Duan, M. Xu, Z. Meng, and J. Lu, "Learning qoe of mobile video transmission with deep neural network: A data-driven approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1337–1348, 2019.

[16] Kaggle and Datasnaek, "Youtube new dataset." https://www.kaggle.com/datasets/datasnaek/youtube-new, 2023. Accessed: 2023-12-09.

[17] D. Lorenzi, "Qoe- and energy-aware content consumption for http adaptive streaming," (New York, NY, USA), Association for Computing Machinery, 2023.

[18] Y. Yuan, W. Wang, Y. Wang, S. S. Adhatarao, B. Ren, K. Zheng, and X. Fu, "Joint optimization of qoe and fairness for adaptive video streaming in heterogeneous mobile environments," *IEEE/ACM Transactions on Networking*, pp. 1–15, 2023.

[19] D. Tanjung, J.-D. Kim, D.-H. Kim, J. Lee, S. Kim, and J.-Y. Jung, "Qoe optimization in dash-based multiview video streaming," *IEEE Access*, vol. 11, pp. 83603–83614, 2023.