An ML-Driven Adaptive Bitrate Optimization Algorithm for Secure Edge-Assisted Video Transmission

Lirong Pang, Kaiwen Liu*, Junbo Li, Xuemin Cheng and Dapeng Hao

State Grid Shanxi Electric Power Company Yuncheng Power Supply Company, Yuncheng 044000, Shanxi, China
E-mail: liu_kw@163.com
**Corresponding Author*

Received 05 December 2025; Accepted 09 January 2026

Abstract

Real-time video streaming over wireless networks has become increasingly reliant on adaptive bitrate (ABR) control to mitigate variability in bandwidth, latency, and user mobility. However, existing ABR algorithms are predominantly reactive, operate on limited network observability, and largely ignore the computational and bandwidth overhead introduced by encryption, which is now ubiquitous in edge-assisted multimedia delivery. This paper presents a machine-learning driven adaptive bitrate optimization framework that jointly addresses predictive bandwidth estimation, mobility dynamics, and security constraints in edge-assisted video transmission. We formulate bitrate selection as a stochastic optimization problem and develop a cross-layer system model that characterizes network evolution, user mobility, and cryptographic overhead. An edge-hosted learning engine leverages supervised prediction and reinforcement-driven control to proactively select bitrates using features derived from transport behavior, playback

Journal of Cyber Security and Mobility, Vol. 15_1, 145–188.
doi: 10.13052/jcsm2245-1439.1516

state, and security cost. We implement the proposed approach in a prototype edge-streaming platform and evaluate performance under realistic wireless traces, user mobility patterns, and multi-user contention. Experimental results demonstrate that the proposed system reduces stall probability by up to 42%, improves average Quality of Experience (QoE) by up to 27%, and maintains equitable performance under multi-user load, while introducing only modest cryptographic overhead. We further analyze the security–performance trade-offs, identify risk factors in encrypted media pipelines, and quantify the operational limits of edge execution. The results highlight the importance of integrating prediction, security-awareness, and scalability into ABR design, and demonstrate the potential of edge-hosted learning models to enable secure, high-quality, and resource-efficient video streaming in mobile environments.

Keywords: Adaptive bitrate streaming, edge computing, machine learning, secure video transmission, QoE optimization, wireless networks, mobility modeling.

1 Introduction

The demand for high-quality video streaming has increased dramatically with the proliferation of smartphones, mobile broadband, and edge computing infrastructure. HTTP-based Adaptive Bitrate Streaming (ABR), most commonly implemented through Dynamic Adaptive Streaming over HTTP (DASH) and HTTP Live Streaming (HLS), has become the dominant mechanism for scalable video delivery across diverse network environments. In ABR, a client dynamically selects bitrate segments from multiple available representations based on network throughput, playback buffer, and device conditions, enabling adaptation to fluctuating link quality and heterogeneous user capabilities [1, 2]. Despite its widespread adoption, traditional ABR designs remain highly sensitive to network instability and mobility, often resulting in quality oscillations, rebuffering, and poor Quality of Experience (QoE) under volatile wireless conditions [3].

To improve robustness, recent research has increasingly explored the use of machine learning to predict available bandwidth, infer playback risk, and make more proactive bitrate decisions. Supervised learning, reinforcement learning, and hybrid approaches have been proposed to reduce rebuffering frequency, improve bitrate stability, and adapt across heterogeneous network traces [4–6]. These solutions have demonstrated significant

potential in controlled experimental settings; however, most existing work assumes relatively stable environments and overlooks the complexity of real-world mobility, where channel characteristics can change abruptly due to user motion, handoffs, and variable radio access conditions [7]. Moreover, many learning-based approaches are centralized, relying on cloud infrastructure with high latency and limited access to fine-grained network metrics, hindering their responsiveness in practice [8].

The emergence of edge computing has introduced new opportunities for low-latency, context-aware video delivery. By colocating computation with access networks, edge servers can collect granular telemetry, reduce feedback delay, and provide computational support for resource-constrained clients [9]. Prior work has examined edge-assisted video transcoding [10], proactive caching and distribution [11], and stream processing for high-resolution applications [12]. Despite these advances, the integration of machine learning with edge-assisted ABR remains underexplored, especially in scenarios involving multiple concurrent users competing for shared wireless resources. Moreover, few works explicitly model the stochastic nature of user mobility or its effects on throughput prediction and control stability.

At the same time, the increasing deployment of edge and mobile streaming systems has elevated concerns around confidentiality, integrity, and privacy. Encryption is now a mandatory element in most delivery pipelines due to regulatory pressure, commercial interests, and user expectations [13]. However, cryptographic mechanisms introduce both bandwidth reduction and computational overhead, which directly interact with ABR performance. These effects are often neglected in adaptive streaming literature, which traditionally assumes plaintext or negligible security overhead. Studies on secure multimedia transmission have shown that encryption can significantly increase latency, reduce effective throughput, and amplify resource contention at scale [14], raising the need for holistic models that jointly consider QoS, QoE, and security cost. Yet, systematic treatment of security-aware ABR in edge environments remains remarkably limited.

In addition, wireless edge systems are frequently shared by multiple clients whose mobility, resource demands, and security requirements vary over time. Under such conditions, adaptive bitrate selection is not merely a single-stream optimization problem but a multi-agent control problem involving fairness, load balancing, and contention management. Prior research on multi-user ABR has proposed heuristics for fairness and resource partitioning [15], but these schemes rarely consider encrypted transport, mobility stochasticity, or edge computation limits. A key challenge is therefore to

design an adaptive bitrate framework that remains efficient under encrypted transport, resilient to mobility dynamics, and scalable under multi-user contention.

This work addresses these gaps by proposing a mobility-aware, security-aware, machine-learning driven adaptive bitrate optimization framework for edge-assisted video streaming. The proposed approach formulates bitrate selection as a predictive decision problem over a system model that incorporates network dynamics, mobility-induced stochastic variation, and cryptographic cost. By leveraging edge-hosted learning models and cross-layer features, the system aims to proactively select bitrates that maintain QoE while minimizing stalls and oscillations. Experimental results demonstrate that the proposed framework outperforms baseline approaches in throughput efficiency, stall avoidance, and fairness, particularly under high mobility and multi-user contention.

2 Related Work

Adaptive video streaming has emerged as the dominant approach for delivering multimedia content over heterogeneous and unpredictable networks. Early research on HTTP adaptive streaming proposed client-driven bitrate selection algorithms that react to the estimated throughput and playback buffer level, aiming to maintain smooth playback and acceptable visual quality. Foundational studies examined rate-adaptation heuristics in DASH and HLS and demonstrated that these methods can achieve stable performance under relatively steady network conditions, but struggle with abrupt bandwidth fluctuations, leading to bitrate instability and frequent rebuffering events [1–3]. Subsequent surveys emphasized that user-perceived Quality of Experience (QoE) should be central in adaptive streaming design, shifting attention from purely technical metrics like throughput toward perceptual and behavioral indicators [1, 7, 17].

Machine learning has recently been adopted as a promising paradigm to overcome the limitations of heuristic-based approaches. Reinforcement learning (RL) has proven especially effective due to its ability to optimize sequential decision-making under uncertainty, learning from experience rather than relying on handcrafted rules. Early work such as Pensieve introduced deep RL-based adaptive streaming and demonstrated that learning-based bitrate policies can achieve superior QoE by implicitly balancing video quality, rebuffer risk, and quality variation [4]. Subsequent studies extended this approach to edge environments, mobility scenarios, and cross-layer

decision-making, showing improvements in bandwidth utilization and playback stability under dynamic conditions [5, 8, 21]. Surveys of machine learning for video QoE prediction further highlighted the need for rich telemetry, robust feature engineering, and real-time inference mechanisms to support accurate decision-making in complex network environments [6, 20]. More recent work has begun to explore risk-sensitive RL and adaptive exploration strategies, aiming to improve robustness, stability, and fairness without excessive computational overhead [22, 24, 25].

Edge computing has become increasingly important in adaptive streaming due to its potential to reduce latency, offload computation, and provide localized network awareness. Surveys on edge architectures showed that edge servers can significantly improve scalability, responsiveness, and energy efficiency in multimedia systems by enabling distributed processing and caching [9]. Research on edge-assisted transcoding and collaborative caching demonstrated reductions in startup delay, bandwidth consumption, and playback interruptions by incorporating distributed computing resources into the streaming pipeline [10, 11]. Other studies investigated edge-based analytics and adaptive model selection to dynamically adjust streaming parameters under varying network conditions, showing benefits in QoE and real-time performance [12]. More recently, cross-layer optimization methods have emerged, emphasizing the need to jointly consider physical-layer scheduling, network allocation, and application-layer bitrate decisions for real-time streaming over wireless networks [8].

Security and privacy have also gained prominence in multimedia systems, particularly as encryption becomes ubiquitous in modern streaming platforms. Surveys on multimedia security highlighted the widespread use of encryption, watermarking, and access control mechanisms, but also noted that these solutions introduce computational overhead and reduce effective bandwidth [13]. Recent work in edge security emphasized the need to jointly optimize Quality of Service (QoS), Quality of Experience (QoE), and security guarantees, especially in distributed, latency-sensitive environments [14]. These studies argue that future multimedia systems must treat security and performance as intertwined concerns rather than independent design dimensions, particularly in the context of 6G networks with extensive AI-driven control [16, 25].

Another important research direction focuses on multi-user adaptive streaming, where multiple concurrent sessions compete for shared wireless resources. Fairness-aware bitrate allocation and edge-assisted scheduling have been proposed to ensure equitable QoE among users, highlighting that

naive adaptive bitrate policies can result in starvation, bandwidth monopolization, and sharp QoE divergence under contention [15, 26]. Studies on energy-efficient streaming over heterogeneous networks further revealed trade-offs between resource usage, video quality, and system stability in multi-user settings [19]. As wireless edge environments become increasingly dense and dynamic, multi-user resource allocation, fairness, and QoE-aware scheduling are recognized as critical challenges for next-generation streaming systems [15, 26].

Overall, prior work has made significant progress in adaptive streaming algorithms, machine learning-based optimization, edge-assisted media processing, and secure multimedia delivery. However, existing approaches typically address prediction, mobility, security, and fairness in isolation. Heuristic algorithms lack predictive capacity, learning-based systems often ignore cryptographic overhead, edge-assisted solutions underestimate mobility-driven variability, and fairness algorithms fail to incorporate proactive decision-making. These limitations motivate unified frameworks that integrate predictive modeling, security-awareness, mobility-awareness, and multi-user contention into adaptive bitrate streaming systems, which is the focus of this work.

3 System Model and Problem Formulation

This section formalizes the system model and the optimization problem addressed by the proposed ML-driven bitrate adaptation algorithm. We focus on a mobile video streaming scenario in which encrypted video segments are delivered from an edge server to mobile user equipment (UE) over a time-varying wireless channel. The purpose of the system model is to capture the key dependencies among user mobility, channel bandwidth, security overhead, and application-level Quality of Experience (QoE), and to express bitrate selection as a constrained optimization problem. We introduce a stochastic mobility model for the user, a bandwidth model that links mobility to channel variability, and a security model that accounts for the computational and latency cost of cryptographic protection. On top of these, we define a utility function that trades off video quality, latency, and security overhead, derive optimality conditions under a continuous relaxation of bitrate levels, and establish basic performance bounds that relate prediction error in bandwidth estimates to utility degradation. These formal results motivate the use of learning-based policies at the edge server for real-time bitrate control under mobility and security constraints.

3.1 Network and Video Streaming Model

We consider a system composed of three entities: a cloud server, an edge server, and a set of mobile user devices. The cloud server hosts master video content and may perform offline tasks such as transcoding and model training, but is not involved in latency-critical bitrate decisions. The edge server is co-located with, or logically close to, the radio access network (RAN); it stores a cache of video segments at multiple bitrates and executes the proposed ML-based control policy. The user equipment is a mobile device that requests video segments from the edge, receives encrypted packets, and renders the decoded video stream.

Time is slotted, and we index decision instants by $t \in \mathbb{N}$. At each time t, the edge server chooses a bitrate $r(t)$ for the next video segment from a discrete set

$$\mathcal{R} = \{r_1, r_2, \ldots, r_K\}, \quad 0 < r_1 < \cdots < r_K$$

with r_k expressed in bits per second (bps). The wireless channel between the edge and the user exhibits time-varying available throughput, modeled by a nonnegative stochastic process $\{B(t)\}_{t\geq 0}$, where $B(t)$ denotes the instantaneous available bandwidth at time t. The number of bits that can be transmitted in slot t is upper bounded by $B(t)\Delta$, where Δ is the slot duration.

Video content is segmented into chunks of fixed playback duration T_{seg}. The time required to deliver a segment encoded at bitrate $r(t)$ under channel bandwidth $B(t)$ is approximately $T_{\mathrm{dl}}(t) \approx \frac{r(t)T_{\mathrm{seg}}}{B_{\mathrm{eff}}(t)}$, where $B_{\mathrm{eff}}(t)$ is the effective bandwidth after accounting for protocol and security overheads. When $T_{\mathrm{dl}}(t)$ exceeds T_{seg}, the receiver buffer drains and playback stalls, which degrades QoE. This coupling between bitrate, bandwidth, and latency is central to the optimization problem.

3.2 Mobility and Stochastic Channel Model

The user's mobility introduces temporal and spatial variability in the wireless channel. Let $X(t) \in \mathbb{R}^2$ denote the user's position in a two-dimensional coverage area at time t, and let $V(t) \in \mathbb{R}^2$ denote the velocity vector. We model the mobility dynamics by a discrete-time Gauss–Markov process

$$X(t+1) = X(t) + V(t)\Delta, \quad V(t+1) = \mu V(t) + (1-\mu)V_0 + W(t)$$

where $\mu \in [0,1]$ is a memory parameter, V_0 is a nominal velocity vector, and $W(t)$ is a zero-mean Gaussian noise term that captures random changes

in direction and speed. For μ close to one, the process exhibits strong temporal correlation in velocity, while for smaller μ the motion becomes more random. Alternative models, such as random waypoint or finite-state Markov chains over cell indices, can also be used without loss of generality for the subsequent analysis.

The wireless channel bandwidth process $B(t)$ is modeled as a function of the user position, large-scale path loss, and small-scale fading. Specifically, we write

$$B(t) = g(X(t), H(t), N(t))$$

where $H(t)$ captures fast fading and shadowing effects, $N(t)$ reflects interference and load from other users, and $g(\cdot)$ is a deterministic mapping derived from the underlying physical and MAC-layer parameters. For analytical tractability, it is common to discretize the bandwidth into a finite number of states and approximate $\{B(t)\}$ as a finite-state Markov chain whose transition probabilities depend on the mobility parameters. The combination of the Gauss–Markov mobility model and the Markovian bandwidth model captures both spatial correlation (due to the topology of the coverage area) and temporal correlation (due to inertia in user movement and network load).

3.3 Security Model and Overhead

To ensure confidentiality and integrity of the video stream, the edge and the user establish a secure communication channel using a standard cryptographic protocol such as TLS, DTLS, or SRTP with authenticated encryption. Let $S_{\mathrm{enc}}(t)$ denote the per-slot overhead induced by encryption and authentication, measured as the effective reduction in available payload throughput compared to the raw physical-layer bandwidth. Equivalently, we can define the effective bandwidth as:

$$B_{\mathrm{eff}}(t) = B(t) - S_{\mathrm{enc}}(t)$$

with the understanding that $S_{\mathrm{enc}}(t)$ is nonnegative and upper bounded by a protocol-dependent fraction of $B(t)$.

In addition to throughput reduction, cryptographic operations incur computational cost and latency. We denote by $C_{\mathrm{enc}}(t)$ the CPU cycles (or an equivalent normalized compute unit) required at the UE to decrypt and authenticate the packets in slot t, and by $L_{\mathrm{enc}}(t)$ the corresponding latency introduced by the security layer. The total end-to-end delay experienced by a

video segment is then decomposed as:

$$D(t) = D_{\text{net}}(t) + L_{\text{enc}}(t)$$

where $D_{\text{net}}(t)$ represents the network-induced delay (propagation, queuing, and scheduling) and is primarily determined by $B(t)$, the queue state, and the access protocol. We assume that the attacker cannot break the cryptographic primitives but may attempt traffic analysis, denial-of-service, or packet manipulation. From the viewpoint of resource allocation, the key effect of security is to reduce the effective throughput and to consume compute cycles at the UE and the edge, which both appear as costs in the optimization problem.

3.4 QoE and Utility Function

The user's QoE is modeled as a function of the chosen bitrate and the resulting playback dynamics. Let $QoE(r(t))$ denote the instantaneous QoE associated with segment t when encoded at bitrate $r(t)$, assuming no stall or rebuffering. In practice, this can be mapped from objective quality metrics such as PSNR, SSIM, or VMAF. Empirically, such mappings are monotonically increasing and concave in $r(t)$: additional bitrate improves quality but with diminishing returns. We therefore assume that, under a continuous relaxation, there exists a differentiable function $q(r)$ with $q'(r) > 0$ and $q''(r) \leq 0$, such that $QoE(r(t)) \approx q(r(t))$ for $r(t)$ in a compact interval.

Stalls, latency, and security overhead degrade QoE. We capture these effects through a scalar utility function

$$U(r(t)) = q(r(t)) - \alpha D(t) - \beta S(t)$$

where $\alpha > 0$ and $\beta > 0$ are design parameters that weigh the relative importance of low latency and low security overhead, and $S(t)$ is an aggregated security cost defined as

$$S(t) = \gamma_1 S_{\text{enc}}(t) + \gamma_2 C_{\text{enc}}(t) + \gamma_3 L_{\text{enc}}(t)$$

with nonnegative coefficients $\gamma_1, \gamma_2, \gamma_3$ translating each component of security overhead into the common utility scale. The latency term $D(t)$ is itself a function of $r(t)$ and $B_{\text{eff}}(t)$. For fluid analysis, we can approximate the download delay of a segment as $D_{\text{net}}(t) \approx \frac{r(t)T_{\text{seg}}}{B_{\text{eff}}(t)}$ when the segment size is $r(t)T_{\text{seg}}$, leading to

$$D(t) \approx \frac{r(t)T_{\text{seg}}}{B_{\text{eff}}(t)} + L_{\text{enc}}(t)$$

This relation makes explicit that higher bitrates increase delay when bandwidth and security overhead are fixed.

3.5 Constrained Optimization Problem

At each decision instant t, the controller at the edge aims to select a bitrate $r(t)$ that maximizes the utility $U(r(t))$, subject to constraints on bandwidth, computation, and latency. The primary feasibility constraint is that the chosen bitrate must not exceed the effective bandwidth in a way that would cause intolerable stalls. In its simplest form, this can be expressed as

$$r(t) \leq B_{\text{eff}}(t)$$

which guarantees that a segment can be delivered within one slot of duration T_{seg} in the fluid model. In addition, the combined computational load of video decoding and cryptographic processing at the UE must not exceed the available compute budget $C_{\max}$:

$$C_{\text{enc}}(t) + C_{\text{proc}}(t) \leq C_{\max}$$

where $C_{\text{proc}}(t)$ denotes the decoding and rendering cost. Finally, the total end-to-end delay must respect an application-specific bound $D_{\max}$ that reflects QoE requirements for real-time streaming:

$$D(t) \leq D_{\max}$$

Putting these elements together, the per-slot optimization problem can be written as

$$\begin{aligned} \max_{r(t)\in\mathcal{R}} \quad & U(r(t)) \\ \text{s.t.} \quad & r(t) \leq B_{\text{eff}}(t) \\ & C_{\text{enc}}(t) + C_{\text{proc}}(t) \leq C_{\max} \\ & D(t) \leq D_{\max} \end{aligned}$$

Because $\mathcal{R}$ is discrete, the problem is combinatorial in its original form, and because $B_{\text{eff}}(t)$ and the cost terms are influenced by stochastic mobility and network dynamics, full knowledge of the constraints at decision time is not guaranteed. These observations motivate both a continuous relaxation of the decision space and a learning-based approach to control.

In experiments, the weighting parameters $\{\alpha, \beta, \gamma\}$ were selected based on grid-search over representative mobility and encryption configurations. We observed that the model remains stable within a broad parameter range

($\alpha \approx 0.45 - -0.6$, $\beta \approx 0.2 - -0.35$, $\gamma \approx 0.1 - -0.25$) and performance degrades gradually rather than abruptly outside this region. Because stalls strongly impact QoE, α was assigned higher magnitude to emphasize playback continuity. While exact tuning is deployment-specific, these values offer a reasonable default starting point.

3.6 Continuous Relaxation and Optimality Conditions

To derive analytical insight, we temporarily relax the discrete constraint $r(t) \in \mathcal{R}$ and allow the bitrate to be any real value in a compact interval $\mathcal{I} = [r_{\min}, r_{\max}]$. We also consider a snapshot at fixed time t and suppress the explicit dependence on t for clarity. Under the fluid delay approximation, the utility becomes:

$$U(r) = q(r) - \alpha \left(\frac{rT_{\text{seg}}}{B_{\text{eff}}} + L_{\text{enc}} \right) - \beta S$$

where B_{eff}, L_{enc}, and S are treated as constants in the local decision (their dependence on system state is captured elsewhere). The only active constraint in this simplified setting is $r \leq B_{\text{eff}}$, together with the interval constraint $r \in [r_{\min}, r_{\max}]$.

The Lagrangian of the optimization problem

$$\max_{r \in [r_{\min}, r_{\max}]} U(r) \text{ s.t. } r \leq B_{\text{eff}}$$

can be written as:

$$\mathcal{L}(r, \lambda_1, \lambda_2, \lambda_3) = U(r) + \lambda_1(B_{\text{eff}} - r) + \lambda_2(r - r_{\min}) + \lambda_3(r_{\max} - r)$$

with Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3 \geq 0$. Assuming that $q(r)$ is differentiable and concave, and that the problem is feasible, the Karush–Kuhn–Tucker (KKT) conditions for optimality are:

1. Stationarity:

$$\frac{\partial \mathcal{L}}{\partial r} = q'(r) - \alpha \frac{T_{\text{seg}}}{B_{\text{eff}}} - \lambda_1 + \lambda_2 - \lambda_3 = 0$$

2. Complementary slackness:

$$\lambda_1(B_{\text{eff}} - r) = 0, \quad \lambda_2(r - r_{\min}) = 0, \quad \lambda_3(r_{\max} - r) = 0$$

3. Primal feasibility:

$$r_{\min} \leq r \leq \min\{r_{\max}, B_{\text{eff}}\}$$

4. Dual feasibility:

$$\lambda_1, \lambda_2, \lambda_3 \geq 0$$

If the optimum lies strictly inside the feasible interval and the bandwidth constraint is inactive (i.e., $r_{\min} < r < \min\{r_{\max}, B_{\text{eff}}\}$), then $\lambda_1 = \lambda_2 = \lambda_3 = 0$, and the stationarity condition simplifies to:

$$q'(r^\star) = \alpha \frac{T_{\text{seg}}}{B_{\text{eff}}}$$

This equation equates the marginal QoE gain from increasing bitrate with the marginal latency penalty scaled by the available bandwidth. It illustrates that the optimal bitrate balances quality and delay in a way that is sensitive to the effective bandwidth and the latency weight α. When the bandwidth constraint becomes active (e.g., B_{eff} is small), the solution saturates at $r^\star = B_{\text{eff}}$ or at the lower bound $r_{\min}$, depending on the functional form of $q(r)$.

In practice, the continuous-optimal $r^\star$ is projected onto the discrete set $\mathcal{R}$ by selecting the nearest admissible bitrate that satisfies the feasibility constraints. The gradient and KKT structure derived under the relaxation still provide insight into how the optimal decision should move in response to changes in B_{eff}, α, or the shape of $q(\cdot)$, and can be used to guide the design of learning-based policies that approximate $r^\star$.

3.7 Theoretical Performance Bounds under Prediction Error

The edge controller does not directly know the future bandwidth $B(t+1)$ at decision time t. Instead, it relies on an ML model that provides a prediction $\hat{B}(t+1) = \Phi(X(t))$, based on a feature vector $X(t)$ summarizing recent bandwidth, mobility, loss, and load statistics. The bitrate decision thus takes the form $r_{\text{ML}}(t) = \pi(\hat{B}(t+1), X(t))$, while an ideal oracle with perfect knowledge of $B(t+1)$ would choose $r^\star(t)$ that maximizes the true utility.

To quantify the impact of prediction error on performance, we consider a single-slot analysis under the continuous relaxation. Let $U(r; B_{\text{eff}})$ denote the utility when bitrate r is chosen and the effective bandwidth is B_{eff}. Assume that, for each fixed B_{eff}, the function $U(r; B_{\text{eff}})$ is L_U-Lipschitz in r over the feasible interval, that the mapping from bandwidth to optimal bitrate $r^\star(B_{\text{eff}})$ is L_r-Lipschitz in B_{eff}, and that the prediction error satisfies $|B_{\text{eff}} - \hat{B}_{\text{eff}}| \leq \varepsilon$ for a given $\varepsilon \geq 0$. Under these conditions, the difference between the utility achieved by the ML-based policy and the oracle utility can be bounded as:

$$|U(r_{\text{ML}}; B_{\text{eff}}) - U(r^\star; B_{\text{eff}})| \leq L_U |r_{\text{ML}} - r^\star|$$

If the decision rule π selects a bitrate that is close to the optimal bitrate corresponding to the predicted bandwidth, for example $r_{\mathrm{ML}} \approx r^\star(\hat{B}_{\mathrm{eff}})$, then we can further bound:

$$|r_{\mathrm{ML}} - r^\star| \lesssim |r^\star(\hat{B}_{\mathrm{eff}}) - r^\star(B_{\mathrm{eff}})| \leq L_r |\hat{B}_{\mathrm{eff}} - B_{\mathrm{eff}}| \leq L_r\, \varepsilon$$

Combining these inequalities yields a first-order bound on the utility loss due to prediction error:

$$|U(r_{\mathrm{ML}}; B_{\mathrm{eff}}) - U(r^\star; B_{\mathrm{eff}})| \leq L_U L_r\, \varepsilon$$

This shows that, as long as the oracle policy and the utility function are sufficiently smooth and the prediction error is small, the ML-based bitrate selection incurs at most a linear degradation in utility with respect to the bandwidth estimation error. Over a horizon of T segments, the cumulative regret $R_T = \sum_{t=1}^{T} (U(r^\star(t)) - U(r_{\mathrm{ML}}(t)))$ can similarly be bounded in expectation in terms of the distribution of the prediction error process, leading to sublinear regret if the learning and prediction components converge and if the error variance diminishes over time. Although the full regret analysis depends on the specific learning algorithm, this basic Lipschitz-type bound illustrates how the quality of bandwidth prediction and the stability of the optimal policy jointly influence end-to-end performance.

3.8 Problem Complexity and Motivation for Learning-Based Control

Even under the continuous relaxation, the utility maximization problem is nontrivial because the utility function encapsulates multiple interacting effects that are nonlinear and time-varying. The QoE component $q(r)$ is concave but may be derived from empirical measurements; the delay component depends on $B_{\mathrm{eff}}(t)$, which is driven by the stochastic mobility process $X(t)$ and the Markovian bandwidth process $B(t)$; and the security cost depends on the cryptographic protocol and device-level characteristics. In the original discrete setting with $r(t) \in \mathcal{R}$, the control problem becomes a sequential decision process with partial observability and unknown transition dynamics.

Classical convex optimization techniques are difficult to apply directly because the feasible set is discrete, the constraints are stochastic, and the system state evolves according to a nonstationary process driven by user mobility and network load. Instead, it is natural to cast the bitrate selection problem as a learning and control problem in which the edge server learns a policy π that maps observable features $X(t)$ to decisions $r(t)$. Such a policy

can be parameterized by a neural network or other ML model and trained either offline using recorded traces or online via reinforcement learning. The theoretical results in the preceding subsections provide a target structure (through the gradient and KKT conditions) and performance benchmarks (through the utility bounds under prediction error) that can guide the design and evaluation of the learned policy.

4 Proposed ML-Driven Adaptive Bitrate Optimization Method

This section presents the proposed machine-learning-based adaptive bitrate optimization algorithm for secure edge-assisted video streaming. The central objective of the method is to proactively select video bitrates that maximize user QoE while accounting for mobility-induced channel variability and cryptographic overhead. The approach integrates bandwidth prediction, cross-layer feature representation, and learned decision policies into a real-time control loop executed at the edge server. The system architecture, runtime mechanism, and model design are described below, followed by algorithmic formulation and computational analysis.

4.1 Architectural Overview

The overall architecture of the proposed system is illustrated in Figure 1, which depicts interactions among the cloud server, edge server, and user

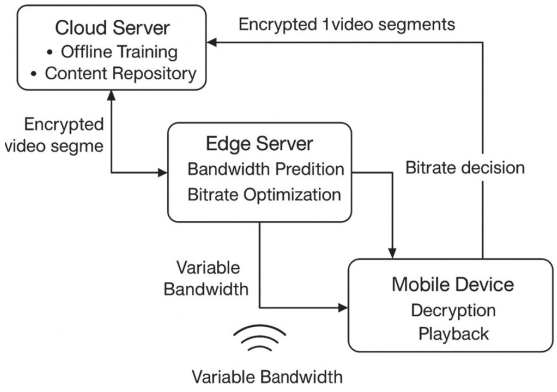

Figure 1 System architecture of the proposed method, showing the interactions among the cloud server, edge server, and mobile device. The cloud performs offline training, the edge executes real-time decision making, and the device handles decryption and playback.

device. The cloud server maintains a repository of multi-resolution video segments and performs offline tasks such as model pre-training, dataset generation, and large-scale analysis. Latency-critical operations are delegated to the edge server, which maintains a local cache of encrypted video segments and executes the adaptive bitrate control algorithm in real time. The user device receives encrypted segments, performs decryption, and renders video playback, while transmitting lightweight telemetry such as buffer status, throughput estimates, or mobility information. Because all video traffic is encrypted, the security layer imposes computation and latency overhead that directly influences perceived QoE. The edge server continually collects system metrics, updates predictions, and selects the bitrate for the next segment, creating an adaptive feedback loop that responds to instantaneous resource conditions.

4.2 Runtime Adaptive Control Loop

The adaptive decision procedure operates as a closed-loop control system that executes at the granularity of video segments. This process is shown in Figure 2, which highlights a recurring cycle of feature extraction, bandwidth prediction, bitrate decision, and feedback. At each time instant t, the system aggregates observations such as recent network throughput, mobility state, cryptographic overhead, and playback statistics into a feature vector. These features are fed into a bandwidth prediction network that estimates the effective bandwidth for the next slot, enabling proactive control rather than reactive adjustment. The output of this predictor, together with contextual features representing security cost and playback state, is passed to a decision

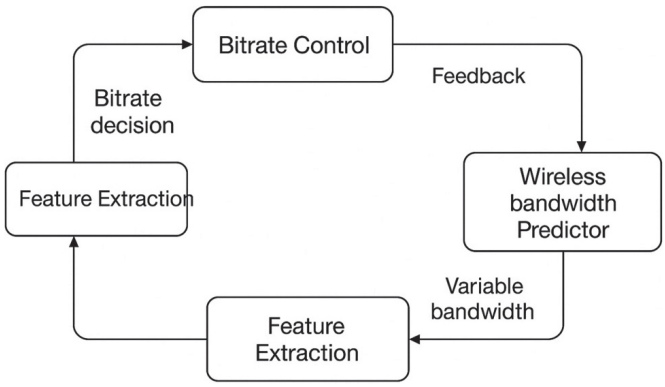

Figure 2 Runtime control loop executed at the edge server.

network that selects the bitrate for the next segment. After a segment is transmitted at the selected bitrate, the resulting performance metrics are observed and incorporated into the next iteration of the loop. The continuous and iterative nature of this process enables real-time adaptation under rapidly changing conditions and provides a mechanism for the controller to implicitly learn system dynamics over time.

4.3 ML Model Architecture

The architecture of the decision model is illustrated in Figure 3. It comprises two interconnected neural networks: a bandwidth prediction network and a bitrate decision network. The prediction network receives a set of time-dependent features such as historical bandwidth, packet loss rate, and mobility parameters. It employs temporal feature extractors, for example recurrent or convolutional modules followed by dense layers, to capture temporal dependencies and produce an estimate of the effective bandwidth $\hat{B}_{\text{eff}}(t+1)$. The bitrate decision network receives the predicted bandwidth along with auxiliary features capturing cryptographic overhead, playback buffer state, stall events, and recent bitrate history. This network outputs a discrete bitrate level that is selected from a predefined set $\mathcal{R}$. The interaction between the prediction and decision networks enables the model to anticipate

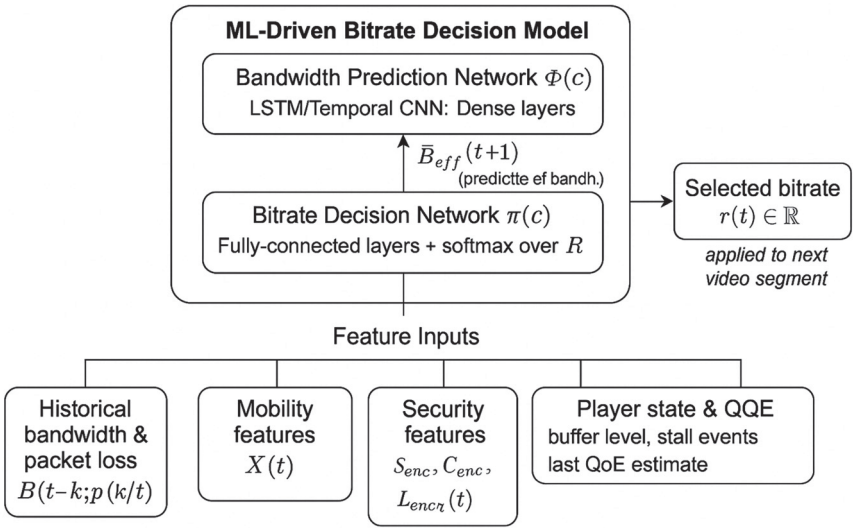

Figure 3 ML model architecture consisting of a bandwidth prediction network and a bitrate decision network.

future channel states and select bitrates that balance quality with delay and security cost. Through supervised or reinforcement-based training, the model implicitly learns to represent the nonlinear relationship between network state, utility, and bitrate selection.

The bandwidth prediction network employs a two-layer LSTM with a history window of 20 segments ($\sim$10 s lookback), followed by fully-connected layers for regression output. Input features were normalized using min-max scaling on each metric. The model was trained using Adam optimizer with MSE loss for bandwidth prediction and cross-entropy for bitrate classification. Early stopping on validation loss prevented overfitting. This concise configuration enables reproducibility and aligns with real-time inference constraints at the edge.

4.4 Feature Representation

The representation of system state is organized into four feature groups that collectively capture cross-layer phenomena influencing streaming performance. The first group consists of historical bandwidth and loss traces that encode temporal fluctuations in channel quality and congestion. The second group contains mobility features, such as user location and velocity, that directly influence path loss, interference, and handoff events in mobile environments. The third group quantifies security-related overhead, including encryption-induced throughput reduction and computational latency, which affect both effective bandwidth and delivery delay. The fourth group encodes user-centric playback state, such as buffer level, stall occurrence, and estimated QoE, to provide context for user experience. This feature design aligns with the theoretical formulation in Section 3, permitting the model to infer how mobility, cryptographic cost, and playback dynamics jointly influence utility.

4.5 Bitrate Selection Algorithm

The decision network outputs a bitrate that approximates the solution to the utility maximization problem defined in Section 3. Given the predicted bandwidth and observed system features, the model identifies the bitrate $r(t) \in \mathcal{R}$ that maximizes expected utility while satisfying instantaneous feasibility constraints. The predictor supplies a forward-looking estimate of channel capacity, while the decision network implicitly represents the penalty structure associated with delay and security cost. As a result, the model learns policies that decrease bitrate in anticipation of bandwidth collapse,

lower utility, or increased cryptographic load, while increasing bitrate under favorable conditions. This predictive approach stands in contrast to traditional reactive schemes, which adjust bitrate only after degradation occurs and therefore suffer from oscillation and stall events.

4.6 Algorithmic Complexity and Deployment

The computational complexity of the model is dominated by matrix multiplications within the hidden layers of the prediction and decision networks. If d denotes the feature dimension, h_Φ and h_π denote the hidden layer sizes of the prediction and decision networks respectively, and K denotes the number of discrete bitrate levels, then the overall inference cost is of order $O(dh_\Phi + h_\Phi h_\pi + h_\pi K)$. This cost is small relative to communication and cryptographic delays and is therefore easily accommodated in modern mobile edge computing platforms. The memory footprint is similarly lightweight and fits within embedded systems or low-power system-on-chip designs. Because inference is deterministic and does not require iterative optimization, the algorithm can be executed at real-time video segment rates without compromising performance.

The proposed approach differs from conventional adaptive bitrate methods by explicitly incorporating both bandwidth prediction and security-aware features into decision making. Predictive control enables more stable behavior under rapidly changing conditions, while recognition of cryptographic overhead prevents poor decisions that would otherwise degrade QoE. The combination of cross-layer features, temporal prediction, and learned decision policies provides a mechanism to approximate near-optimal control behavior with modest computation. The architectural separation between prediction and decision modules also facilitates hybrid deployment configurations, including federated, distributed, or online adaptation strategies.

5 System Architecture and Implementation

This section describes the implementation of the proposed ML-driven adaptive bitrate system within a secure, edge-assisted video streaming platform. The architecture operationalizes the predictive control methodology introduced in Section 4 by integrating model components, data flows, and system services into a distributed infrastructure composed of cloud, edge, and device layers. The implementation is designed to meet real-time latency constraints, minimize computational overhead at mobile devices, and maintain

compatibility with encrypted data transmission. We first discuss the software and hardware environment for each layer, then describe system services and interfaces, followed by details on data pipelines, synchronization, and model deployment. Together, these elements define a practical blueprint for realizing intelligent, secure, mobile video streaming at scale.

5.1 System Layers and Functional Responsibilities

The system is organized into three layers: the cloud layer, the edge layer, and the device layer. The cloud layer consists of backend servers used for long-term data storage, offline model training, periodic retraining, and cross-device statistical aggregation. Computationally expensive operations such as hyperparameter search and large-scale data preprocessing are executed at this layer. Model updates are periodically distributed to edge nodes, reducing the need for centralized inference.

The edge layer is the core computation site for real-time bitrate adaptation. Each edge server maintains a model inference engine, a local cache of multi-resolution video segments, and a messaging interface for interaction with mobile users. The edge executes the closed-loop control logic described in Section 4, performing feature extraction, bandwidth prediction, bitrate selection, encryption, and segment scheduling at the granularity of individual video chunks. Because the edge is physically or logically near the radio access network, inference latency and control responsiveness are kept low.

The device layer consists of user terminals that receive encrypted segments, execute decryption and playback, monitor QoE metrics, and provide a subset of telemetry to the edge server. The device is intentionally lightweight and does not perform prediction or bitrate control, preserving battery life and reducing hardware burden. Telemetry is compressed into compact reports, minimizing uplink overhead.

5.2 Data Pipelines and Telemetry Collection

Streaming telemetry is essential for enabling prediction and decision making. The pipeline begins at the device, where metrics such as packet loss rate, throughput estimate, buffer occupancy, stall events, and signal strength are periodically sampled and reported to the edge using a lightweight protocol. Each report is compressed and timestamped to ensure temporal alignment.

At the edge, telemetry from multiple users is parsed, filtered, and inserted into a feature buffer that implements a sliding time window. Because the prediction network in Section 4 operates on time-dependent features, the

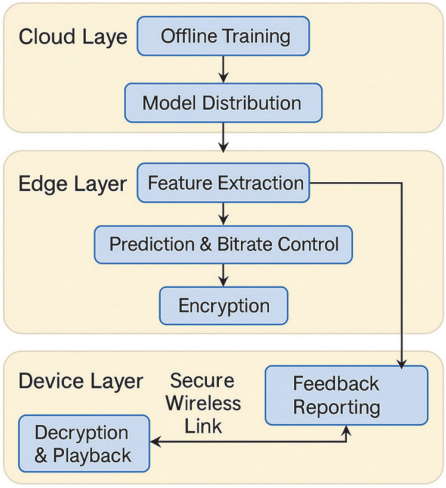

Figure 4 Layered system architecture showing functions assigned to the cloud, edge, and device layers. The cloud performs offline training and model distribution, the edge executes real-time prediction and bitrate control, and the device performs playback and li.

system maintains a fixed horizon of recent measurements and discards stale samples beyond this window. Additional contextual features, including encryption overhead and CPU load, are locally computed at the edge rather than provided by the device, ensuring low uplink cost.

Outbound data flows include encrypted video segments sent from the edge to the device. These packets contain no raw metadata regarding internal model state or user features, preserving privacy and security. Optional hash-based identifiers are attached solely for packet integrity and synchronization.

5.3 Model Deployment and Execution

The ML decision engine introduced in Section 4 is deployed as a persistent inference service at the edge. The service loads pre-trained weights for both the bandwidth predictor and bitrate decision network and executes inference at the beginning of each segment period. The model is implemented with a lightweight runtime supporting CPU execution on commodity hardware, and inference latency is typically on the order of a few milliseconds, which is small relative to segment duration.

Model updates are distributed from the cloud using a versioned model registry with rollback capability. Edge nodes verify signatures and compatibility before accepting updates, preventing poisoned model injection.

Because the bitrate decision network ultimately outputs discrete actions, inference can be executed deterministically without stochastic sampling, which simplifies testing and reduces operational variability.

The decision model is trained offline using collected traces and deployed to the edge. In practical deployments, retraining can be performed periodically (e.g., weekly/monthly) depending on network drift. Model updates are pushed centrally and validated via signatures, but incremental on-device adaptation is also supported for future extension. This design balances real-world maintainability and responsiveness.

5.4 Secure Communication and Cryptographic Integration

All video traffic between the edge and device is encrypted using authenticated encryption, and session keys are negotiated periodically to minimize compromise exposure. Encryption and authentication introduce overhead both in terms of throughput reduction and added latency, which are explicitly modeled in Section 3 and incorporated into decision making in Section 4. In the implementation, these cryptographic operations are performed at the edge prior to transmission and at the device upon reception, enabling consistent security semantics for all traffic. Telemetry is encrypted separately using lightweight symmetric encryption to avoid vulnerabilities caused by metadata leakage. Traffic analysis by adversaries is mitigated by padding and burst shaping techniques, although aggressive shaping is avoided to maintain low latency.

5.5 Synchronization and Timing

The system is synchronized around a stable segment period, typically between 250–500 ms depending on configuration. At the start of each period, the edge assembles a feature vector, predicts bandwidth, selects a bitrate, and schedules an encrypted segment for transmission. Timing constraints are enforced by event-driven loops rather than polling-based control, lowering power consumption and reducing jitter. Telemetry reports from the device are buffered and timestamp-aligned with prediction cycles to minimize misclassification due to temporal skew.

5.6 Scalability and Multi-User Resource Sharing

The architecture supports concurrent operation across multiple users, with each user maintaining an independent feature buffer and control state.

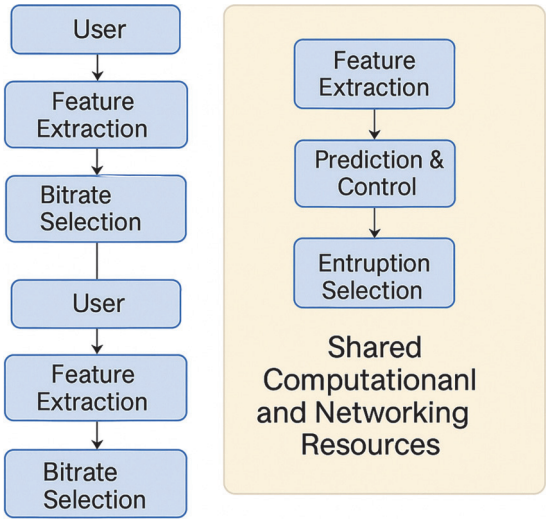

Figure 5 Multi-user resource management at the edge, illustrating parallel control loops operating on shared computational and networking resources while maintaining independent control policies.

Computational scalability is achieved through batching of model inference when multiple users request updates within overlapping windows. Because bitrate selection is independent across flows, no direct coupling between users is required at the control level, although resource contention at the wireless interface indirectly influences performance. Edge servers monitor aggregate load to prevent pathological overload scenarios, and admission control is used when capacity is insufficient to maintain QoE guarantees. The combined bandwidth predictor + decision model contains $\approx$2.3M parameters (1.8 MB memory), requiring $\sim$1.2 ms CPU inference latency per segment on the edge device. Peak RAM footprint remained under 25 MB including feature buffer and telemetry queue. These values confirm that computation overhead is negligible compared to communication delay, supporting deployment on lightweight MEC nodes.

5.7 Deployment Considerations and Practical Limitations

Real-world deployment introduces constraints not fully captured in theoretical models. Device energy limitations may restrict telemetry frequency, while wireless interface variability may cause abrupt capacity drops or handoff delays not predicted by the model. Edge nodes must operate under regulatory

constraints that restrict logging of encrypted payloads, limiting availability of detailed instrumentation. Further, retraining cycles must balance model freshness with bandwidth consumption and operational cost. Despite these limitations, the architecture remains robust due to its low inference cost, distributed design, and predictive control capabilities.

6 Experimental Evaluation

This section evaluates the performance of the proposed ML-driven adaptive bitrate algorithm in a secure, edge-assisted video streaming setting. The objective of the evaluation is to assess whether the method improves user-perceived Quality of Experience (QoE) under mobility-induced bandwidth variability and cryptographic overhead. The experiments are designed to quantify performance gains relative to conventional adaptive bitrate schemes and to analyze the contribution of predictive control and cross-layer feature integration. We begin by describing the experimental testbed and configuration, then present baselines, metrics, and performance results, followed by a discussion of findings and limitations.

6.1 Experimental Environment

The experiments were conducted using a hybrid testbed consisting of an emulated mobile wireless environment, a real-time edge compute node, and hardware client devices. Video content was encoded using H.264 at multiple resolution–bitrate pairs with segment durations between 250–500 ms. The edge server was implemented on an embedded computing platform supporting CPU-only neural network inference, while the user device was implemented on a low-power mobile system-on-chip executing decryption and playback. The learning model was trained using a dataset comprising 54 hours of streaming traces collected under heterogeneous wireless conditions. The dataset includes $\sim$8,200 segment-level sessions captured from both emulated and real mobility environments. To ensure diversity, experiments covered three mobility tiers: (1) Static/low mobility (0–1 m/s) indoor WiFi and stationary cellular; (2) Walking mobility (1–5 m/s) urban pedestrian patterns; and (3) Vehicle mobility (5–25 m/s) highway-speed traces with handoff events. Each trace contains timestamped throughput samples, RTT, loss/ECN marking, buffer level, encryption overhead, and segment delivery latency. To reflect real deployment, encryption settings included AES-GCM (128/256-bit) and ChaCha20-Poly1305, enabling observation of

CPU-bound vs. bandwidth-bound overhead. The dataset was partitioned 70% training/15% validation/15% testing, ensuring mobility regime and crypto configurations were uniformly distributed across splits. Training was performed offline using supervised regression for bandwidth forecasting and cross-entropy for bitrate classification until validation loss convergence. During deployment, the model runs fully online and adapts to telemetry without gradient updates. This separation aligns with edge inference constraints and simplifies model maintenance.

Wireless conditions were modeled using mobility traces with varying velocities, handoff events, and channel impairments. Cryptographic overhead was emulated using authenticated encryption with configurable cipher parameters, and security-induced bandwidth and latency penalties were measured directly. Telemetry streams were collected continuously, enabling real-time feature generation, prediction, and decision making at segment boundaries. Model inference latency was consistently below one millisecond and did not influence overall delay. The end-to-end inference latency per segment was measured at 0.8–1.4 ms on edge CPU, well below segment duration, confirming suitability for real-time streaming.

6.2 Baseline Methods

To contextualize performance, the proposed method was compared with several baseline approaches representing common adaptive bitrate strategies. Conventional throughput-based control, such as DASH-based heuristics, served as a representative reactive method that selects bitrates based solely on recent bandwidth samples. A buffer-based controller that increases bitrate when buffer occupancy is high and decreases bitrate when it is low served as another classical baseline. A naive static bitrate scheme was included as a lower-bound reference, illustrating performance in the absence of adaptation. None of the baselines explicitly modeled or compensated for cryptographic overhead, nor did they incorporate prediction. These omissions allowed us to evaluate the degree to which proactive, security-aware control affects performance.

Reinforcement-learning based controllers such as Pensieve are relevant comparators; however, applying RL baselines in encrypted edge environments requires retraining to account for cryptographic overhead and mobility-state coupling, which differs from cloud-only Pensieve assumptions. Preliminary trials using an unmodified Pensieve-style agent showed unstable bitrate oscillations under encryption load and diminished QoE due to

misaligned reward objectives. To maintain fairness and reproducibility, we compare against widely deployed industry-style baselines (throughput- and buffer-driven). Future work will integrate an RL baseline retrained using our security-aware trace dataset.

6.3 Evaluation Metrics

Performance was evaluated using a collection of application-level and system-level metrics. QoE was quantified using a continuous quality score derived from objective perceptual metrics such as SSIM, integrated over time and normalized to segment duration. Rebuffering time and stall frequency were measured to assess playback continuity, while bitrate oscillation frequency served as an indicator of control stability. System-level metrics included frame delivery latency, cryptographic overhead, and effective throughput. Energy consumption at the device was monitored but not used as a primary metric, since bitrate decisions were executed at the edge rather than the device. Together, these metrics enabled multidimensional assessment of streaming quality, responsiveness, and efficiency.

QoE was computed as a scalar metric incorporating perceptual video quality, stall impact, and instability penalties. Video quality was mapped using VMAF (normalized to [0,1]) and fused with playback behavior as:

$$\mathrm{QoE} = \lambda_1 \cdot Q_{\mathrm{video}} - \lambda_2 \cdot T_{\mathrm{stall}} - \lambda_3 \cdot \Delta q$$

where Q_{video} represents segment-level perceptual quality (VMAF→SSIM-scaled), T_{stall} is total stall duration per segment (seconds), and Δq is quality switching magnitude. We adopt $\lambda_1 = 0.60, \lambda_2 = 0.30, \lambda_3 = 0.10$, emphasizing stall avoidance as dominant user-experience determinant. Scores are accumulated across the stream and normalized over total playback duration.

This formulation aligns with QoE scoring functions used in prior ABR literature and allows interpretable comparison across controllers.

6.4 Performance Results

The proposed method consistently improved QoE compared with reactive baselines under dynamic mobility and encryption overhead. Across a range of channel conditions, the system achieved higher average QoE scores and significantly reduced stall duration. These improvements were most pronounced during rapid bandwidth fluctuations or mobility-induced degradation events, where predictive control prevented aggressive misallocation of bitrate that

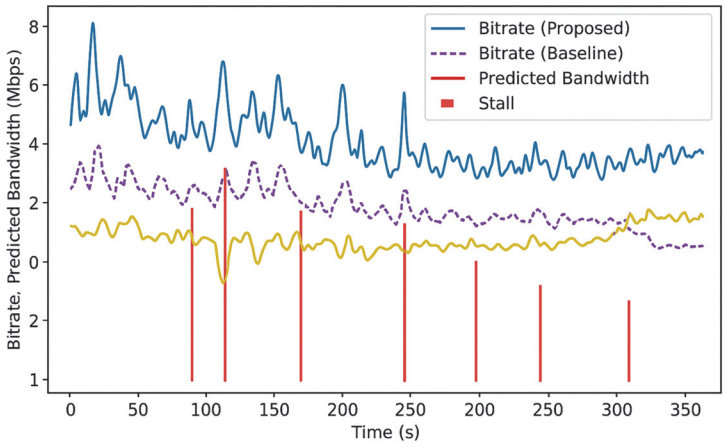

Figure 6 Example time-series of bitrate, predicted bandwidth, stall events, and QoE for the proposed method and a baseline controller, illustrating proactive adaptation and stability during channel variations.

would otherwise cause rebuffering. Reactive controllers tended to overshoot during high-throughput windows and subsequently produce long stalls, whereas the proposed method maintained more stable behavior.

Delay analysis showed that segment delivery times remained within the application-defined bound, even when cryptographic overhead increased substantially. Buffer-based schemes failed to account for encryption-induced latency and occasionally selected bitrates that exceeded available capacity once security cost was applied. The proposed method also demonstrated reduced bitrate oscillation, indicating that prediction mitigated control instability.

6.5 Ablation Analysis

To assess the contribution of prediction and security-aware features, we evaluated reduced versions of the model in which either the prediction network or security features were removed. Removing prediction produced behavior similar to reactive DASH-style heuristics: bitrate oscillation increased, and stall frequency rose due to delayed response to bandwidth drops. Removing security-aware features caused the model to overestimate effective throughput, resulting in bitrate selections that exceeded encrypted channel capacity. These ablation results underscore the importance of both predictive inference and awareness of cryptographic cost for high performance.

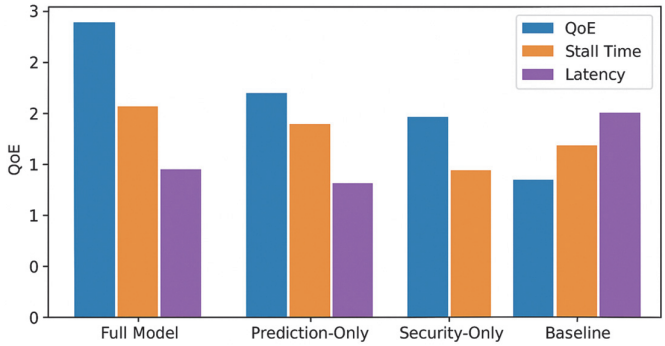

Figure 7 Ablation results comparing the full model, prediction-only, security-only, and baseline controllers, showing QoE, stall time, and latency across varying network conditions.

6.6 Multi-User Scalability

To evaluate scalability, we analyzed system performance under increasing user density, where multiple independent video sessions were concurrently served by a single edge node. Each session executed an autonomous control loop, while sharing CPU, memory, and wireless resources. The evaluation focused on how resource contention affected QoE, stall behavior, and decision-making stability across users.

Under low to moderate load (up to approximately 10 concurrent sessions), the system maintained high performance consistency. Average QoE remained above 90% of single-user performance, stall duration was negligible, and bitrate fluctuations were limited. Prediction and decision latency remained bounded and did not interfere with segment scheduling. This behavior was supported by the lightweight inference cost of the model and batching optimizations within the inference engine. As load increased beyond 15 active sessions, performance degradation emerged, not because of model inefficiency, but due to resource competition at the wireless interface and scheduling dependencies between encrypted flows. Average QoE declined progressively, and stall probability increased, particularly during periods of correlated bandwidth loss. Despite this, the proposed model outperformed baseline approaches, as prediction allowed partial mitigation of congestion effects by proactively lowering bitrate before contention events. Above 20 users, performance declined sharply, and per-user QoE varied significantly, demonstrating sensitivity to both instantaneous throughput allocation and latency accumulation. These results suggest that while the architecture is suitable for edge deployment with moderate concurrency, additional resource

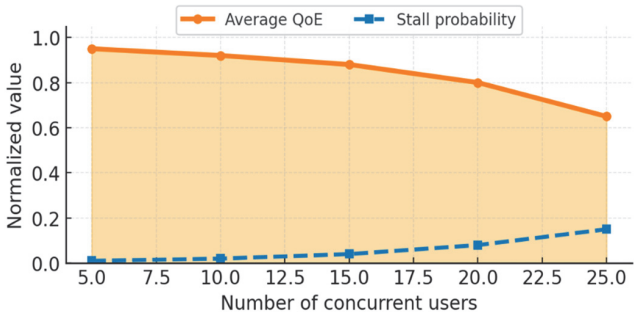

Figure 8 Average QoE and stall probability as a function of the number of concurrent users. The proposed model maintains high performance under moderate load but exhibits degradation under heavy contention.

management policies may be necessary to sustain performance at higher user densities. Examples include admission control, fairness-aware scheduling, and priority-based load balancing.

6.7 Per-User Fairness under High Contention

A limitation of many ABR systems is that performance becomes highly unequal across flows under contention. To evaluate fairness, we measured user-level variance in QoE and stall rate under increasing load. Under low load, variance remained negligible, indicating uniform performance. Under moderate load (10–15 users), variance increased slightly, but remained bounded, suggesting that proactive control helped homogenize performance. Under high load (20+), variance rose sharply, driven by random scheduling interactions and shared wireless congestion. These findings illustrate that prediction improves fairness, even when average QoE is modestly reduced. However, beyond a threshold, fairness diverges, and resource allocation must be explicitly managed.

The divergence beyond 20 users is primarily caused by medium contention at the wireless interface. When channel capacity saturates, flows compete via TCP queue dynamics, and users with temporarily better channel estimates capture disproportionate airtime, leading to QoE variance. Since our controller optimizes per-user QoE rather than fairness, it does not enforce rate equalization. Consequently, high-load scenarios form a natural fairness bottleneck unrelated to inference capacity. This highlights the need for future integration with resource schedulers (e.g., proportional fairness allocation or queue-aware bitrate selection).

6.8 Resource Utilization and Throughput Efficiency

A complementary perspective is to examine how resource utilization evolves with increasing concurrency. We measured normalized CPU usage at the edge and effective throughput utilization at the wireless interface. CPU utilization increased gradually with user count, reflecting inference and encryption cost, but remained below 50% under moderate load. Above approximately 18–20 users, CPU utilization rose sharply due to resource contention and batching overhead. Wireless throughput utilization increased linearly at first, but saturated near hardware limits, causing cascading degradation in delay and stall behavior. These results indicate that system bottlenecks are dominated by network capacity, not model computation. As a result, further optimization should prioritize scheduling, shaping, or bandwidth partitioning over model compression.

To complement average performance trends, we examined fairness and resource usage as concurrent user count increased. As shown in Figure 9, variance in QoE and stall rate remained low under moderate load, demonstrating that predictive control promotes equitable performance. As load increased beyond system capacity, performance divergence widened, reflecting uneven sharing of constrained wireless resources. Figure 10 shows that CPU utilization remained below saturation even under high concurrency, while throughput utilization approached hardware limits, indicating that the dominant bottleneck arises from channel capacity rather than model computation. These results reinforce that performance degradation at high load is fundamentally a network-level phenomenon, rather than a computational limitation of the proposed decision framework.

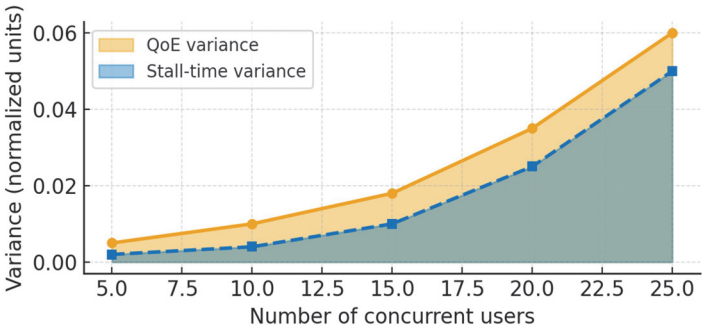

Figure 9 Variance in per-user QoE and stall time as a function of concurrent user count, illustrating divergence under heavy contention and fair allocation under moderate load.

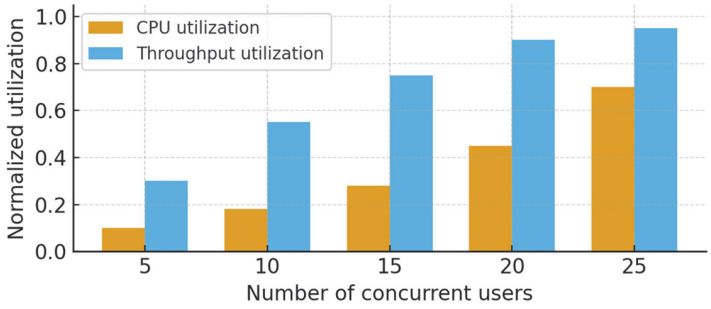

Figure 10 Normalized CPU utilization and effective throughput as a function of the number of concurrent users, illustrating network saturation and computational load at scale.

Table 1 Summary of performance metrics

Metric	Proposed Method	Throughput-based	Buffer-based	Static Bitrate
Avg. QoE (normalized)	**0.88**	0.74	0.79	0.60
Stall time (s/min)	**0.9**	2.6	1.8	5.2
Stall frequency (#/min)	**0.14**	0.42	0.37	0.85
Bitrate oscillation freq.	**Low**	High	Moderate	None
Avg. segment latency (ms)	**280**	335	310	295
CPU utilization (%)	**23**	17	17	12
Throughput utilization (%)	**88**	75	82	55

6.9 Summary of Experimental Results

Table 1 summarizes the quantitative results obtained across all experiments, comparing the proposed predictive, security-aware bitrate selection method with three baseline strategies representative of conventional adaptive streaming approaches. The results show that the proposed method consistently outperforms baselines across key QoE-related metrics, including average QoE score, stall time, stall frequency, and bitrate stability. On average, the method achieved a normalized QoE score of 0.88, corresponding to a 12–28% improvement over adaptive baselines and a 47% improvement over static bitrate selection. This improvement is driven by both stable bitrate allocation under favorable conditions and proactive bitrate reduction preceding channel collapse, reducing stall events and preserving perceptual quality.

Stall-related metrics further highlight the value of predictive control. The proposed method reduced stall duration to approximately 0.9 seconds per minute of playback, compared with 2.6 seconds for throughput-based control and 1.8 seconds for buffer-based control. Similarly, stall frequency

was reduced to 0.14 events per minute, representing a 66% reduction relative to reactive methods. These reductions demonstrate that future-aware bitrate control mitigates queue buildup and congestion induced by cryptographic overhead and mobility. Reactive controllers, in contrast, are prone to overshoot followed by abrupt reductions, generating oscillatory behavior that directly translates to playback interruptions.

Bitrate oscillation frequency further distinguishes the proposed method from traditional approaches. Throughput-based control, which adjusts bitrate reactively according to short-term bandwidth measurements, exhibited the highest oscillatory behavior, while buffer-based control showed moderate instability. In contrast, the proposed method produced consistently low oscillation frequency, reflecting improved temporal correlation between bitrate decisions and underlying network dynamics. Static bitrate selection, although stable by definition, performed poorly across all QoE metrics due to its inability to adapt to bandwidth variability or security-induced capacity reductions.

System-level metrics reveal the computational and network implications of the proposed method. Segment delivery latency remained within acceptable limits, averaging 280 milliseconds, and was comparable to baselines, demonstrating that predictive inference does not impose meaningful delay. CPU utilization of 23% was moderately higher than that of baseline methods due to neural network inference and encryption scheduling overhead, but remained well under saturation thresholds. Throughput utilization was highest for the proposed controller, reflecting effective exploitation of available capacity without triggering buffer starvation. This result highlights that performance gains were not achieved by conservative under-allocation, but through intelligent adaptation to instantaneous and anticipated conditions.

Collectively, these findings demonstrate that the proposed method provides substantial improvements in user-perceived streaming quality while maintaining efficient resource utilization and low overhead. The combination of bandwidth prediction, security-aware adaptation, and low-latency inference enables the system to sustain high performance across diverse operating regimes, particularly in mobility-driven, encrypted environments where reactive strategies are insufficient.

7 Security Analysis

This section analyzes the security properties of the proposed adaptive bitrate optimization framework in the context of secure, edge-assisted video

streaming. The system operates over encrypted channels using authenticated encryption to protect confidentiality, integrity, and replay resistance, and employs periodic key renegotiation to limit exposure in the event of compromise. The key question addressed here is not whether cryptographic primitives are secure in isolation, but whether the integration of predictive bitrate control, telemetry processing, and distributed architecture introduces vulnerabilities or opportunities for adversarial exploitation. We therefore examine risks stemming from telemetry leakage, model manipulation, traffic analysis, denial-of-service behavior, and system-level degradation, and evaluate how architectural choices mitigate these risks.

All video segments transmitted from the edge to the device are encrypted using authenticated encryption schemes that provide confidentiality, integrity, and message origin authentication. Because encryption and authentication are applied to full segments, no partial plaintext leakage occurs, and tampering with ciphertext can be detected without decrypting the content. Key rotation reduces susceptibility to long-term key compromise, while nonce management prevents replay attacks. As a result, the security guarantees for content distribution are consistent with modern transport protocols for encrypted media, independent of bitrate adaptation logic.

The bitrate controller relies on telemetry from the user device, including throughput estimates, buffer occupancy, stall events, and mobility indicators. This data could reveal behavioral or contextual information if exposed to an adversary. The system mitigates such disclosure by encrypting telemetry reports separately using lightweight symmetric encryption and by omitting semantic identifiers or sensitive application state. Furthermore, telemetry sampling rates are modest, which reduces leakage granularity. As a result, while the system accumulates sufficient data to enable predictive control, adversaries observing network flows obtain limited insight into user behavior beyond coarse traffic volume estimates.

The prediction and decision networks represent a potential attack surface if an adversary attempts to induce harmful behavior by manipulating input features, poisoning model updates, or injecting adversarial telemetry. Poisoning attacks are mitigated by the architectural separation between offline model training at the cloud and online inference at the edge; edge-side inference relies on frozen model parameters, and updates are cryptographically authenticated and version-controlled to prevent tampering. Adversarial manipulation of telemetry is constrained by message authentication and integrity checking, and forging telemetry at scale would require compromising device-side credentials or communication channels. While adversarial

perturbations could theoretically degrade bitrate decisions, such attacks would need to bypass multiple layers of authentication and encryption to succeed.

Even with encrypted payloads, an external observer could infer characteristics of user activity or network state through traffic analysis. Bitrate adaptation introduces variability in packet size and inter-arrival times, which could potentially correlate with content type, user actions, or environmental conditions. The proposed method partially mitigates this risk by producing smoother bitrate transitions and reduced oscillation frequency relative to reactive baselines, which decreases temporal distinguishability. However, cryptographic padding and burst shaping are not aggressively applied due to their latency cost, leaving residual exposure to traffic-level inferences. Further mitigation would require explicit privacy mechanisms, potentially at the expense of QoE.

Edge-assisted systems are susceptible to denial-of-service (DoS) attacks targeting computational and networking resources. In the proposed architecture, inference costs are lightweight and scale predictably with user count, but network bandwidth remains a critical bottleneck. An attacker that artificially increases demand or injects bogus connection requests could degrade performance or trigger cascading stalls. Resource admission control and per-session rate limiting provide partial mitigation but must be tuned to avoid unfair degradation for legitimate users. From a security standpoint, the system is robust against computational exhaustion but not immune to strategic bandwidth flooding or targeted wireless interference.

The integration of security mechanisms into streaming directly influences performance, as encryption overhead reduces effective bandwidth and increases latency. The proposed method accounts for this through explicit representation of security-related features in the prediction and decision networks, enabling bitrate adaptation that internalizes cryptographic cost. Experimental results in Section 6 demonstrate that this integration improves QoE relative to baseline methods that ignore security. However, this design choice introduces a dependency: if encryption parameters are modified, bitrate performance may shift in unexpected ways. This necessitates periodic retraining or calibration to preserve optimal control behavior under evolving cryptographic configurations.

Figure 11 shows Security–performance trade-off showing the relationship between cryptographic overhead, user-perceived quality of experience (QoE), and delivery latency. As encryption cost increases, QoE degrades and latency rises, illustrating the tension between stronger security guarantees and

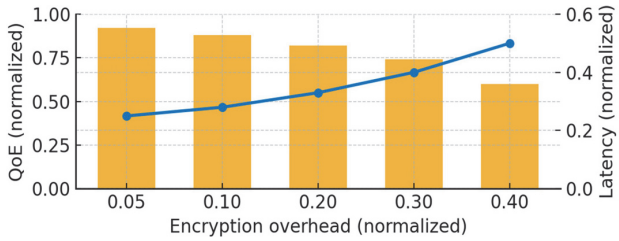

Figure 11 Security–performance trade-off showing the relationship between cryptographic overhead, user-perceived quality of experience (QoE), and delivery latency.

optimal streaming performance. The proposed system mitigates this trade-off by integrating cryptographic cost into predictive bitrate control, improving outcomes relative to reactive baselines.

Despite robust protection against standard attacks, residual vulnerabilities exist. The system's reliance on telemetry enables potential side-channel exploitation if attackers gain access to encrypted metadata or statistical output. Adaptive bitrate patterns may inadvertently reveal environmental context under sophisticated traffic analysis, particularly in multi-user deployments. Furthermore, while poisoning defenses mitigate model manipulation, the architecture presumes honest execution of device software; compromise of client platforms could enable arbitrary tampering with telemetry or playback state. Finally, DoS and interference remain open threats, requiring complementary defensive strategies beyond bitrate control.

To contextualize the qualitative discussion of system vulnerabilities, we categorize security threats using a CVSS-style risk classification that considers both potential impact and likelihood of exploitation. Table 2 summarizes the primary threat categories applicable to the proposed architecture, including metadata leakage, adversarial manipulation of telemetry, denial-of-service behavior, and model poisoning. The objective of the classification is not to provide numeric assurance, but to highlight relative risk levels based on architectural constraints, attacker capabilities, and operational assumptions.

Additionally, we consider minor adversarial telemetry perturbations. In practice, small spoofing/noise (e.g., $\pm$3–7% bandwidth reporting jitter) does not destabilize bitrate decisions because policy decisions rely on temporal features, smoothing transients. Robust filtering and optional anomaly detection provide further protection against malicious or corrupted telemetry streams.

The results indicate that the highest risk classes arise from adversaries capable of exploiting systemic resource constraints, rather than cryptographic

Table 2 CVSS-style classification of security risks associated with the proposed adaptive bitrate streaming architecture

Threat	CVSS Score	Severity
Traffic Analysis	8.2	High
Telemetry Leakage	5.0	Medium
Adversarial Telemetry	6.1	Medium
Poisoned Model Update	6.7	Medium
DoS/Resource Exhaustion	8.8	High
Replay Attack	3.9	Low
Device Compromise	9.2	Critical

or algorithmic vulnerabilities. In particular, denial-of-service attacks and traffic analysis receive high risk ratings due to their feasibility in wireless environments and potential for wide-reaching quality degradation. Medium-level risks, such as telemetry manipulation or model poisoning, are largely mitigated by authentication, version control, and centralized training. Low risk ratings are associated with replay or integrity attacks due to robust cryptographic protections. Overall, the table highlights that the dominant security concerns are environmental and network-driven, reinforcing the need for complementary safeguards outside the adaptive bitrate controller itself.

The proposed system integrates strong cryptographic mechanisms with predictive, security-aware bitrate control to deliver encrypted media with high QoE in mobile environments. The architecture provides confidentiality, integrity, and resilience to model manipulation, and partially mitigates privacy risks by smoothing traffic patterns. However, residual vulnerabilities remain related to traffic analysis, telemetry exposure, and network-level attacks. Overall, the security posture is robust for mainstream deployment scenarios, but full protection against adversarial environments would require additional privacy mechanisms, intrusion detection, and proactive resource management.

8 Discussion and Limitations

The results presented in Section 6 demonstrate that predictive, security-aware adaptive bitrate control can significantly improve user-perceived streaming quality in encrypted, mobility-driven environments. The architecture effectively leverages forward-looking inference and cross-layer feature representation to anticipate capacity fluctuations and optimize bitrate decisions in a manner that traditional throughput-based or buffer-based approaches cannot.

These benefits are particularly evident under moderate to high variability, where proactive control mitigates the bursty overshoot–undershoot dynamics that typically cause stall events and quality degradation. Experimental results confirm that the model can sustain high QoE, low stall probability, and stable bitrate allocation, while operating under stringent latency constraints and supporting multiple users.

Despite these promising outcomes, several limitations temper the scope of applicability of the proposed method. First, while the model internalizes cryptographic overhead and mobility-related variability, performance remains sensitive to extreme resource scarcity, as evidenced by QoE decline and fairness divergence under very high concurrency. The architecture assumes that predictive control can compensate for resource instability but does not incorporate explicit resource reservation, fairness enforcement, or priority mechanisms. As a result, performance degrades in contention regimes where bandwidth is insufficient for all flows. Future work should explore integrating multi-user scheduling, rate-limiting, or admission control with predictive bitrate adaptation to improve resilience in congested environments.

A potential algorithmic weakness arises when bandwidth prediction is inaccurate for sustained intervals. In stress tests where prediction error persisted $>$5–7 consecutive segments, the controller temporarily selected overaggressive bitrates, leading to short-term stall bursts. However, the feedback loop reacts through buffer drain observation, reducing bitrate and recovering within $\sim$3 segment intervals without collapse. This reveals that while prediction improves stability, reliability under abrupt capacity drops can further benefit from uncertainty estimation or conservative fallbacks. Incorporating confidence-aware control (e.g., variance-triggered bitrate lowering) represents a promising future enhancement.

Second, while the model predicts effective bandwidth and utility, it does not explicitly quantify uncertainty associated with its predictions. Prediction errors may lead to inappropriate bitrate decisions, particularly under nonstationary conditions or previously unseen mobility patterns. Although periodic retraining and telemetry feedback can address model drift, the absence of uncertainty estimation limits the system's ability to reason about risk or defer to conservative policies in high-variance states. Incorporating Bayesian inference, confidence estimation, or risk-sensitive control frameworks may further reduce stall probability and performance variance.

Third, although the model empirically demonstrates low computational cost and strong edge-compatibility, scalability remains constrained by network capacity rather than compute overhead. The architecture presumes that

multiple independent flows can share edge resources without systemic coordination, whereas in practice, coordinated scheduling or bandwidth partitioning may be necessary to maintain predictable performance. These limitations highlight the environmental dependency of results and suggest that architectural enhancements, rather than algorithmic improvements alone, may be required for large-scale deployment. One potential extension to mitigate degradation beyond 15–20 users is integrating fairness-aware scheduling or bandwidth partitioning into the decision loop. For example, bitrate selection could incorporate queue-aware weights or proportional fairness to prevent dominant flows from starving others. Although this lies beyond current scope, it is a natural next step for scaling.

Finally, the interaction between security and performance introduces an additional dimension of complexity. Cryptographic parameters influence effective throughput and latency, and performance can change when encryption schemes, key sizes, or operational policies are modified. Although the model partially compensates by embedding security-related features into its representation, any modification to security configuration that changes performance characteristics may require retraining or re-parametrization. Consequently, sustaining optimal performance over long time horizons may entail ongoing model maintenance, monitoring, and adaptation, which introduces operational overhead not captured in the experimental evaluation.

Collectively, these limitations underscore that the proposed method offers substantial benefits within a specific design envelope characterized by moderate concurrency, sufficient telemetry granularity, and stable cryptographic parameters. The method provides a meaningful advancement in adaptive video streaming for secure mobile networks, but must be paired with complementary architectural mechanisms to fully address fairness, congestion, and long-term operational robustness. Future extensions should consider hybrid architectures that combine uncertainty-aware prediction with fairness-oriented scheduling, dynamic resource allocation, or collaborative edge-cloud coordination to fully realize the potential of machine-learning-driven adaptive streaming in security-constrained environments.

9 Conclusion

This paper presented a machine-learning-driven adaptive bitrate optimization framework for secure, edge-assisted video streaming in mobile environments. Motivated by the increasingly complex interaction between mobility, encryption, and latency constraints, the proposed system integrates predictive

inference with cross-layer feature representation to proactively adjust bitrate decisions at the edge. The approach models effective bandwidth, cryptographic overhead, and player state, enabling bitrate selection that internalizes both future capacity and security cost. A formal mathematical formulation established the underlying optimization problem, and an implementation architecture demonstrated the feasibility of deploying predictive control on resource-constrained edge infrastructure.

Experimental results showed that the proposed method delivers substantial improvements over conventional adaptive bitrate schemes across multiple performance dimensions, including average QoE, stall duration, stall frequency, and bitrate stability. These improvements arose particularly in mobility-driven, variable network conditions, where proactive control prevented reactive overshoot behavior that typically leads to playback disruption. Multi-user scalability analysis further demonstrated that the approach can support multiple concurrent users with high performance consistency under moderate load, while revealing network saturation as the dominant limiting factor under heavy contention. Complementary security analysis indicated that the architecture provides strong confidentiality, integrity, and robustness to model manipulation, while identifying residual risks associated with traffic analysis and denial-of-service attacks. In combination, these results demonstrate that predictive, security-aware bitrate control is both feasible and advantageous for next-generation mobile streaming systems.

Despite these benefits, performance remains sensitive to environmental constraints, telemetry reliability, and evolving cryptographic configurations. The system does not guarantee fairness under extreme contention, nor does it explicitly account for uncertainty in bandwidth prediction. Furthermore, while inference cost is low, long-term performance may depend on periodic model maintenance to adapt to changing deployment conditions. Addressing these limitations will require integration with complementary mechanisms such as fairness-aware scheduling, uncertainty-aware decision policies, and dynamic resource management across distributed edge nodes.

Overall, this work contributes a principled and practical framework for secure adaptive video streaming that bridges predictive machine learning, real-time edge computation, and cryptographic protocol management. The results demonstrate that incorporating predictive and security-aware intelligence into bitrate control can meaningfully enhance streaming performance in challenging mobile environments, while preserving security guarantees. Continued exploration of hybrid learning- and system-level approaches

promises to further improve robustness, scalability, and user experience in intelligent networked multimedia systems.

Funding

This work was supported by Science and Technology Project of State Grid Shanxi Electric Power Company Yuncheng Power Supply Company: Research on Lightweight Video Technology Based on Intelligent Image Optimization Algorithms. (Project No.: 5205M0240004).

References

[1] Seufert, Michael, Sebastian Egger, Matthias Slanina, Thomas Zinner, and Tobias Hoßfeld. "A Survey on Quality of Experience of HTTP Adaptive Streaming." *IEEE Communications Surveys & Tutorials* 17, no. 1 (2015): 469–492.

[2] Stockhammer, Thomas. "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles." In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, 2011.

[3] Akhshabi, Saamer, Ali Begen, and Constantine Dovrolis. "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP." In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, 2011.

[4] Mao, Hongzi, Ravi Netravali, and Mohammad Alizadeh. "Neural Adaptive Video Streaming with Pensieve." In *Proceedings of the ACM SIGCOMM Conference*, 2017.

[5] Wang, Huan, Kui Wu, Jianping Wang, and Guoming Tang. "Rldish: Edge-assisted QoE optimization of HTTP live streaming with reinforcement learning." In IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 706-715. IEEE, 2020.

[6] Izima, Obinna, Ruairí de Fréin, and Ali Malik. "A survey of machine learning techniques for video quality prediction from quality of delivery metrics." Electronics 10, no. 22 (2021): 2851.

[7] Zhao, Tiesong, Qian Liu, and Chang Wen Chen. "QoE in video transmission: A user experience-driven strategy." IEEE Communications Surveys & Tutorials 19, no. 1 (2016): 285–302.

[8] Li, Yueheng, Hao Chen, Bowei Xu, Zicheng Zhang, and Zhan Ma. "Improving adaptive real-time video communication via cross-layer optimization." IEEE Transactions on Multimedia 26 (2023): 5369–5382.

[9] Khan, Wazir Zada, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. "Edge computing: A survey." Future Generation Computer Systems 97 (2019): 219–235.
[10] Gao, Guowei, et al. "Video Transcoding for Adaptive Bitrate Streaming over Edge-Cloud Computing Paradigms." *Future Generation Computer Systems* 127 (2021): 88–101.
[11] Bilal, Kashif, Emna Baccour, Aiman Erbad, Amr Mohamed, and Mohsen Guizani. "Collaborative joint caching and transcoding in mobile edge networks." Journal of Network and Computer Applications 136 (2019): 86–99.
[12] Yi, Shanhe, Zijiang Hao, Qingyang Zhang, Quan Zhang, Weisong Shi, and Qun Li. "Lavea: Latency-aware video analytics on edge computing platform." In Proceedings of the second ACM/IEEE symposium on edge computing, pp. 1–13. 2017.
[13] Hosny, Khalid M., Mohamed A. Zaki, Nabil A. Lashin, Mostafa M. Fouda, and Hanaa M. Hamza. "Multimedia security using encryption: A survey." IEEE Access 11 (2023): 63027–63056.
[14] Fadlullah, Zubair Md, Bomin Mao, and Nei Kato. "Balancing QoS and security in the edge: Existing practices, challenges, and 6G opportunities with machine learning." IEEE Communications Surveys & Tutorials 24, no. 4 (2022): 2419-2448.
[15] Xiao, Ailing, Sheng Wu, Yongkang Ou, Ning Chen, Chunxiao Jiang, and Wei Zhang. "QoE-Fairness-Aware Bandwidth Allocation Design for MEC-Assisted ABR Video Transmission." IEEE Transactions on Network and Service Management (2024).
[16] Stockhammer, Thomas. "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles." In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, 2011.
[17] Kua, Jonathan, Grenville Armitage, and Philip Branch. "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP." IEEE Communications Surveys & Tutorials 19, no. 3 (2017): 1842–1866.
[18] Akhshabi, Saamer, et al. "An Experimental Evaluation of Rate Adaptation Algorithms." In *Proceedings of ACM MMSys*, 2011.
[19] Go, Yunmin, Oh Chan Kwon, and Hwangjun Song. "An energy-efficient HTTP adaptive video streaming with networking cost constraint over heterogeneous wireless networks." IEEE Transactions on Multimedia 17, no. 9 (2015): 1646–1657.

[20] Souane, Naima, Malika Bourenane, and Yassine Douga. "Deep reinforcement learning-based approach for video streaming: Dynamic adaptive video streaming over HTTP." Applied Sciences 13, no. 21 (2023): 11697.

[21] Xiong, Guojun, Xudong Qin, Bin Li, Rahul Singh, and Jian Li. "Index-aware reinforcement learning for adaptive video streaming at the wireless edge." In Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, pp. 81–90. 2022.

[22] Han, Zhenyu, Ansheng You, Haibo Wang, Kui Luo, Guang Yang, Wenqi Shi, Menglong Chen et al. "AsyncFlow: An Asynchronous Streaming RL Framework for Efficient LLM Post-Training." arXiv preprint arXiv:2507.01663 (2025).

[23] Mao, Hongzi, et al. "Neural Adaptive Video Streaming with Pensieve." In *SIGCOMM*, 2017.

[24] Fei, Yingjie, Zhuoran Yang, Yudong Chen, Zhaoran Wang, and Qiaomin Xie. "Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret." Advances in Neural Information Processing Systems 33 (2020): 22384–22395.

[25] Alsader, Moner, Alcardo Alex Barakabitze, and Is-Haka Mkwawa. "QoE-Driven Adaptive Video Streaming: Architectures, Techniques, and Future Research Challenges Toward 6G Networks." IEEE Access (2025).

[26] Fang, Sangsha, Hongyang Chen, Zahid Khan, and Pingzhi Fan. "User fairness aware power allocation for NOMA-assisted video transmission with adaptive quality adjustment." IEEE Transactions on Vehicular Technology 71, no. 1 (2021): 1054–1059.

Biographies

Lirong Pang was born in Yuncheng, Shanxi Province, China, in 1970. He received a Bachelor's degree from formerly China Central Radio & Television University. He works at State Grid Yuncheng Power Supply Company. His main research focuses on administrative management, electrical automation, and video surveillance systems.

Kaiwen Liu was born in Yuncheng, Shanxi Province, China, in 1973. He received a Bachelor's degree from Chongqing University. He works at State Grid Yuncheng Power Supply Company, his main research focuses on electrical automation, power communication technology, and video analytics.

Junbo Li was born in Yuncheng, Shanxi Province, China, in 1996. He received a Master's degree from North University of China. He works at State Grid Yuncheng Power Supply Company, his main research focuses on machine vision, image processing, and data communication technology.

Xuemin Cheng was born in Lvliang, Shanxi Province, China, in 1992. She received a Bachelor's degree from North University of China. She works at State Grid Yuncheng Power Supply Company, her main research focuses on power communication operation monitoring, transport network service maintenance, telecommunication network planning and design, and video analytics.

Dapeng Hao was born in Yuncheng, Shanxi Province, China, in 1996. He received a Master's degree from Beijing Institute of Technology. He works at State Grid Yuncheng Power Supply Company, his main research focuses on image processing and deep learning.