Energy Demand as AI Model Selection Criteria? Assessing Quality and Energy Consumption for AI based KQI Prediction in Video Streaming

Frank Loh*, Carina Baur*, Flavian Raithel*, David Stüber*, Omran Ayoub†, Tobias Hoßfeld*
*University of Würzburg, Institute of Computer Science, Würzburg, Germany
†University of Applied Sciences and Arts of Southern Switzerland, Lugano, Switzerland
Contact: frank.loh@uni-wuerzburg.de

***Abstract*—Video-based communication is central to today's digital society. While YouTube and Netflix once dominated video traffic, traditional broadcasters now run their own streaming services, and live-streaming platforms are expanding. Users expect high-resolution video with minimal buffering and latency, especially for live content. Meeting these demands requires infrastructure that balances performance, cost, energy, and resources, supported by comprehensive traffic monitoring and analysis. Artificial Intelligence plays a key role, particularly in encrypted traffic classification and quality prediction, with tree-based Machine Learning models like Random Forests (RF) widely used. However, research often emphasizes small accuracy gains while overlooking the raising energy and resource costs of such models, an increasing conflict with sustainability goals. To study this trade-off, we implement RF models with varying feature counts and complexities to predict video re-buffering and buffer health, two key quality indicators. Using a dataset of more than 11,000 YouTube sessions, we analyze how prediction performance scales with model complexity and energy consumption across the full pipeline, revealing several unexpected results.**

***Index Terms*—quality prediction, AI, ML, energy consumption, streaming quality**

I. Introduction

Video-based communication is central to modern applications. In addition to YouTube and Netflix once dominated the streaming landscape, services such as Disney+, Amazon Prime Video, or Twitch have greatly expanded their services. Video now accounts for a major share of global Internet traffic, with large-scale events, especially live sports, driving peak loads in Europe and the U.S. [1]. From the user perspective, high-quality streaming requires high bitrate, few quality shifts, and uninterrupted playback [2]. Ensuring this quality during peak demand requires substantial infrastructure and often over-provisioning, while insufficient capacity degrades QoE and increases user churn. Although clients select video quality, network and service providers remain responsible for transmission, monitoring, and assurance, which demand significant computing and energy resources. Proactive traffic monitoring and timely QoE issue detection enable efficient resource allocation but further increase energy consumption.

Artificial Intelligence (AI) has become a key tool for network monitoring and management. However, concerns arise about its sustainability, as even trivial tasks (e.g., using ChatGPT for simple queries) consume notable energy. While some AI models clearly outperform analytical methods, others yield only marginal gains for specific inputs, often with unknown energy impact. Despite growing focus on sustainable AI, many developers still neglect or do not know energy demands and environmental impact during training and inference.

To better understand the resource implications of AI-based approaches for predicting video streaming QoE degradation, we focus on well-established Random Forest (RF) models in a domain generating most network traffic. Our analysis explores the energy and resource footprint of monitoring and prediction systems, largely underexplored, through a threefold approach: (1) identifying key procedures in Key Quality Indicator (KQI) prediction driving energy and resource use, (2) quantifying energy consumption across procedures from traffic monitoring to inference, and (3) evaluating trade-offs between AI model complexity and energy usage. We consider two critical KQIs for analyzing video traffic: I) *stalling* as the primary degradation factor and II) *buffer health* as an indicator of stream stability. Our study uses a large-scale dataset of over 11,000 mobile YouTube video sessions [3]. We derive the following three research questions (RQs).

RQ1: Which procedures of video streaming KQI monitoring and prediction are most energy-intensive, and where are potential opportunities for savings?

RQ2: Can we identify differences in energy consumption characteristics for different KQI prediction approaches, i.e., buffer health prediction as regression and stalling prediction as classification task?

RQ3: What trade-offs exist between prediction quality and energy consumption in AI-based KQI prediction for video streaming, and how might these impact future model design?

In Section II, we summarize background and related work, and in Section III, we discuss all relevant procedures for KQI prediction in video streaming and highlight our resource and energy measurement approach. Afterwards, we evaluate and discuss our results in Section IV, and conclude in Section V.

II. Background and Related Work

This section provides relevant background and literature on video streaming, quality assessment, and network monitoring.

Video Streaming: Video streaming is the process of simultaneously requesting and delivering video and audio

978-3-903176-75-1

content from, e.g., a Content Delivery Network (CDN), to the user on a best-effort basis. While audio is typically encoded at a constant bitrate [4], video is offered in multiple quality levels, ranging from 144p to 1080p on mobile [5], and up to 4K or higher on other devices. Each uplink packet requests a video segment, often called a chunk or group of pictures (GOP), which is then delivered via the downlink [5]. Chunk sizes depend on the streaming type and platform. For example, Twitch.tv live streams request up to 2 s per uplink [6], while YouTube requests range from 5 s to 11 s [4, 7]. Consequently, downlink traffic exceeds uplink volume by far.

Quality Assessment: For an end user, a high-quality stream matters more than traffic statistics. KQIs include resolution changes, startup delays, and playback interruptions, called stallings [2]. Other factors like video blur or artifacts are explored in [8]. More recently, studies have expanded KQI considerations to include energy use [9] and carbon footprint [10], highlighting the environmental impact of streaming but balancing quality and sustainability remains a challenge.

Network Monitoring: While prior research has examined the streaming process and its resource needs, the energy and resource demands of monitoring remain often overlooked. Monitoring is essential for detecting and predicting KQIs [11], yet encryption and growing traffic volumes make this increasingly challenging. Existing methods use full packet traces [12, 13] or focus solely on uplink requests [7, 5, 14], with varying accuracy. A recent work highlights differences in data needs [15] but general resource and energy requirements of monitoring different streams and apply various KQI prediction models are still not well researched.

KQI prediction approaches range from statistical models [5] to simple (e.g., RF) or complex AI and deep learning methods [12, 16, 17, 18]. To assess energy requirements for these models and close this gap in the literature, we implement and compare RF models, generally seen as lightweight and thus, common in KQI prediction [12, 16, 13], with varying model complexity and feature sets. Using a large dataset from [3], we analyze and identify trade-offs between model complexity, prediction quality, and energy consumption.

III. Key Quality Indicator Prediction Procedures

A typical KQI prediction workflow for video streaming includes traffic capturing, data processing, feature extraction, model training, and inference. Each step impacts resource usage, especially energy consumption. We detail these impacts and our measurement methods below.

A. Traffic Capturing and Dataset

Data acquisition usually includes the capturing of raw network traffic, the filtering of video traffic, and the preprocessing of the video data to set up a comprehensive dataset.

Traffic Capturing: To examine resource demands for traffic capturing, we set up a testbed with two systems connected via CAT 6 LAN. System A (AMD Ryzen 5 PRO 5650G, 4 GB RAM) sends packets using Iperf to System B (Intel® Core™ i7-4770, 16 GB RAM), emulating video streaming. System B captures traffic with tshark, while CPU usage of the capturing process is measured. We focus on CPU utilization, impacting energy use more than RAM [19], and since our tests show only minimal variation in RAM usage.

Table I: Feature scenarios based on window lengths and traffic direction with name and number (No.) of included features.

	up- and downlink		**uplink only**	
window lengths	name	No.	name	No.
1s, 2s, 3s, 5s, 10s, 20s	all_long	228	uplink_only_long	114
1s, 2s, 3s, 5s, 10s	all_medium	190	uplink_only_medium	95
1s, 2s, 3s	all_short	152	uplink_only_short	76

Streaming Dataset: To predict video streaming KQIs, we use a public dataset of over 11,000 YouTube mobile streams under varying network conditions [3]. The dataset includes application, transport, and network layer data and video qualities from 144p to 1080p. For our analysis, we use timestamps, buffer health, and stalling flags from the application data and timestamps, packet direction (uplink/downlink), and packet size, discarding encrypted payloads from the network data. In total, about 373 million network traffic and 4.6 million application data samples are used as input for our models.

B. Feature Set and Scenarios

Next, features are extracted from the captured downlink and/or uplink traffic during feature selection.

Feature Selection: We evaluate up to 228 features as the most comprehensive feature set proposed by [13]. Similar studies [12, 20] use up to 200 and similar features, so similar analyses would likely yield comparable results. The literature [13, 12] use different window lengths including network traffic from a different duration. In total, features are computed from pre-processed traffic for uplink and downlink over window lengths of 1 s, 2 s, 3 s, 5 s, 10 s, and 20 s. For each window length, we calculate throughput, active time (defined as inter-arrival times under 100 ms), packet count, packet size, and inter-arrival times. Packet size and inter-arrival times are aggregated by mean, median, max, min, and standard deviation. Packet count and size are also computed separately for packets larger than 100 B, representing actual payload. For an overview, we refer to [13].

Feature Scenarios: We examine different feature scenarios based on traffic direction, uplink only and both uplink and downlink, and on window length, as in the literature [12, 13, 7, 14, 16]. We group window length in three categories: (1) Short including 1 s, 2 s, and 3 s windows; (2) Medium, adding 5 s and 10 s windows; and (3) Long, adding 20 s windows. This results in six different scenarios in Table I.

Feature Importance Based Reduction: To reduce the feature set space, we apply a selection criterion based on feature importance, which quantifies each feature's contribution to predictive performance. In RF, importance is computed by averaging the decrease in impurity (e.g., Gini or entropy) each feature provides across all trees, with scores normalized between 0 and 1. The RF model is first trained on the full feature set, and the resulting scores guide the selection of

Table II: Required number of features after feature importance based reduction for buffer health (regression task) and stalling prediction (classification task) using different models.

	all features	buffer health reduced	stalling reduced
all_long	228	10 (~ 4 %)	37 (~ 16 %)
all_medium	190	27 (~ 14 %)	39 (~ 21 %)
all_short	152	25 (~ 16 %)	37 (~ 24 %)
uplink_only_long	114	12 (~ 11 %)	40 (~ 35 %)
uplink_only_medium	95	24 (~ 25 %)	46 (~ 48 %)
uplink_only_short	76	33 (~ 43 %)	51 (~ 67 %)

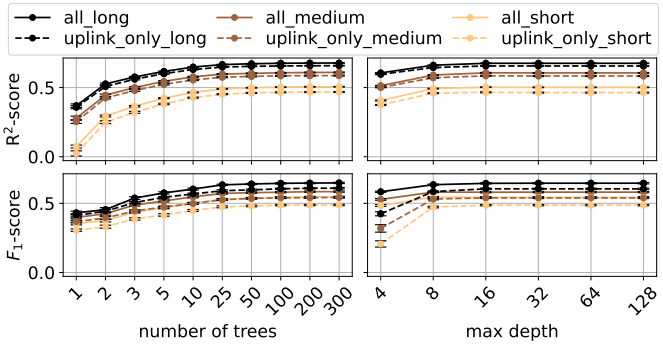

Figure 1: Impact of different number of decision trees (left) and maximal depth of trees (right) on model performance (R^2 for buffer health regressor and F_1 for stalling classifier).

relevant features. Table II shows the number of features per scenario after removing those with importance below 0.01. Original counts are listed alongside reduced counts for buffer health regression and stalling classification, with percentages in brackets. Scenarios with more initial features see the largest reductions, as many contribute little to prediction. In some cases, reduced counts for high-dimensional scenarios are even lower than those for smaller ones, suggesting more redundancy. Across scenarios, inter-arrival-time features are consistently important, especially for stalling. In *all_long*, features from 10 s and 20 s windows dominate, while in *uplink_only_short*, inter-arrival times remain most relevant, with longer windows slightly more important. Buffer health regression often depends heavily on a few dominant features—for instance, maximum inter-arrival times (uplink/downlink) in *all_long* account for nearly 80 % of total importance, and active time percentage in a 5 s window explains 30 % in *uplink_only_short*. This indicates future models may perform well with even fewer features. In contrast, stalling classification shows a more balanced distribution, with the top ten features ranging from 3 % to just over 10 %.

C. Model Training

To assess the energy impact of model training, we consider the resource demands of training RF models, which build multiple decision trees to predict buffer health (regression) or stalling events (classification). Model complexity, and thus training load, depends on hyperparameters such as the number and depth of trees. The default configuration, based on scikit-learn, uses 100 trees with a maximum depth of 16 and serves as a reference. During the study, one parameter is varied at a time, using 5,000 training samples to balance speed and baseline performance. Model performance is measured via R^2 for regression and F_1 for classification. Figure 1 shows results for tree counts (1–300) and depths (4–128) using the base feature sets, without feature reduction, with uplink-only setups shown as dashed lines. Each configuration is evaluated across all feature scenarios, averaged over 30 runs with 95 % confidence intervals to account for randomness in training. Performance improves markedly up to eight nodes in depth and about 25 trees, with diminishing returns beyond that. For stalling classification, uplink-only scenarios show larger performance drops between four and eight tree depths compared to bidirectional features, likely due to limited decision capacity: at four nodes, each tree can make only four decisions, insufficient for accurate classification. Based on these findings, we define three complexity classes:

1) *High* – 100 trees, max depth 16,
2) *Medium* – 50 trees, max depth 8,
3) *Low* – 10 trees, max depth 4.

While more data and hyperparameter tuning could improve accuracy, we aim on assessing trade-offs between prediction quality and energy demand and do not optimize the model.

D. Inference

Finally, the preprocessed data is fed into the model. Inference generally consumes less energy than training, but is run continuously, in our case once per second. While a single inference is low-cost, the cumulative energy demand becomes substantial across many devices and streams, especially at high frequency. In large-scale deployments, this continuous, distributed workload can lead to significant overall energy use and operational costs.

E. Energy Consumption Measurements

Energy consumption is evaluated using PyRAPL [21], which retrieves CPU and RAM energy usage via hardware signals, collecting energy data during script execution. Tests are run on an Intel Core i7-7700 CPU with 32 GB RAM. Each measurement is repeated 30 times to achieve statistically significant results. Models are implemented in Python 3.12 using scikit-learn and pandas, with code available on Github [1]

IV. Evaluation

This section examines RF-based KQI prediction models for video streaming, varying in complexity and feature sets with focus on energy consumption and prediction accuracy. We start with traffic capturing, then assess energy usage for data processing, feature extraction, training, and inference. Finally, we analyze performance to energy consumption trade-offs and outline and discuss practical implications.

[1] https://github.com/lsinfo3/AI-energy-measurements

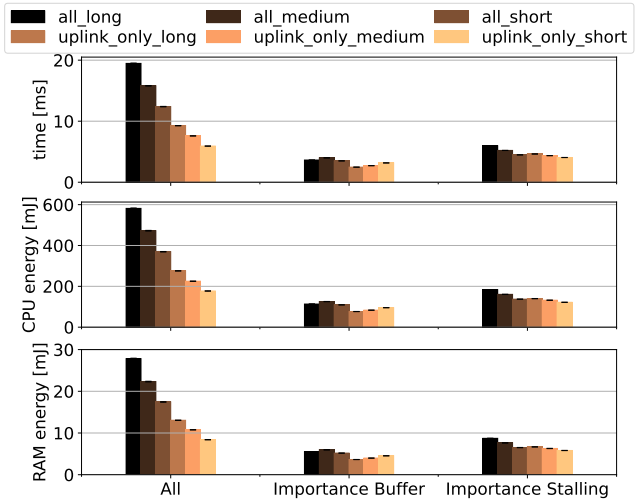

Figure 2: Time (top), CPU (middle), and RAM energy (bottom) to extract a sample from raw data for all features and feature importance based reduced scenarios.

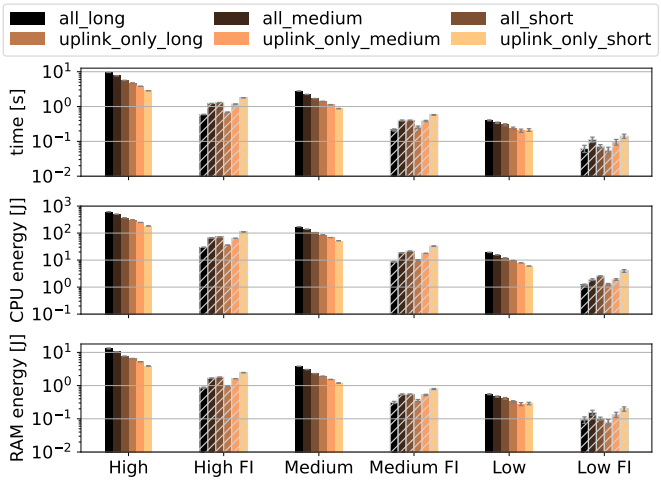

Figure 3: Time, CPU, and RAM energy consumption to train one RF model at high, medium, and low complexity. Using features of six base scenarios with (FI) and without feature importance-based reduction.

A. Packet Capturing

To assess the requirements for capturing video traffic, we generate and transmit data matching bitrates from 100 kbps to 30 Mbps, covering resolutions from 144p to 4K. Each capture runs for 60 s (without transient phase), using the testbed from Section III-A, with each bitrate transmitted 60 times for statistical significance. System B receives and records the data while CPU utilization is measured to evaluate monitoring overhead. Between 100 kbps (uplink-only YouTube mobile traffic [16]) and 2.5 Mbps (720p), CPU usage rises from 0.25 % to 2 %, with diminishing increases at higher throughput. Thus, data capturing shows minor potential for improvement. As energy use under load is already well studied [22, 23], we do not explore it further; significant savings may arise only by reducing or disabling active capture devices during idle periods.

B. Data Processing and Feature Extraction

Greater potential for savings is expected for data processing and feature extraction, which transforms raw traffic into model-ready features. Figure 2 shows the average time (ms) and energy (mJ) per extracted sample for CPU and RAM across different feature scenarios. The left bars show six base scenarios without feature importance-based reduction, where buffer health and stalling models use the same features. Middle and right bars show scenarios after removing features with importance below 0.01, separately for buffer health and stalling models. Values are based on 13,475 samples from 30 randomly selected videos, with 95 % confidence intervals. CPU energy dominates over RAM. In full feature scenarios, energy use correlates with feature count (Table II), but this weakens after feature reduction. For instance, the stalling *all_long* scenario uses 37 features yet has the highest demand, while *uplink_only_short*, with 51 features, has the lowest. Buffer health models show a similar pattern, indicating feature type, not just quantity, affects energy consumption.

C. Model Training and Performance

After feature assessment, we evaluate how feature reduction and model complexity impacts training time and energy consumption, followed by an analysis of predictive performance.

1) Impact of Feature Reduction on Energy Consumption: We first assess the impact of feature importance–based reduction on energy consumption and training time. For each scenario and model complexity, 30 models are trained to ensure statistical significance. Training time depends on the number of samples and features per sample. Using 5,000 training samples, we evaluate the effects of complexity and feature reduction, measuring model size (nodes) and performance on a separate test set. Small confidence intervals across all scenarios indicate low variance between runs.

Buffer Health Regression: We begin by assessing the buffer health regression models. Figure 3 shows training time and CPU/RAM energy consumption for high, medium, and low complexity models (see Section III-C), evaluated across the six feature scenarios, with and without feature importance–based reduction (*FI*, dashed bars). Without reduction, resource usage and energy consumption increase with more features, consistent across complexities. Lower complexity significantly reduces energy demand (CPU energy drops from over 100 J at high to under 20 J at low complexity). With reduction, trends vary. For example, *uplink_only_short* becomes the most resource-intensive, likely due to its higher remaining feature count. Model performance and size (node count) remain largely unaffected by feature reduction, reflecting model complexity rather than feature count.

Stalling Classification: For stalling classification, training time and CPU/RAM energy consumption follow similar trends, about an order of magnitude lower, thus not plotted in detail. Feature reduction has minimal effect and model complexity remains the main driver of energy use. Model size and

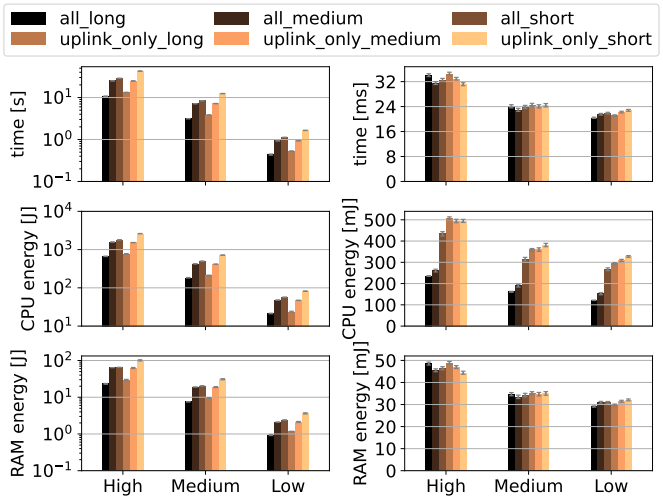

Figure 4: Time, CPU, and RAM energy consumption for training (left) and inference (right) for six feature importance-reduced scenarios at high, medium, and low model complexity.

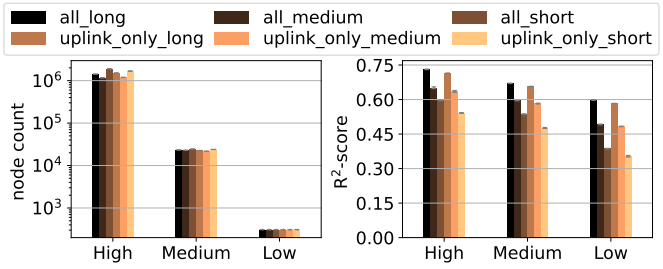

Figure 5: Number nodes (left), model performance (right) for feature scenarios of high, medium, and low model complexity.

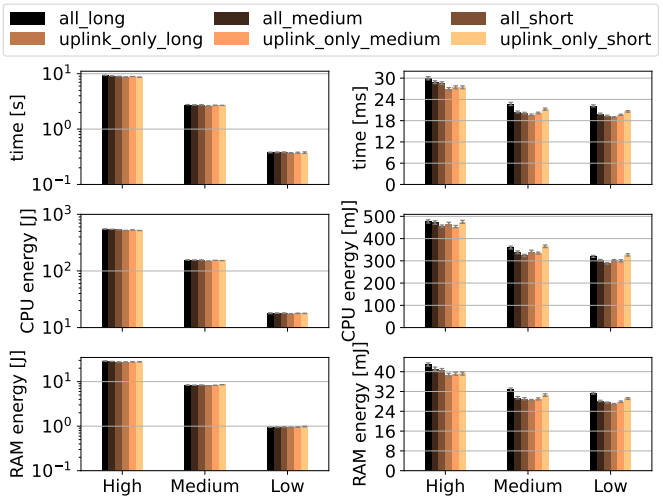

Figure 6: Time, CPU, and RAM energy consumption for training (left) and inference (right) for six feature importance-reduced scenarios at high, medium, and low model complexity.

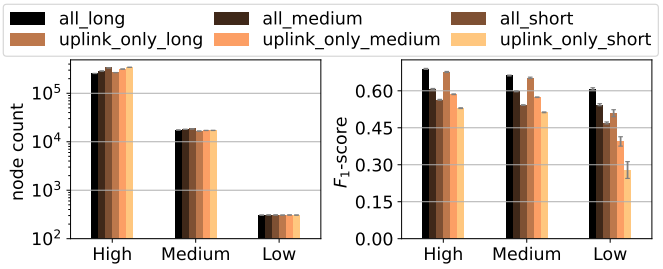

Figure 7: Number nodes (left), model performance (right) for feature scenarios of high, medium, and low model complexity.

prediction performance show similar patterns, though uplink-only scenarios perform worse at low complexity, indicating limited predictive value of uplink features under constrained capacity. Other findings align with buffer health regression and are thus omitted here.

2) Energy and Quality Assessment for Model Complexities: Model complexity has a larger impact on energy consumption than feature reduction. We therefore examine its effect on energy use and predictive performance using 100,000 samples from [3]. Models are trained on reduced feature sets for practical deployment. In addition to energy for training, typically performed once or for updates only, we measure inference energy based on 1,500 randomly selected predictions, reflecting continuous operation in the network.

Buffer Health Regression: Figure 4 shows training (left) and inference (right) time, CPU, and RAM energy for buffer health regression models with 100,000 samples across six feature-reduced scenarios and three model complexities. Training uses a logarithmic scale due to larger energy differences than per-sample inference. RAM energy is about an order of magnitude lower than CPU. Training demand is driven mainly by model complexity, with minor feature-set variation. During inference, low-complexity models are 30 % faster and use less RAM, though differences between medium and low complexity are small. Notably, *all_long* and *all_medium* scenarios consume only half the CPU energy of others. Figure 5 shows node counts and R^2. Counts are consistent within complexity levels, though high-complexity models grow with more training data. More complex models generally perform better, but scenario choice also matters; e.g., *all_long* low-complexity outperforms some medium-complexity models. Using longer windows (e.g., 20 s) increases RAM energy and prediction time, yet significantly reduces CPU energy.

Stalling Classification: Figure 6 shows training (left) and inference (right) time, CPU, and RAM energy for stalling classification models across six feature-reduced scenarios and three complexity levels. Unlike buffer health regression, training time and energy are nearly identical within each complexity level and minimally affected by feature sets. Prediction shows small differences, often not statistically significant (e.g., *all_long* vs. *uplink_only_short*). Overall, training stalling models consumes less energy than buffer health regression, while prediction demands are similar. Figure 7 shows consistent node counts for medium and low complexity, slight

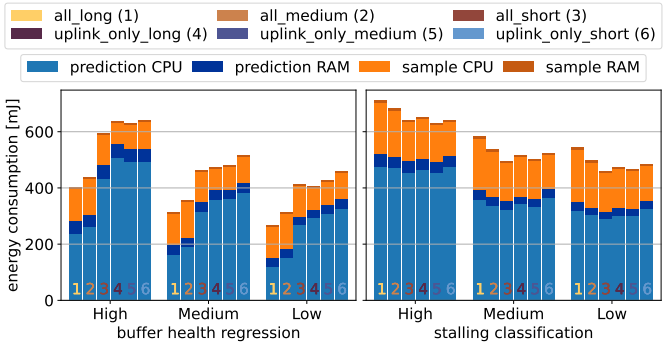

Figure 8: Total energy per sample, split into CPU and RAM usage for feature calculation and prediction, across six feature scenarios and three model complexities for buffer health regression and stalling classification.

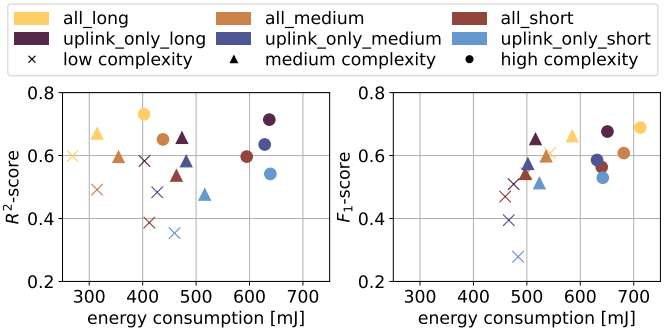

Figure 9: Total energy per sample vs. model performance across six feature scenarios and three model complexities for buffer health regression and stalling classification.

variation at high complexity, and growth with more samples. F_1-score depends on both model complexity and features; low-complexity uplink-only models perform poorly, while *all_long* low-complexity outperforms several medium- and high-complexity models, highlighting feature set importance.

Thus, the first part of **RQ1** can be addressed: *Energy consumption varies across the four KQI prediction procedures. While traffic capturing energy is low, sample extraction, feature assessment, and model training show substantial variation. Careful model selection, complexity assessment, and feature design are essential for energy-efficient operation, offering significant optimization potential.*

While model training varies by architecture, it is usually performed only once or for updates. In contrast, feature extraction and inference occur continuously during deployment, making their runtime efficiency and energy consumption more critical. Figure 8 shows CPU (light blue) and RAM (dark blue) energy for prediction, and CPU (light orange) and RAM (dark orange) for sample calculation, with numbers (1)–(6) denoting feature scenarios. Across scenarios, prediction consistently consumes more energy than sample calculation, with RAM using less than CPU. Trends align with earlier observations for buffer health regression and stalling classification. Interestingly, in buffer health regression, *all_long* (longest time windows, uplink and downlink) uses the least energy, while *uplink_only_short* consumes the most. Stalling classification shows more expected patterns, though *all_short* and *uplink_only_medium* are lower than others. On average, stalling models consume more energy than buffer health models. This addresses the second part of **RQ1**: *Beyond feature selection, model choice, complexity, and energy consumption during inference shows significant variation and opportunities for savings.* It also answers **RQ2**: *Energy use differs when predicting KQIs. While raw traffic capture and full feature computation are identical for buffer health and stalling, tailored feature sets and models vary in energy demand. Sample calculation for stalling consumes more energy than for buffer health, whereas stalling model training is roughly an order of magnitude less energy-intensive.*

D. Energy Consumption and Performance Trade-Off

Figure 9 shows the trade-off between energy consumption and model performance (R^2 for buffer health, left; F_1 for stalling, right). The x-axis shows energy use, markers indicate complexity (cross: low, triangle: medium, circle: high), and colors denote feature scenarios, all with feature importance–based reduction for smaller, practical feature sets.

Generally, larger feature sets improve performance. For buffer health, *all_long* at high complexity achieves the best result. While higher complexity often consumes more energy, some top-performing models, like the *all_long* one, use less than simpler models (fifth-lowest energy, about 400 mJ). *All_medium* at medium complexity achieves the third-best score at about 100 mJ less, sacrificing about 5 % performance. For stalling, *all_long* (high complexity) uses the most energy (BOUT 700 mJ), while *uplink_only_medium* performs within 5 % of the best but uses about 200 mJ less. *Uplink_only_short* performs poorly in both energy and accuracy.

This addresses the first part of **RQ3**: *Clear trade-offs exist between prediction quality and energy consumption. More complex models generally consume more energy but achieve better performance. Reducing features or using seemingly simpler features does not always lower energy use, and some low-accuracy models require disproportionately high energy.*

E. Discussion and Implications for Real Usage

Our findings highlight three key takeaways that should be considered for practical deployment:

1) Training consumes much more energy than inference;
2) Energy requirements for predicting different KQIs vary;
3) Thorough feature evaluation, careful model design, and systematic testing are essential for achieving optimal results.

While these points may seem at least partially intuitive, a detailed examination of them offers valuable insights that can improve how streaming KQIs are assessed and predicted.

1) Energy Consumption for Training and Inference: At first glance, training seems far more energy-intensive than inference. However, even energy-heavy training can improve long-term efficiency. For instance, our full buffer health regression model uses over 1,000 J for training (Figure 4), while

inference consumes only 300–600 mJ, depending on features and complexity. Once trained, a model can be reused; in the *all_long* low-complexity scenario, inference uses about 4,000 times less energy than training, equivalent to 4,000 one-second predictions. With the large volume of video streams today, investing in centralized, renewable-powered training enables lightweight inference at distributed locations, reducing overall energy consumption and carbon footprint while supporting scalable KQI prediction.

2) Energy Consumption for Different KQIs: Energy consumption depends on the predicted KQI, so the most critical KQIs for an application must be identified. Many approaches (e.g., [16, 12]) target different KQIs such as re-buffering events or quality changes. Our analysis shows that relevant input features vary by KQI, requiring distinct feature sets after feature reduction. Although predicting buffer health seems more energy-intensive than stalling, it can also infer stalling and quality changes, making a slightly more energy-demanding buffer health model potentially more efficient than maintaining separate models for each KQI.

3) Feature Assessment and Model Testing: Currently, most AI models and features are selected primarily for prediction quality, often targeting marginal gains in F_1-score or accuracy. We propose including energy consumption as an additional KQI during model selection, allowing trade-offs between accuracy and energy use to be assessed for each use case. Some scenarios prioritize maximum accuracy, while others may accept slightly lower performance for substantial energy savings. In video streaming, even small improvements can prevent critical issues like stalling, so energy saved with lightweight, adequately performing models could be redirected to improve overall network performance. Thus, the second part of **RQ3** can be answered: *Future model design should consider energy consumption as a KQI, optimizing models for both energy efficiency and prediction performance.*

V. Conclusion

In this work, we predict buffer health and stalling as key QoE indicators in video streaming using RF models. We analyze how feature spaces and model complexity affect prediction accuracy and energy use, emphasizing relative comparisons due to hardware dependence. Results show both feature choice and model complexity strongly influence energy demand: complex models generally consume more energy for higher accuracy, but input features also shape efficiency. Uplink-only features underperformed compared to combined uplink–downlink sets, while longer time-window features improved buffer health regression. Notably, the most accurate models were not always the most energy-intensive, highlighting the importance of informed feature and model selection. We propose including energy consumption as an additional evaluation dimension alongside accuracy. Future work will extend the analysis to GPU-based training and deep learning models, where infrequent retraining and low-cost inference may offset high training energy.

Acknowledgments

The work is funded by the Federal Ministry of Research, Technology and Space, Grant 18KIS2282 “SUSTAINET-Advance”, sub-project 6G-ECONETS of the University of Würzburg and supported by the Swiss Innovation Agency Innosuisse under the SUSTAINET project 119.588 INT-ICT.

References

[1] Computer Weekly, “Sporting Events Drive 2023’s Biggest Daily Spikes in European and US Network Traffic,” accessed: 2025-01-30. [Online]. Available: https://www.computerweekly.com/news/366568294/Sporting-events-drive-2023s-biggest-daily-spikes-in-European-and-US-network-traffic

[2] M. Seufert *et al.*, “A Survey on Quality of Experience of HTTP Adaptive Streaming,” *IEEE Communications Surveys & Tutorials*, 2014.

[3] F. Loh *et al.*, “YouTube Dataset on Mobile Streaming for Internet Traffic Modeling and Streaming Analysis,” *Scientific Data*, 2022.

[4] F. Loh, F. Wamser, C. Moldovan, B. Zeidler, D. Tsilimantos, S. Valentin, and T. Hoßfeld, “Is the Uplink Enough? Estimating Video Stalls from Encrypted Network Traffic,” in *Network Operations and Management Symposium*. IEEE, 2020.

[5] F. Loh, A. Pimpinella, S. Geißler, and T. Hoßfeld, “Uplink-based live session model for stalling prediction in video streaming,” in *Network Operations and Mgmt Symposium*. IEEE, 2023.

[6] F. Loh *et al.*, “Machine learning based study of qoe metrics in twitch.tv live streaming,” in *Network Operations and Mgmnt Symp.* IEEE, 2023.

[7] C. Gutterman *et al.*, “Requet: Real-time qoe metric detection for encrypted youtube traffic,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2020.

[8] Z. Shang, J. P. Ebenezer, Y. Wu, H. Wei, S. Sethuraman, and A. C. Bovik, “Study of the Subjective and Objective Quality of High Motion Live Streaming Videos,” *IEEE Transactions on Image Processing*, 2021.

[9] G. Bingöl *et al.*, “An analysis of the trade-off between sustainability and quality of experience for video streaming,” in *International Conference on Communications Workshops*. IEEE, 2023.

[10] T. Hoßfeld, M. Varela, L. Skorin-Kapov, and P. E. Heegaard, “A Greener Experience: Trade-Offs between QoE and CO 2 Emissions in Today’s and 6G Networks,” *IEEE Communications Magazine*, 2023.

[11] F. Loh, *Monitoring the Quality of Streaming and Internet of Things Applications*. Bayerische JMU Würzburg (Germany), 2023.

[12] S. Wassermann *et al.*, “Vicrypt to the Rescue: Real-Time, Machine-Learning-Driven Video-QoE Monitoring for Encrypted Streaming Traffic,” *IEEE Transactions on Network and Service Management*, 2020.

[13] I. Orsolic and L. Skorin-Kapov, “A Framework for in-Network QoE Monitoring of Encrypted Video Streaming,” *IEEE access*, 2020.

[14] S. C. Madanapalli *et al.*, “ReCLive: Real-Time Classification and QoE Inference of Live Video Streaming Services,” in *International Symposium on Quality of Service*. IEEE, 2021.

[15] F. Loh *et al.*, “High Complexity and Bad Quality? Efficiency Assessment for Video QoE Prediction Approaches,” in *International Conference on Network and Service Management*. IEEE, 2024.

[16] F. Loh, F. Poignée, F. Wamser, F. Leidinger, and T. Hoßfeld, “Uplink vs. Downlink: Machine Learning-based Quality Prediction for HTTP Adaptive Video Streaming,” *Sensors*, 2021.

[17] T. N. Duc *et al.*, “Convolutional Neural Networks for Continuous QoE Prediction in Video Streaming Services,” *IEEE Access*, 2020.

[18] N. Eswara *et al.*, “Streaming Video QoE Modeling and Prediction: A Long Short-Term Memory Approach,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[19] K. Nguyen *et al.*, “Investigation of Serverless Consumption and Performance in Multi-Access Edge Computing,” in *International Conference on Information Networking*. IEEE, 2024.

[20] F. Bronzino *et al.*, “Inferring Streaming Video Quality from Encrypted Traffic: Practical Models and Deployment Experience,” *ACM on Measurement and Analysis of Computing Systems*, 2019.

[21] PyRAPL, “PyRAPL,” accessed: 2025-07-04. [Online]. Available: https://pypi.org/project/pyRAPL/

[22] M. A. Hodkin *et al.*, “Energy-adaptive Network Switching via Intra-device Scaling,” in *Int. Conf. on Communications*. IEEE, 2024.

[23] N. Mehling, F. Loh, and T. Hoßfeld, “Low-Cost Energy Measurement and Multi-Port Traffic Generation for Network Devices,” 2024.