Towards an ML Assisted DASH-based Architecture: Leveraging Predictive Network Analyses with Interpretability

Eduardo R Peretto*, Manoel Narciso Reis Soares Filho[†], Débora Cristina Santos Sousa[‡],
Luciano Paschoal Gaspary[§] and Bruno Iochins Grisci[¶]
Institute of Informatics, Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
Email: *erperetto@inf.ufrgs.br, [†]mnrsfilho@inf.ufrgs.br, [‡]debora.sousa@inf.ufrgs.br, [§]paschoal@inf.ufrgs.br
, [¶]bigrisci@inf.ufrgs.br

***Abstract*—While Dynamic Adaptive Streaming over HTTP (DASH) is the standard for media delivery, most adaptive bitrate (ABR) algorithms remain reactive. Integrating predictive insights without compromising system design is a key challenge. This paper presents a feasibility study of an ML-enhanced DASH architecture that generates lightweight, interpretable prediction hints to enable proactive ABR. We validate the framework using a case study on over 10,000 hours of real traffic traces from Brazil's Rede Ipê backbone. Using Random Forest and Gradient Boosting models, we compare a DASH-only feature set against one enriched with network metrics (RTT, traceroute). Our results demonstrate the viability of the architecture and highlight key network indicators that drive predictions. By focusing on interpretability and statistical validation, our work provides a transparent framework for integrating predictive modules into DASH ecosystems, laying the groundwork for more robust, next-generation ABR algorithms.**

***Index Terms*—Predictive Adaptive Streaming, AI-driven Decision-making, Multimedia Architectures**

I. Introduction

Adaptive streaming enables the delivery of video over the internet by dynamically adjusting the quality of the delivered content according to network conditions [1]. One widely adopted adaptive streaming technique is Dynamic Adaptive Streaming over HTTP (DASH). In the DASH approach, the video is divided into small segments, which are made available to the client at a variety of bitrates, that is, different amounts of data per second, corresponding to different levels of compression and visual quality [2]. The DASH client must request in real-time, during its execution, the next content segments in a way that optimizes the bitrate while avoiding playback interruptions (i.e., buffering), thereby ensuring a seamless viewing experience even in fluctuating environments [3].

While traditional DASH algorithms are based on reactive methods that simply respond to the current situation of the network, predictive methods capable of anticipating the future network state with a certain degree of accuracy could significantly improve the efficiency of such systems [4], [5]. By forecasting likely changes in bandwidth or latency, predictive approaches enable the system to adapt in advance, for example by pre-loading higher-quality video segments during anticipated high-bandwidth periods or lowering quality before a predicted drop, thereby ensuring more seamless playback compared to purely reactive techniques [4].

This work presents a foundational feasibility study of an architecture for integrating ML prediction hints into DASH to improve ABR algorithms. Our key contribution lies not in the discovery of specific prediction rules, but in providing a validated interpretability framework that demonstrates how lightweight, transparent models can be embedded in DASH. Our experiments compare a base feature set—containing only DASH measurements—against an enriched configuration that incorporates path and latency metrics, thereby revealing which measurements most strongly influence future bitrate selections and establishing a roadmap for subsequent investigations. It is important to note that this study focuses on the prediction module; a full end-to-end evaluation of how these hints translate into QoE improvements within an ABR is left as future work.

Our design philosophy is centered on interpretability, a crucial factor for network operators who must validate automated decisions to ensure operational trust. This emphasis aligns with foundational research highlighting that model transparency is a prerequisite for deploying trustworthy ML-driven systems in production environments [6]. By exposing the contribution of each input variable, operators gain actionable insights into network behavior, facilitating proactive tuning of streaming parameters and fostering confidence in automated bitrate decisions. Such clarity not only supports more effective resource allocation but also ensures that stakeholders can audit and trust the adaptation logic as it responds to changing conditions [7]. Building on this, we employ a family of ensemble predictors - namely Random Forest (RF) and Gradient Boosting (GB) - known for their balance of expressiveness and explainability,

This research was funded by *Rede Nacional de Ensino e Pesquisa* (RNP), through the *Comitê Técnico de Monitoramento de Redes* (CT-MON) project, and also by *Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul* (FAPERGS) [24/2551-0000725-3]. This study was financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001.

and we evaluate them under both feature configurations.

Our dataset derives from measurements collected on Rede Ipê, the Brazilian National Research and Education Network backbone that connects over five hundred academic institutions. Between June and October 2024, two PoPs (Rio de Janeiro and Salvador) generated application-level traces, issuing periodic bursts of fifteen DASH segment requests and logging bitrates, request durations and success rates. Simultaneously, MonIPÊ captured ten-sample round-trip time (RTT) histograms and traceroute paths from server PoPs (Brasília, Fortaleza, Teresina and Vitória) toward the clients. All records were serialized, subjected to validation, and synchronized into 1-hour slots, producing over ten thousand observation windows.

This paper is organized as follows. Section I introduces the study and its objectives, while Section II reviews related works in predictive adaptive streaming, network performance prediction, and the machine learning techniques adopted for this work. Section III then introduces the proposed models within the DASH ecosystem, describing its design principle, operational flow, and expected benefits for proactive bitrate adaptation. Sections IV and V describe the dataset — including data sources and pre-processing steps — and the methodology, detailing data filtering, feature selection, model configuration, and evaluation procedures, respectively. Section VI presents the experimental setup and results, followed by a discussion of the findings. Finally, Section VII concludes the paper by summarizing key insights, addressing limitations, and suggesting directions for future research.

II. Related Work

Conventional Dynamic Adaptive Streaming over HTTP (DASH) clients rely on purely reactive bitrate adaptation strategies, adjusting to metrics like throughput or buffer occupancy at runtime [4]. While simple to deploy, these heuristics often underperform in variable network conditions, causing avoidable rebuffering and bitrate instability [4], [8]. To overcome these limitations, many studies have proposed predictive DASH architectures that anticipate network fluctuations to adapt video delivery proactively [4], [5], [8]–[10].

Interpretability of these adaptive bitrate (ABR) predictors, however, has been less studied. The internal decision-making logic of many machine learning models is often opaque, complicating troubleshooting and failing to provide clear insights into why certain ABR decisions are made. In contrast, interpretable models increase operational trust and adoption by making the decision logic transparent. This transparency allows providers to mitigate systematic failures, customize ABR logic for specific users or content, and audit bitrate decisions for fairness and bias in large-scale deployments. Frameworks such as feature importance can help operators understand the impact of network parameters on bitrate selection [11].

Overall, a gap remains in the literature regarding the integration of interpretable ML models within a practical, modular DASH architecture. Our work addresses this gap by presenting a feasibility study of such an architecture, moving beyond simulation to validate it with data from a production backbone. We propose a modular design with an embedded prediction service that leverages inherently interpretable models, namely Random Forest and Gradient Boosting, to deliver explainable and trustworthy insights for next-generation ABR systems.

III. Bridging Prediction Models and DASH Architecture

Recent studies have showcased how predictive modeling can improve Quality of Experience (QoE) in video streaming [12]–[14]. However, integrating such models into existing ecosystems presents significant architectural challenges. Our proposed architecture, illustrated in Figure 1, is designed to embed predictive intelligence into the DASH workflow in a modular, efficient, and non-disruptive manner.

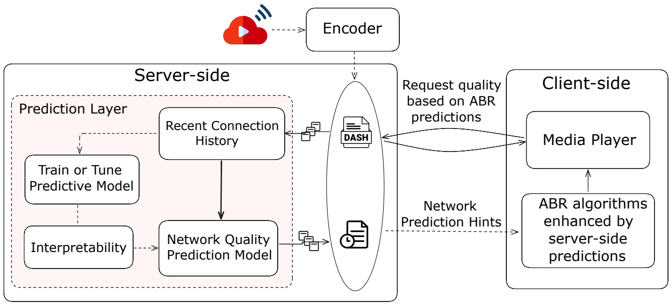

Encoder
Server-side
Prediction Layer
Recent Connection History
Train or Tune Predictive Model
Interpretability
Network Quality Prediction Model
DASH
Request quality based on ABR predictions
Network Prediction Hints
Client-side
Media Player
ABR algorithms enhanced by server-side predictions

Fig. 1: Proposed architecture of Adaptive-Streaming Ecosystem enriched by network prediction.

A core design principle is to decouple the computationally intensive prediction logic from the client's Adaptive Bitrate (ABR) algorithm. As shown on the server-side of Figure 1, a dedicated *AI Network Prediction* module is responsible for all ML-related tasks. This module processes the *Recent Connection History* to train and execute a *Network Quality Prediction Model* . By centralizing this logic on the server, clients with limited computational power are not burdened with running complex models, making the approach widely applicable across heterogeneous devices.

The system delivers tailored, server-side hints specifically for each client connection. Instead of dictating a specific bitrate, the server generates *Network Prediction Hints* based on its analysis—forecasting impending congestion or periods of high stability—which are sent to the client alongside standard DASH content. The client's ABR algorithm is then free to use this information to enhance its decisions, for instance, by proactively selecting a lower bitrate before a predicted drop in throughput to avoid a stall. Crucially, this hint-based mechanism ensures non-disruptive adoption, as legacy clients that do not support this feature can simply ignore the supplementary data and operate using their default ABR logic. This approach ensures backward compatibility and allows for a gradual rollout, enabling proactive streaming strategies without requiring an overhaul of the existing DASH ecosystem.

While this predictive approach offers significant advantages—such as proactively mitigating buffering by anticipating network fluctuations—its practical implementation faces several hurdles. Key challenges include maintaining model accuracy and generalization across dynamic network conditions, managing the operational costs of scalability and frequent retraining, and ensuring seamless, secure integration with existing streaming architectures without compromising data privacy or interoperability.

IV. Exploring a Realistic Dataset

The empirical foundation of our study rests on measurements gathered between June and October 2024 over the Rede Ipê infrastructure[1], the nationwide academic backbone that interconnects more than five hundred research and educational institutions in Brazil. Within this backbone, two Points of Presence (PoPs), located in Rio de Janeiro (RJ) and Salvador (BA), were instrumented as clients, while four PoPs in Brasília (DF), Fortaleza (CE), Teresina (PI) and Vitória (ES) fulfilled the role of servers. This deployment yields a total of eight logical client–server pairs, denoted respectively by the sets $\mathcal{C} = \{\mathrm{RJ}, \mathrm{BA}\}$ and $\mathcal{S} = \{\mathrm{DF}, \mathrm{CE}, \mathrm{PI}, \mathrm{ES}\}$.

End-to-end network performance was captured through two complementary tools. Neubot DASH [15], obtained via the M-Lab platform, emulated an HTTP adaptive streaming client by issuing a burst of fifteen DASH segment requests every five minutes from each node in $\mathcal{C}$ to every node in $\mathcal{S}$. For each request, the experiment logged the HTTP transaction timestamp, segment bitrate, and various transport-layer statistics, storing all data for a five-minute window in JSON-lines format. Concurrently, MonIPÊ, a network observation tool maintained by the Rede Nacional de Ensino e Pesquisa (RNP), monitored the Rede Ipê backbone by recording round-trip time (RTT) histograms from ten consecutive latency samples and issuing traceroute probes from servers to clients to log intermediate hops and delays. These network-level results were serialized in JSON, enabling a detailed reconstruction of both per-hop path characteristics and temporal latency patterns.

Together, Neubot DASH and MonIPÊ deliver a rich, synchronized view of streaming performance and underlying network state. The DASH traces reveal how application-level bitrate decisions interplay with delivery success, while RTT and traceroute logs expose the transient behaviors of the transport and routing layers. Prior to analysis, all raw files were subjected to rigorous validation: any JSON record missing expected fields, incomplete histograms or truncated traceroute paths, as well as DASH windows lacking the full complement of segment requests, were discarded. The resulting corpus comprises over ten thousand validated five-minute windows, spanning all eight client–server combinations and capturing diurnal, weekly and cross-geographic variations. This dataset forms the backbone for our feature engineering and model training pipelines described in Section V, offering a realistic substrate for evaluating predictive approaches in adaptive streaming.

V. Methodology

The methodological workflow is summarized in Figure 2. We collected DASH and MonIPÊ measurements from Rede Ipê and synchronized them at 5-minute intervals. All valid records were aggregated into 1-hour slots; slots with fewer than 10 DASH or 5 RTT/Traceroute samples were discarded, resulting in 10,142 usable windows. Feature engineering included incremental differences of bitrate, RTT, and path metrics, followed by normalization with a standard scaler.

We modeled four predictive tasks: deltas of bitrate mean and standard deviation at $t+5$ and $t+10$ minutes (`mean_1`, `stdev_1`, `mean_2`, `stdev_2`). To assess the contribution of network-level signals, we compared two feature sets: (i) Base, containing only DASH statistics; (ii) Enriched, adding RTT and traceroute features such as hop-count variability and aggregated RTT.

Random Forest (RF) and Gradient Boosting (GB) were selected for their balance of accuracy and interpretability, as recommended in ABR studies [16]. Hyperparameters (e.g., trees, depth, learning rate) were optimized via grid search with cross-validation, minimizing Mean Absolute Percentage Error (MAPE). The dataset was split into 75% training and 25% testing, yielding eight models (RF and GB across both feature sets and all four targets). Performance was compared on the held-out set, and feature importances were analyzed to interpret decisions.

A. Experimental Setup

Experiments were run in Python 3.10.12 using `scikit-learn` (v1.5.1), `pandas` (v2.2.2), and `scipy`. To ensure robustness, all experiments were repeated with 11 random seeds.

B. Evaluation Protocol

Model performance was evaluated using three metrics: (1) *MAPE*, as in prior studies [5], [9], [10]; (2) *Train Time*, measuring computational cost; and (3) *Inference Time*, reflecting runtime feasibility.

C. Explainability Algorithm

We employed RuleFit [17], [18], which extracts decision-tree rules from RF and GB models and combines them with linear effects. This approach provides readable rule sets and feature-importance scores, enabling operators to understand which variables drive predictions and why.

VI. Results

This section presents the results obtained after evaluating the performance of the Random Forest and Gradient Boosting models across all tested configurations. We report on predictive accuracy, training time, inference time, and the results of a statistical analysis to evaluate the significance of differences between approaches. In total, 176 experimental runs were conducted, covering all combinations of models, feature sets, target variables, and random seeds.

[1] https://redeipe.rnp.br/home

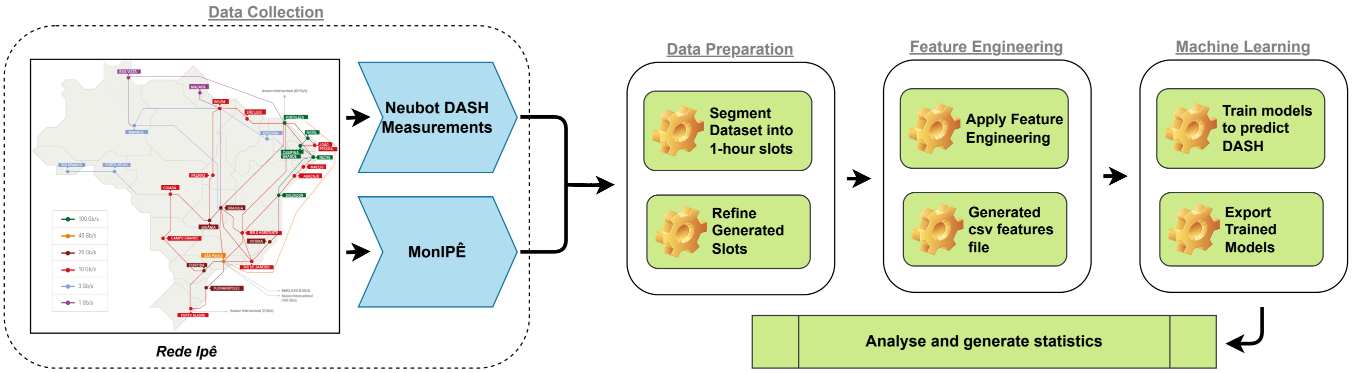


Fig. 2: The methodological workflow, describing the data collection and processing steps.

A. Evaluating MAPE and Training Time

Table I presents the MAPE and the training time along with its standard deviation across 11 runs (each run with a random seed used for splitting the dataset into test-train) for each model-feature combination and target metric. The MAPE is calculated relative to the target's variation since the last measurement. Overall, both RF and GB deliver comparable accuracy, with only minor performance gains observed when additional RTT/Traceroute features are incorporated. RF tends to require substantially longer training times (tens to over one hundred seconds), while GB typically completes training in under one second for most cases. This disparity highlights GB's advantage when rapid model retraining or frequent updates are required. Furthermore, Table I also shows the target metrics obtained using a simple arithmetic average method, as a referential baseline. It can be observed that the trained models outperform the baseline, although further studies are required to assess its real-world meaningfulness and practical impact, as discussed in Section VII.

B. Feature Importance

As discussed previously, the prediction interpretability is the main target of the running experiments, and it can be enhanced through the estimation of the importance of the features. We conducted a comprehensive feature importance analysis to identify the most critical factors influencing the prediction of subsequent Dynamic Adaptive Streaming over HTTP (DASH) measurements. The analysis revealed that the following features are the most significant:

- `rates_mean`: The average bitrate of all 10 historical DASH measurements.
- `dash_last_rate`: The average of the most recent DASH bitrate measurement.
- `rates_stdev`: The average standard deviation of the 10 historical DASH bitrate measurements.
- `dash_last_rate_std`: The standard deviation of the most recent DASH bitrate measurement.
- `dash0_rate_mean`: The mean bitrate of the oldest DASH measurement.
- `client_server_id`: An identifier representing the specific client-server combination.
- `rtt0_mean`: The mean Round-Trip Time (RTT) of the first measurement conducted in MonIPÊ.

Figure 3 illustrates the feature importance results for the $t + 1$ DASH prediction, comparing both enriched and base datasets for mean and standard deviation forecasting. Overall, the models consistently rely on `rates_mean` and `dash_last_rate` for predicting future bitrate measurements, with supplementary contributions from `client_server_id`, `dash0_rate_mean`, and `rtt0_mean` in enriched feature sets. These findings provide a more comprehensive understanding of which factors most significantly impact DASH performance, guiding future refinement and feature engineering strategies for improved predictive accuracy.

Table I also reports the average inference time for single predictions across various model and feature set configurations, aggregated over 11 independent runs. Notably, all models achieved inference times below 1 ms, demonstrating exceptional efficiency suitable for real-time applications. Both RF and GB models exhibited low computational overhead, affirming their viability for online adaptation in Dynamic Adaptive Streaming over HTTP (DASH) systems. The minimal standard deviations indicate consistent performance across multiple executions, ensuring reliable responsiveness in latency-sensitive environments.

1) Gradient Boosting Models Explainability: Looking at the feature rules extracted from the trained Gradient Boosting models by RuleFit using the `mean_1` feature, we observe that the features `dash_last_rate` and `dash9_rate_mean` appear consistently as the most important to the model predictions with a coefficient of impact on the linear regression of approximately $-19,000$ and $-10,000$ in both datasets. The impact of these model features is coherent with the feature importance obtained. Features such as `rates_mean` and `dash8_rate_mean` also appeared in the top five important features globally. The impact coefficient for `rates_mean` had a positive impact on the predicted bitrate values, as opposed to the impact of the last dash rates obtained, being the impact of `dash8_rate_mean` slightly smaller than that of `dash9_rate_mean`, possibly being a dilution of importance

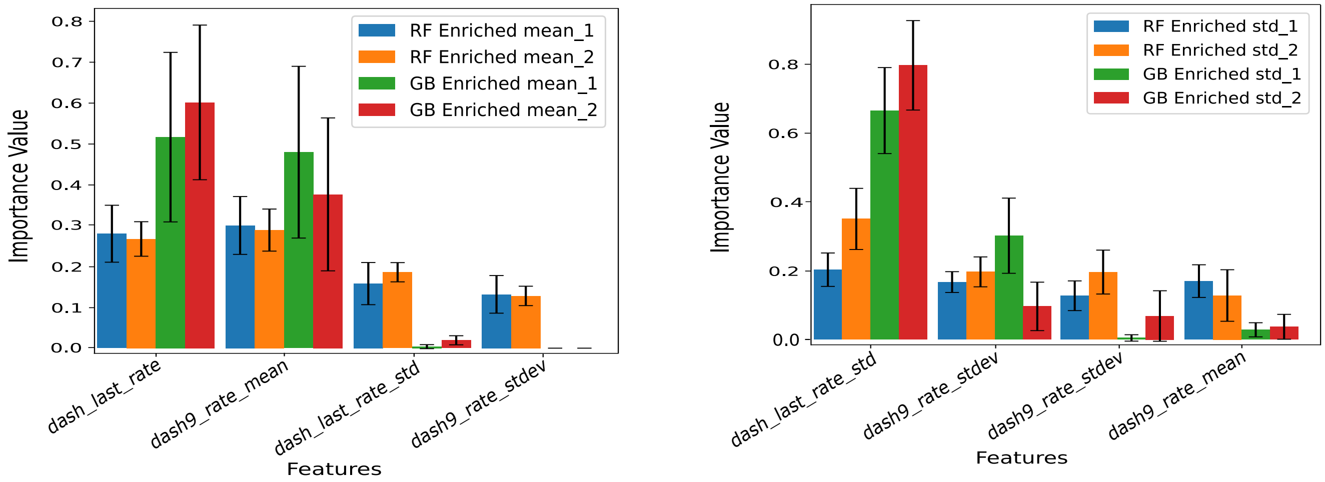
TABLE I: Aggregated MAPE, Training Time, and Inference Time for RF and GB models, under all runs over both feature configurations: Base (Table Ia) and Enriched (Table Ib). N/A ⟶ Not Applicable

Model	Target	MAPE	Train Time (s)	Inference (ms)
GB	mean_1	3.64 ± 0.88	2.34 ± 0.37	0.18 ± 0.02
GB	mean_2	6.24 ± 3.49	3.02 ± 0.80	0.17 ± 0.01
GB	std_1	2.86 ± 0.50	2.24 ± 0.22	0.22 ± 0.01
GB	std_2	3.33 ± 0.97	2.37 ± 0.66	0.22 ± 0.01
RF	mean_1	3.60 ± 0.70	2.87 ± 0.62	0.33 ± 0.03
RF	mean_2	5.85 ± 2.76	6.72 ± 0.73	0.38 ± 0.02
RF	std_1	2.97 ± 0.54	2.70 ± 0.67	0.26 ± 0.02
RF	std_2	3.46 ± 0.89	1.19 ± 0.12	0.27 ± 0.01
Arithmetic Average	mean_1	5.59 ± 0.52	N/A	N/A
	mean_2	11.07 ± 2.50		
	std_1	4.40 ± 0.51		
	std_2	3.90 ± 0.29		

(a) Base Feature Configuration

Model	Target	MAPE	Train Time (s)	Inference (ms)
GB	mean_1	3.72 ± 1.14	4.60 ± 1.12	0.26 ± 0.03
GB	mean_2	6.11 ± 3.15	5.05 ± 0.89	0.17 ± 0.02
GB	std_1	2.87 ± 0.49	2.98 ± 0.65	0.33 ± 0.03
GB	std_2	3.32 ± 0.92	2.99 ± 0.54	0.32 ± 0.03
RF	mean_1	3.69 ± 1.14	1.41 ± 0.20	0.32 ± 0.02
RF	mean_2	6.36 ± 3.49	9.02 ± 1.85	0.98 ± 0.01
RF	std_1	3.03 ± 0.52	2.65 ± 0.53	0.28 ± 0.01
RF	std_2	3.27 ± 0.80	1.52 ± 0.13	0.28 ± 0.01
Arithmetic Average	mean_1	5.59 ± 0.52	N/A	N/A
	mean_2	11.07 ± 2.50		
	std_1	4.40 ± 0.51		
	std_2	3.90 ± 0.29		

(b) Enriched Feature Configuration



(a) mean1 and mean2 feature importances with enriched feature set.

(b) std1 and std2 feature importances with enriched feature set.

Fig. 3: Feature importances for bitrate mean variation and standard deviation at $t + 1$ and $t + 2$ using enriched dataset. The base dataset performed similar to it's enriched counterpart.

TABLE II: Kruskal–Wallis H-test for MAPE distributions across model–feature configurations.

Target	H	p-value	Significant? ($\alpha = 0.05$)
mean_1	0.0821	0.9938	No
mean_2	0.2849	0.9628	No
std_1	0.8250	0.8434	No
std_2	0.5516	0.9074	No

as they get farther in time from the prediction. This suggests that the model is learning to capture temporal bitrate dynamics to identify patterns in bitrate progression, such as gradual rises due to motion buildup or sudden drops following static content.

Aside from the linear type metrics, it generated rules for dash_last_rate that were fairly different between the datasets. The rule generated on the Base dataset was able to cover 2% of the data points with the following rule: dash_last_rate ≤ -2.631, while the generated rule on the Enriched dataset had a coverage of 97% of the data points, and was approximately the inverse of the previously presented rule. Aside from that, on the enriched dataset, the dash0_rate_mean was also a feature with a global importance on the top five of the linear rules, with a positive coefficient impact of 5.000.

Switching to the results of the Gradient Boosting Models for the mean_2 features, the aforementioned trends are maintained, but the absolute value of the coefficient of dash_last_rate drops by about $8,000$, decreasing its negative impact on the outcome prediction. While the dash9_rate_mean coefficient had a less noticeable increase in its absolute coefficient value of $5,000$ in the Base dataset, and remained the same in the Enriched dataset. A similar rule, coefficient and coverage were found for the dash_last_rate, but now a rule for dash9_rate_mean that had a scope of 98% of the data points and a relatively high importance was found, of `dash9_rate_mean` > -2.224.

As for the standard deviation Gradient Boosting models, in both $t+1$ and $t+2$, the models trained on the Base dataset preserved the use of the dash rate values, but using their standard deviation values. Furthermore, the trend in coefficient impact was maintained, newer DASH values had a negative impact and older DASH values a positive impact. Lastly, on this dataset, no rule with a high importance had a scope of data greater than 1%. As for the Enriched dataset, the feature `tr0_rtt_stdev` was added in place of the `dash0_rate_stdev`, when compared to the Base dataset, but also having a positive coefficient impact.

This might indicate that the models struggled more to identify local rules that generalized well across the ensemble's predictions, possibly relying instead on the global impact of individual features. Nevertheless, it is important to note that although the extracted rules had low sample coverage, they still exhibited a high importance, which may suggest that such rules were capturing patterns associated with rare but influential cases—potentially indicating relevance to data outliers or extreme behaviors within the dataset. Furthermore, the overall importance values decreased from the mean-based models to the standard deviation based ones. Given that the underlying datasets were the same, the fact that the top rule in the standard deviation models had a considerably lower importance score suggests an increased difficulty in extracting informative rules, which reinforces the previous hypothesis of the model's learning struggles. Lastly, the lack of statistical difference between the Base and Enriched models can also be seen by the extracted rules, in both the mean and standard deviation trained models, as the top features are mainly DASH features.

2) Random Forest Models Explainability: Now analyzing the feature rules extracted from the trained Random Forest models by RuleFit, in both datasets analyzed, using the mean features, there was a similar trend to that observed in the Gradient Boosting, with `dash_last_rate` and `dash9_rate_mean` being again the features with the most importance, with comparable values of coefficient of impact in the `mean_1` Random Forest model as compared to the Gradient Boosting. As for the rules made in the mean models, the Base `mean_1` model first generated rules focused on those features individually, such as `dash_last_rate` ≤ -2.631 with a coefficient value of $28,468.28$, although it presented a small percentage of coverage, of only 2%, its importance remained high, which could indicate that this rule could be pointing to outlier samples. Another feature rule was $dash9\_rate\_mean > -2.186$ with a coefficient of $-20,097.53$ and a support value of 97%, reaching most of the sample data points, comparable to what was observed in the Gradient Boosting `mean_1` model. In the Enriched dataset, the support coverage for these rules was inverted compared to the Base dataset, with both rules being fairly similar.

Moving to the results for the mean_2 features, the absolute value of the coefficient of `dash_last_rate` was again decreased by about $3,000$ in the Base dataset and $9,000$ on the Enriched dataset. Contrary to the increase observed in the Gradient Boosting model, the `dash9_rate_mean` also had a decrease from the `mean_1` to `mean_2` features in about $4,000$ in both datasets. As both of them have a decrease in overall impact on the model's prediction and they were considered the most globally important features, that might indicate that the Random Forest model is struggling more to use the presented features for predictions made in a farther future. A key difference observed in the `mean_2` features model on the Enriched dataset is the importance of `rtt0_mean` feature, with a linear coefficient of $-3,224$ but with a relatively low importance.

In the standard deviation model for $t+1$, in the Base dataset, the `dash_last_rate` was dropped from the feature importance podium, while in the Enriched dataset, it began to have a positive coefficient impact. Once again, with this set of features, no extracted rule had coverage in more than 1% of the data. Lastly, no substantial difference between the Random Forest models trained on standard deviation features from time $t+1$ and time $t+2$, in terms of the rules extracted, were observed.

VII. Conclusion

In this study, we demonstrated the feasibility of a highly interpretable ML-enhanced DASH architecture designed for proactive bitrate adaptation. The primary goal was not to maximize predictive accuracy, but to validate a transparent framework suitable for real-world deployment. Consequently, the core contribution of this work is a reusable interpretability framework that enables trustworthy ABR strategies, rather than a set of specific prediction rules.

Our experiments on the *Rede Ipê* academic backbone confirmed the approach's viability: with inference times below 1 ms and rapid training cycles, both Random Forest and Gradient Boosting are suitable for near real-time use. The best models achieved a MAPE of approximately 3.6 when forecasting mean bitrate and 2.8 for its standard deviation. Our analysis also revealed that while enriching the feature set with RTT and traceroute data provided modest improvements, the difference was not statistically significant, with historical DASH metrics remaining the dominant predictors. Ultimately, our findings highlight that the choice between these models depends less on marginal performance gains and more on practical deployment constraints, such as hardware limitations and specific interpretability requirements.

Both Random Forest and Gradient Boosting demonstrated reasonable performance: training times for most model–feature combinations were under one second, and inference times remained below 1 ms per instance, confirming suitability for near real-time deployment. Furthermore, our RuleFit-based explainability pipeline (Section VI) produced detailed feature-importance rankings, reinforcing the value of interpretable predictors. By combining network-path metrics with historical bitrate data, content providers can proactively adjust streaming parameters to reduce buffering events and enhance video quality.

Overall, the findings highlight the practical balance between model complexity and computational demands: both Random Forest and Gradient Boost offer swift inference and comparable accuracy. Consequently, the choice between these approaches may depend more on specific deployment constraints—such as prediction time windows length, hardware limitations, and interpretability degree requirements than on any notable differences in predictive performance.

A. Limitations

This study has two main limitations. First, the proposed ML methods depend on high-quality labeled data and face challenges in generalizing across diverse network conditions and user behaviors [19]. Second, our work deliberately focused on validating the interpretability framework on a realistic dataset rather than on its end-to-end QoE impact. Consequently, we did not integrate the predictive hints within a client-side ABR algorithm, leaving the experimental demonstration of concrete QoE improvements as an important direction for future work.

B. Future Work

Future work should focus on deploying the proposed framework within a real-world DASH client to quantitatively evaluate its impact on user satisfaction and playback quality. Integrating these server-side prediction hints with client ABR algorithms is the immediate next step for concrete QoE-optimization studies. In parallel, prediction accuracy could be enhanced while preserving interpretability by incorporating new features—such as jitter, instantaneous bandwidth estimates, and user-centric metrics—or by exploring alternative ML models on expanded datasets. A broader research direction involves leveraging these interpretable insights for adaptive networking, enabling servers to dynamically reconfigure network paths or resource allocations to improve streaming performance.

Data and code availability

The codes for all the models discussed in this work, as well as the datasets used for training them, are available in the following online repository: https://github.com/eduardoperetto/AI-data-challenge. In addition, the original dataset provided by the RNP is available at the following link: https://dadosderede.rnp.br/dataverse/datachallenge.

Acknowledgment

Our research and the preparation of this article were funded, both intellectually and financially, by *Rede Nacional de Ensino e Pesquisa* (RNP), through the *Comitê Técnico de Monitoramento de Redes* (CT-MON) project, and also by *Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul* (FAPERGS) [24/2551-0000725-3]. This study was financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001. Therefore, we extend our sincere gratitude to the institutions.

References

[1] R. Doreswamy, A. G. Colaco, V. Sevani, P. Patil, and H. Tyagi, "Rate adaptation for low latency real-time video streaming," in *2023 National Conference on Communications (NCC)*, 2023, pp. 1–6.

[2] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over http," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1842–1866, 2017.

[3] W. Shi, Q. Li, Q. Yu, F. Wang, G. Shen, Y. Jiang, Y. Xu, L. Ma, and G.-M. Muntean, "A survey on intelligent solutions for increased video delivery quality in cloud-edge-end networks," *IEEE Communications Surveys & Tutorials*, 2024.

[4] K. Khan, "Enhancing adaptive video streaming through ai-driven predictive analytics for network conditions: A comprehensive review," *International Transactions on Electrical Engineering and Computer Science*, vol. 3, no. 1, pp. 57–68, 2024.

[5] J. Woo, S. Hong, D. Kang, and D. An, "Improving the quality of experience of video streaming through a buffer-based adaptive bitrate algorithm and gated recurrent unit-based network bandwidth prediction," *Applied Sciences*, vol. 14, no. 22, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/22/10490

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[7] B. I. Grisci, M. J. Krause, and M. Dorn, "Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data," *Information sciences*, vol. 559, pp. 111–129, 2021.

[8] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 325–338. [Online]. Available: https://doi.org/10.1145/2785956.2787486

[9] Z. Zhong, M. Liu, L. Yang, Y. Wang, Y. Xu, and J.-N. Hwang, "Video streaming with kairos: An mpc-based abr with streaming-aware throughput prediction," in *Proceedings of the 35th Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 84–90. [Online]. Available: https://doi.org/10.1145/3712678.3721886

[10] S. Fernandes, J. Kelner, and D. Sadok, "An adaptive-predictive architecture for video streaming servers," *Journal of network and computer applications*, vol. 34, no. 5, pp. 1683–1694, 2011.

[11] C. Molnar, *Interpretable Machine Learning*, 2nd ed. Munich, Germany: Mucbook, 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[12] A. Ahmad, A. B. Mansoor, A. A. Barakabitze, A. Hines, L. Atzori, and R. Walshe, "Supervised-learning-based qoe prediction of video streaming in future networks: A tutorial with comparative study," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 88–94, 2021.

[13] C. G. Bampis, Z. Li, I. Katsavounidis, and A. C. Bovik, "Recurrent and dynamic models for predicting streaming video quality of experience," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3316–3331, 2018.

[14] I. Orsolic and M. Seufert, "On machine learning based video qoe estimation across different networks," in *2021 16th International Conference on Telecommunications (ConTEL)*. IEEE, 2021, pp. 62–69.

[15] S. Basso *et al.*, "Neubot: A software tool performing distributed network measurements to increase network transparency," Ph.D. dissertation, Politecnico di Torino, 2014.

[16] H. Yousef, J. Le Feuvre, and A. Storelli, "Abr prediction using supervised learning algorithms," in *2020 IEEE 22nd international workshop on multimedia signal processing (MMSP)*. IEEE, 2020, pp. 1–6.

[17] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, no. 3, pp. 916 – 954, 2008. [Online]. Available: https://doi.org/10.1214/07-AOAS148

[18] M. Christoph, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.

[19] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271616000265