BitFormer: Transformer-Based Neural Network for Bitrate Prediction in Real-Time Communications

BitFormer: Transformer-based Neural Network for Bitrate Prediction in Real-Time Communications

Tailai Song, Gianluca Perna, Paolo Garza, Michela Meo, Maurizio Matteo Munafò
Politecnico di Torino, Turin, Italy
first.last@polito.it

***Abstract*—In recent years, an exponential upsurge in the global proliferation of Real-Time Communications (RTC) applications has been witnessed, due to the prosperous development of networks and further fueled by the ramifications of the COVID-19 pandemic. Consequently, the imperative for development of intelligent, resilient, and scalable network infrastructures and technologies has grown significantly. Real-time bitrate prediction could play a crucial role, offering network observability and bolstering proactive system management. By accurately forecasting bitrate, it becomes possible to implement improvements at either application level or network level, such as swift and appropriate bandwidth adaptation. In this paper, we propose a novel Transformer-based deep learning framework called BitFormer designed to predict the short-term bitrate. Our work is based on extensive traffic data collected under various conditions using two prevalent RTC applications, and our model relies solely on packet-level information, which contains the fundamental traffic characteristics and facilitates effortless feature extraction. Through comprehensive evaluations and comparisons, we achieve a superior accuracy of 74% in identifying peak bitrates, while simultaneously ensuring commendable overall performance.**

***Index Terms*—Networking, packet level, Real-time communications, RTP, machine learning, deep learning.**

I. Introduction

Real-time communications (RTC) have emerged as a pivotal element in contemporary society, supporting applications such as video-conferencing and streaming to play a vital role in both professional and recreational domains. The unprecedented popularity of RTC applications in recent years can be attributed to the surging demand for entertainment and improved lifestyles in the post-pandemic era, coupled with the widespread adoption of remote work [1]. As RTC services continue to rapidly evolve, the market has become saturated with a plethora of competing applications [2], benefitting from the global expansion of network infrastructures, increased bandwidth availability, and advancements in 5G technologies. Most applications employ Real-Time Transport Protocol (RTP) [3] over User Datagram Protocol (UDP), while web browsers rely on the widely adopted standard, WebRTC[1], an open-source framework atop RTP.

In this context, there is a burgeoning interest in developing advanced and intelligent network technologies to enhance network performance and Quality of Experience (QoE). Notably, bandwidth management assumes a significant aspect in RTC, encompassing crucial functionalities such as throughput measurement, bandwidth allocation, dynamic transmission adjustments, and traffic prioritization [4], [5], [6], [7]. In light of this, bitrate prediction for traffic flows holds immense potential, proffering a proactive system that yields manifold benefits: i) Optimized bandwidth allocation and utilization can be achieved through precise bitrate estimation, thereby avoiding both over-provisioning and underutilization of network capacity; ii) Adaptive streaming and transcoding can enhance the QoE by dynamically adjusting media quality, resolution, or encoding settings based on the predicted bitrate, ensuring optimal content delivery; iii) Network congestion management can be effectively performed by predicting bitrate requirements, enabling preemptive actions such as traffic shaping, prioritization, or rerouting to mitigate or prevent congestion issues; iv) Resource planning becomes more efficient as service providers and network operators leverage predicted bitrate information to assess and allocate the necessary network resources. However, bitrate prediction is an arduous task, particularly in the context of RTC due to dynamic and ever-changing network traffic, limited computational resources, and time constraints.

In this paper, we present BitFormer, a novel Transformer-based Deep Learning (DL) Neural Network (NN), that exclusively leverages packet-level information for bitrate prediction within a future time window of 500 ms, endeavoring to overcome inherent challenges in the problem. The utilization of packet-level information offers the advantage of minimal extraction efforts, making it suitable for lightweight network devices like edge routers. Additionally, the sequential nature of packet flows bears resemblance to Natural Language Processing (NLP) problems, which have been revolutionized by the game changer - Transformer [8], enabling the proposed model to exhibit promising capabilities in capturing the dynamic and intrinsic patterns of networks. Furthermore, our streamlined and simplified architecture provides computational efficiency and reduced time consumption.

Our work is grounded in abundant real videoconferencing traffic collected on client sides with diverse network connections. We formulate a regression problem and compare the performance with respect to a simple baseline and multiple popular techniques, from an adaptive filter to traditional Machine Learning (ML) and DL approaches. Specifically, our focus lies on the prediction performance of peak values, which are critical in RTC as they represent the bottlenecks of packet flows. Consequently, BitFormer demonstrates enhanced performance, exhibiting satisfactory results overall and particularly excelling in predicting peak values. Additionally, our proposed solution is envisioned to function as a software module for end-users or network devices such as media servers, establishing an AI-based, RTC-aware, comprehensive, and proactive traffic mon-

This work was supported by Cisco Systems Inc. and the SmartData@PoliTO center on Big Data and Data Science.

[1]https://webrtc.org/

itoring and management system. It enables application-level observability at the network control plane, empowering efficient and informed decision-making, and incorporates a feedback mechanism to promptly notify time-varying conditions.

II. Related work

In this section, we provide an overview of relevant literature pertaining to bitrate and packet-level prediction.

Bitrate prediction, also known as bandwidth or throughput prediction, has garnered attention in academic research. A Recursive Least Squares (RLS) [9] filter was introduced in [10] to predict bandwidth for video calls in cellular networks, and a Random Forest (RF) [11] framework was developed in [12] to predict cellular link bandwidth in 4G Long Term Evolution (LTE) networks. The authors in [13] leveraged public datasets of general Internet traffic and adopted multiple ML algorithms, extracting features from aggregated packets to perform short-term bandwidth prediction. Additionally, Long Short-Term Memory (LSTM) [14] model has been explored in [15], where real-time mobile bandwidth prediction was investigated using a LSTM model enhanced by Bayes model fusion. Meanwhile, in [16], [17], [18], the focus was on Adaptive Bitrate (ABR) for HTTP-based video streaming. Tree-based models and DL frameworks were proposed for throughput prediction and integrated into ABR algorithm to optimize QoE.

Unlike time-series prediction, which leverages historical temporal data, packet-level prediction hinges on fine-grained features derived from packets that typically exhibit irregular granularity and implicit correlation with the targets. Authors in [19] employed multitask DL approach to utilize packet-level information for predicting packet-level characteristics. They investigated multiple DL techniques and compared the performance against Markov chain and RF regressor. Packets with 3 predicted and 3 exogenous parameters were arranged in a sequential way to perform sliding window prediction. Furthermore, Transformer was explored in both [20] and [21]. The former work classified real-time network flow types (video, conference, and download), by proposing FlowFormer, an ensemble architecture of LSTM and CNN with attention-based encoders. Particularly, packet information like payload length was collected and compared with predefined thresholds to be aggregated into categorized bins, and packet quantities in such bins were calculated as features. The latter study attempted to model and generalize network dynamics through Transformer, based on packet-level information (e.g., timestamps). The authors implemented the general architecture except that an additional hierarchical aggregation layer preceding the encoder was added to condense lengthy sequence and concatenate older and recent packets. They undertook an end-to-end delay prediction to pre-train the model and envisioned a replaceable decoder for other tasks.

To the best of our knowledge, our work represents a pioneering effort in employing Transformer-based architecture with packet-level information to predict bitrate in RTC. Notably, our study adopts a per-flow approach, concentrating on the analysis of RTP packets collected from diverse connections. Our model has a streamlined architecture, efficiently leveraging a minimal set of packet-level information as features. Consequently, the need for resource-intensive processes such as intricate feature extraction, extensive aggregation, and complex calculations is eliminated, contributing to the lightweight nature of our proposed framework.

III. Problem statement

In this section, we start by presenting the motive behind our approach. Subsequently, we formulate the problem and introduce the dataset used to accomplish the objective.

A. Underlying motive & Feature selection

The rationale of selecting packet-level information for bitrate prediction is threefold: i) Packets embody the finest granularity and serve as the most fundamental element within networks, encapsulating the rapidly-changing dynamics and essential nature of network traffic [22]. ML models trained on such fine-grained features possess a greater likelihood of capturing the underlying patterns, thereby facilitating a more precise prediction; ii) The acquisition of packet-level information requires minimal effort in terms of feature extraction, especially in the context of RTC, given the possible constraints imposed by limited time and computational resources. On top of that, our approach relies exclusively on packet header elements, circumventing potential complications due to packet encryption and enabling a more streamlined workflow with prompt access to relevant information; iii) Packets are readily accessible across the network, affording the possibility of holistic network observability rather than relying solely on the client-side. This enables the prospect of performing bitrate prediction within the network, contributing to the improvement of overall network performance.

In this context, we select 5 elements of the RTP packet as features for bitrate prediction. They are: i) **Frame length**, the packet total length including all its headers and data; ii) **RTP timestamp**, the timestamp field present in the RTP header; iii) **Inter-arrival time**, the time elapsed between the arrivals of two consecutive packets; iv) **Sequence number**, a 16-bit value that is used to identify and order the RTP packets; v) **RTP marker**, a single-bit field used to indicate the last packet of a specific media unit. Frame length serves as a spatial component, which directly indicates the impact of packet size and transmitted bits in the past, endowing the model with the capability to operate in an autoregressive manner. RTP timestamp and inter-arrival time represent temporal components, providing insights into the timing patterns and enabling the model to discern temporal dependencies and dynamics that might influence the bitrate prediction. Sequence number and RTP marker contribute to RTP event components, reflecting the potential impact of specific network events, e.g., packet loss (based on the inconsistency between sequence numbers).

B. Problem formulation

The primary objective is to predict the bitrate within the next 500 ms using packet-level information. Assuming at a time instant t, we formulate a regression problem as follows:

$$\begin{aligned} \hat{R}_t &= f(\overline{x}_{t,1}, \overline{x}_{t,2}, ..., \overline{x}_{t,n}, ..., \overline{x}_{t,1023}, \overline{x}_{t,1024}) \\ &\text{with } n \in [1, 1024], \\ \overline{x} &= (x_{\text{frame length}}, x_{\text{RTP timestamp}}, x_{\text{inter-arrival time}}, \\ &\quad x_{\text{sequence number}}, x_{\text{RTP marker}}), \end{aligned} \tag{1}$$

where t is the current moment and $\hat{R}_t$ is the bitrate in the subsequent 500 ms time window starting from time t and ending at $t + 500\ ms$. $\overline{x}_{t,n}$ represents the feature vector of n^{th} previous packet before time t, and it encapsulates the corresponding packet information constituted by a tuple of the 5 elements. We aim at developing an ML model to learn a function $f(\cdot)$, which performs the regression task and maps our input vectors $\overline{x}$ of previous 1024 packets to the bitrate $\hat{R}$.

C. Dataset

In our work, we collect packet traces from 72 real video teleconferences, using two RTC applications, *Webex* and *Jitsi Meet*, and store them in *pcap* format. All calls comprise 2 to 6 participants, engaging in a connection of WiFi, mobile, or Ethernet, and the cumulative duration is approximately 70 hours. The traffic is collected on client sides and only incoming streams are considered. We adhere to a common definition of an RTP flow - a tuple composed of $(ip_{src},\ ip_{dst},\ port_{src},\ port_{dst},\ ssrc,\ type_{payload})$, and split the traffic on a per-flow basis. For each RTP flow, we systematically extract the necessary information of packets following chronological order, and for each packet, we consider the 5 aforementioned elements. Based on the problem formulation, we start with the initial 1024 packets of a flow, calculating the bitrate within the subsequent 500 ms time window, by summing up the frame lengths of all packets in said window. Afterwards, we progress forward by 500 ms, assimilating new packets but discarding old ones to satisfy the stipulation of 1024 packets, and we iterate this procedure until consuming the entire flow. An illustration is available in the workflow part of Figure 1. Consequently, we obtain a sequence of bitrates in consecutive time windows, each accompanied by the historical information of its preceding 1024 packets. To provide context, 1024 packets denotes an average traffic duration of nearly 12.1 s (18.8 s for audio and 6.1 s for video). The selection of such a quantity stems from the pursuit of encompassing sufficient characteristics pertaining to various media types while endeavoring to strike a balance between an elongated duration including extraneous information and a shorter duration potentially omitting crucial details in proximity to the target. Nonetheless, leveraging the Transformer attention mechanism empowers us to autonomously discern the significance of individual packets, thereby harnessing their inherent values.

IV. Methodology

In this section, we introduce our proposed Transformer-based DL model as well as benchmarks. Then, we present the model development and evaluation process.

A. ML models

The architecture of BitFormer is illustrated in Figure 1. The model takes 1024 entries, each representing the critical information of a packet, as input. This information is fed into the initial linear layer, namely packet embedding layer, to expand the 5 elements of each packet to longer embedded features. After applying positional encoding, the embedded feature sequence is processed by a single layer of Transformer encoder to generate a sequence of encoded features, utilizing multi-head attention mechanism to learn latent patterns, adapt to traffic dynamics, and grasp network fate. Additionally, instead of implementing a Transformer decoder, we directly average the outputted features of each sequence sample, simplifying the architecture and aggregating the encoded features to distill feature essence, and then pass through a linear layer, producing a scalar value, i.e., the predicted bitrate. To compare the performance, we also consider a broad spectrum of domains, implementing several other approaches that appeared in related works as benchmarks, as outlined in Table I.

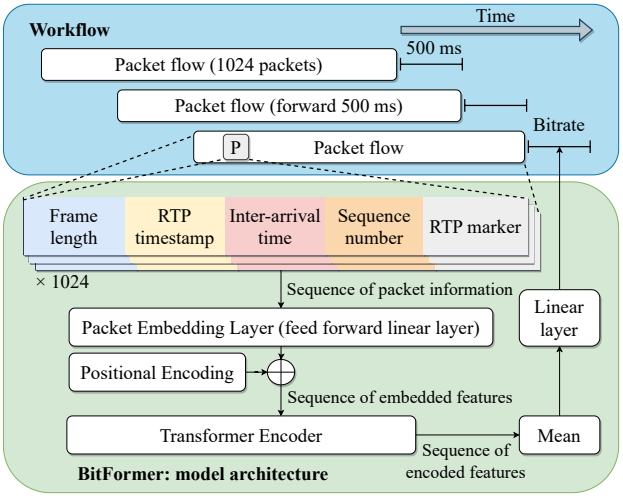


Fig. 1: BitFormer: workflow and model architecture.

TABLE I: Summary of other considered models.

Category	Model
Naive baseline*	Moving Average (MA) [23]
Adaptive filter**	Recursive Least Squares (RLS) [9]
ML method	Random Forest (RF) [11] regressor
DL method	Multi Layer Perceptron (MLP) [24] Long- and Short-term Time-series network (LSTNet) [25] Long Short-Term Memory (LSTM) [14]

* A simple MA approach, which calculates the average bitrate in the past 5 s (10 time windows) as the predicted bitrate for each target sample.

** Other more sophisticated adaptive filters like Kalman filter, or popular autoregressive algorithms like ARIMA are intentionally excluded due to their relatively demanding computational requirements for model updates, which may be prohibitive in the context of RTC with a granularity of 500 ms.

B. Model development & evaluation

In order to derive a generalized solution and avoid data infiltration among RTP flows, we intentionally partition the 72 *pcap* files into 50, 10, and 12, and for each group of files, we randomly extract flows to form training (1,000,000 samples of bitrate), validation (100,000) and test (300,000) datasets.

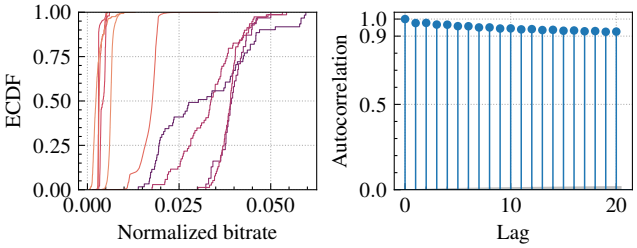
Furthermore, we deliberately introduce slight modifications to the problem formulation in order to comprehensively assess

the performance and analyze the influence of packet-level information. First, the problem can be also formulated as a conventional time series prediction one, in which historical time series samples are used to predict a future value. To achieve this, we construct the datasets in an alternative way, retaining the prediction targets but substituting previous packet features with preceding 20 bitrates (20 500-ms time windows) in 10 s, which roughly aligns with the aforementioned average traffic duration of 1024 packets. Second, it is intuitive to raise a concern regarding the redundancy of packet-level information, since the bitrate is calculated based on frame length, which leads to a theoretically strong correlation between the target bitrate and its previous frame lengths, and may render other features unnecessary. Therefore, in addition to the models training on all packet-level features, we also explore the scenario where only frame length is included. In the former case, we implement MA, RLS, RF, MLP, and LSTM, while in the latter case, LSTNet, LSTM and BitFormer are considered. Consequently, we develop 11 models in total and evaluate the results through Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) between predicted values and ground truths for the test dataset.

More importantly, despite the dynamic nature of bitrates over time, the overall variation could be minor. Figure 2 illustrates the Empirical Cumulative Distribution Function (ECDF) of bitrates in 10 randomly selected RTP flows (left figure) and the autocorrelation of the entire bitrate sequence (right figure). The ECDF demonstrates a steep ascent for most flows, indicating that the majority of bitrates within an individual flow tend to concentrate around a particular value[2]. The autocorrelation of the entire sequence exhibits a remarkably high value ($>$0.9), even when shifted by 20 time instants into the future. Both observations highlight the overall stability of bitrates with moderate variations, rendering the prediction less crucial for normal values with minimal fluctuations but more significant for ultra-high peak values. Our objective of bitrate prediction holds the implication of prioritizing peak values for a number of reasons. Understanding peak bitrates is crucial for capacity planning and service provisioning. Accurately predicting and accounting for peaks enables optimal allocation of network resources, preventing bottlenecks during high-demand periods. Limited and shared network resources are susceptible to bitrate peaks, which can negatively impact network activities, causing delays, increased latency, and reduced performance. Additionally, the peak bitrate directly affects media quality, with exceeding bandwidth or system capacity leading to packet loss, degraded audio/video, buffering, and diminished QoE.

To this end, besides the overall performance, we also specifically evaluate the models for critical bitrates. In particular, we extract the top 10% bitrates as peak values for each RTP flow. As a result, the problem can be framed as a binary classification task: if the predicted value surpasses 90% of the corresponding

[2]Only 10 flows are displayed in the ECDF plot for the sake of a clear visualization, and we have confirmed a similar pattern across the dataset.



(a) ECDF of bitrates in 10 random RTP flows.

(b) Autocorrelation of the entire bitrate sequence.

Fig. 2: Bitrate characterization.

peak value, it is deemed a solid prediction[3], otherwise the prediction is considered poor. We adopt both numerical metrics and classification accuracy to evaluate the performance.

V. Experimental Result

In this section, we present the experimental outcomes and discuss the performance for peak values. Table II presents all results regarding the entirety and peaks for each model.

We first focus on the overall results, comparing the metrics in time series prediction and packet-level prediction with all packet features. While RF outperforms the others, our proposed BitFormer exhibits a comparable performance, with the 2^{nd} best MAPE but moderately larger errors for MSE and MAE. However, all resulting MSEs and MAEs are relative acceptable with minor differences among models. This could be attributed to the dominance of stationary bitrates with slight fluctuations in the traffic (evident in Section IV-B). The predictability of such patterns allows all models to capture the basic trend effectively, yielding a considerable number of small errors that overshadow the differences in their general performance. This observation is further justified by the statistically respectable performance of the naive baseline, MA. In this context, BitFormer, which aims to prioritize peak values, may exhibit slightly more aggressive predictions, resulting in marginally large deviations for normal values.

Moving forward to peak values, BitFormer stands out with its remarkable performance, boasting the highest accuracy (64.3%), nearly 6% higher than the 2^{nd} best (58.2%). On top of that, the lowest MAE as well as MAPE and the decent MSE indicate a precise prediction rather than a simple overestimation, unlike LSTNet with 57.4% accuracy but higher errors, e.g., 15.0% MAPE. Interestingly, all three models with packet-level features produce advanced outcomes for peak values with respect to time series prediction, which demonstrates the superiority of packet-level information in identifying critical bitrates, and meanwhile, BitFormer further harnesses such merits to outperform others. Moreover, a typical example of an RTP flow is presented in Figure 3, which depicts the difference between ground truth and predicted bitrates for each model. In general, all models can follow the overall trend, adapting to abrupt changes. Although most of them perform

[3]The prediction does not have to be identical to the ground truth of a peak. It is reasonable to assign a margin of 10%, e.g., for a peak bitrate of 1 Mbps, a prediction of 0.9 Mbps ($1\ Mbps \times (1-10\%)$) could be considered satisfactory.

TABLE II: Model results: performance comparison of overall bitrate and peak values.

Problem	Time series prediction					Packet level prediction					
Feature	Previous bitrates					All packet features			Only frame length		
Model	MA	RLS	RF	MLP	LSTM	LSTNet	LSTM	BitFormer	LSTNet	LSTM	BitFormer
MSE	0.0094	0.0136	**0.0064**	0.0076	0.0094	0.0106	0.0104	0.0099	0.0796	0.0865	0.0457
MAE	0.0330	0.0338	**0.0285**	0.0318	0.0350	0.0422	0.0373	0.0377	0.1173	0.1306	0.0902
MAPE	12.8%	12.1%	**10.7%**	12.7%	15.1%	17.7%	12.4%	11.3%	38.2%	41.4%	28.3%
MSE_{peak}	0.0266	0.0291	**0.0206**	0.0256	0.0376	0.0278	0.0420	0.0279	0.1498	0.1755	0.1096
MAE_{peak}	0.0724	0.0686	0.0652	0.0735	0.0854	0.0663	0.0829	**0.0604**	0.1611	0.1919	0.1475
$MAPE_{peak}$	15.2%	14.5%	13.4%	14.4%	16.4%	15.0%	14.4%	**12.1%**	28.1%	31.2%	28.7%
Acc_{peak}*	49.1%	54.4%	55.7%	50.1%	45.6%	57.4%	58.2%	**64.3%**	56.2%	54.3%	47.2%

* $Accuracy\ of\ solid\ prediction\ for\ peak\ values = \frac{Number_{predicted\ value \geq peak\ value \times 90\%}}{Number_{peak}} \times 100\%$.

decently for normal values, they tend to underestimate the plateau. Fortunately, BitFormer excels in accurately predicting peak values without penalizing others. Additionally, the only comparable one is RLS, but with a closer glance, BitFormer still demonstrates superior performance. More importantly, the awfully inferior performance (the rightmost part of Table II) with magnitude-level degradation from models trained only with frame length for packet-level prediction explicitly proves and emphasizes the significance of packet-level information, further reinforcing the contribution of all the components that possess internal correlation to model traffic dynamics.

At this stage, it is important to acknowledge that the performance for other models seem abnormal for conventional time series prediction problems [26], [27] due to the following reasons: 1) We develop pre-trained models without model update in most cases, and the packet flows in different datasets are extracted from different *pcaps*, which creates obstacles for model adaptation; 2) The real-time nature of the problem, with a granularity of 500 ms, imposes constraints on the implementation of more sophisticated and computationally expensive state-of-the-art models, such as Autoformer [28]; 3) The prevalence of stable bitrates limits the model to learn patterns associated with critical values, which resembles the dilemma in imbalanced ML [29]; 4) In the context of RTC, it is reasonable to assume a weak long-term correlation, and short-term information with less available features might not be sufficient for accurate time series prediction. More importantly, the packet-level prediction does not strictly adhere to a time series but rather an ordinary sequence, which intrinsically lacks a constant interval, not to mention the absence of periodicity and other perceivable patterns. These aspects can also be viewed as intrinsic difficulties in our problem. However, our proposed model, incorporating multi-head attention mechanism, possesses the capability to capture intricate patterns within different components of packets and uncover their correlation with future bitrate dynamics by exploiting the entire sequence in one single shot, enabling a global perspective for the final prediction. This empowers BitFormer to effectively address these issues and adapt to abrupt changes for peaks, outperforming the other models.

Additionally, we also investigate the performance of predicting bitrates within different future time horizons, 300 and 1000 ms. Herein, we only implement RF with the best overall performance in the previous scenario to compare BitFormer. For RF in time series prediction, we maintain the duration (10 s) to be considered in the past, including 33 historical bitrates as features for the 300 ms case and 10 for the 1000 ms case. As for BitFormer, we still consider 1024 packets but adjust the time shift to align with the predicted time window of either 300 or 1000 ms. Table III showcases all the results. It is noteworthy that BitFormer still demonstrates proficiency in handling peak values without sacrificing overall performance, which coincides with the 500 ms scenario, further consolidating and justifying the consistent and versatile advantage of our proposed model. Comparing performance across different time horizons (Table II and III), 1000 ms results in the best behaviour for both models, with lowest errors in most cases and a staggering accuracy of 73.8%, not to mention the longer predicted time window, which affords a higher degree of freedom for system management to implement optimized policies. This could originate from the possible smoothing-out of transient bitrate variations in a shorter duration, facilitating a more stable prediction. It is noteworthy that the duration of predicted windows adopted in our case is a typical choice in the context of RTC to swiftly response to network dynamics. Although we have not assessed the algorithm's real-world implementation at this point, we still envision its feasibility. To provide a preliminary insight, the time required for a single prediction in a CPU environment (Intel(R) Xeon(R) Gold 6140) is merely $13.5\,\text{ms} \pm 160\,\mu\text{s}$, which does not even factor in any potential optimizations.

TABLE III: Results for different predicted time window.

Granularity	300 ms		1000 ms	
Model	RF	BitFormer	RF	BitFormer
MSE	**0.0135**	0.0162	**0.0048**	0.0092
MAE	**0.0445**	0.0512	**0.0215**	0.0325
MAPE	14.6%	**14.2%**	**12.3%**	12.5%
MSE_{peak}	0.0464	**0.0445**	**0.0167**	0.0257
MAE_{peak}	0.1037	**0.0966**	0.0502	**0.0476**
$MAPE_{peak}$	16.6%	**15.4%**	11.9%	**10.1%**
Acc_{peak}	43.8%	**49.6%**	63.5%	**73.8%**

VI. Conclusion

In this paper, we predict bitrate in the short-term future for RTC traffic by proposing a novel DL model named BitFormer, which partially incorporates a Transformer architecture and solely utilizes RTP packet-level information. To ensure the versatility and robustness of our solution, we base our work

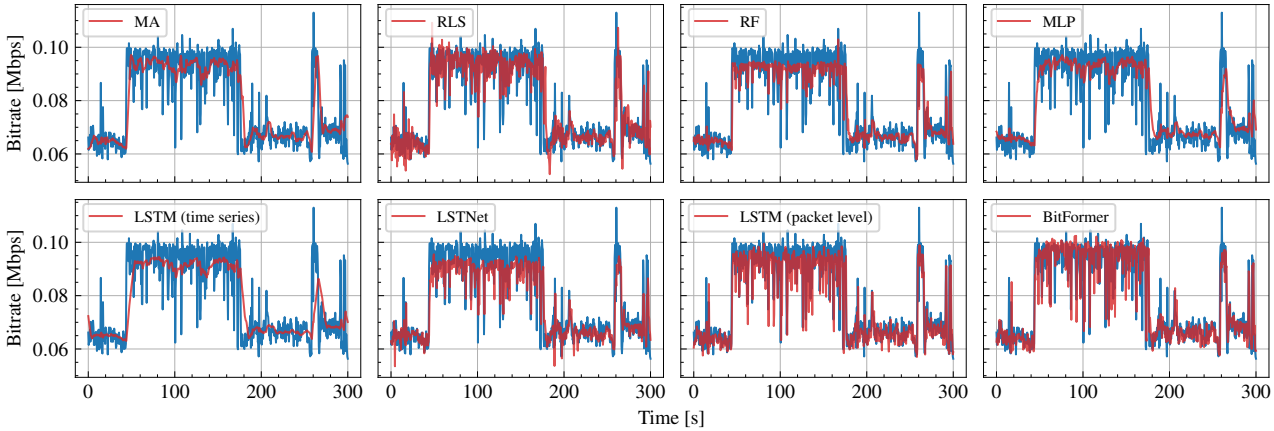


Fig. 3: Bitrates in an example of RTP flow (blue: ground truth, red: predicted value).

on ample real RTC traffic collected under various scenarios and compare our model against numerous technologies. Our proposed framework provides the merits of ease of feature extraction and delicate model architecture to tackle the constraints in RTC. As a result, BitFormer provides satisfactory overall performance and preeminent outcomes for peak values, highlighting the importance of packet-level information and illustrating the feasibility of modeling traffic dynamics. In future work, packet-level information can be further exploited, transplanting the logic to unleash the potential for predicting other RTC traffic metrics.

REFERENCES

[1] C. Athanasiadou and G. Theriou, "Telework: systematic literature review and future research agenda," *Heliyon*, vol. 7, no. 10, p. e08165, 2021.

[2] A. Nistico, D. Markudova, M. Trevisan, M. Meo, and G. Carofiglio, "A comparative study of RTC applications," in *2020 IEEE International Symposium on Multimedia (ISM)*, pp. 1–8, IEEE, 2020.

[3] R. Frederick, S. L. Casner, V. Jacobson, and H. Schulzrinne, "RTP: A transport protocol for real-time applications." RFC 1889, Jan. 1996.

[4] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 8, no. 3, pp. 1–19, 2012.

[5] J. R. Wilcox, *Videoconferencing: The whole picture*. Taylor & Francis, 2017.

[6] C. Liang, M. Zhao, and Y. Liu, "Optimal bandwidth sharing in multiswarm multiparty p2p video-conferencing systems," *IEEE/ACM Transactions On Networking*, vol. 19, no. 6, pp. 1704–1716, 2011.

[7] B. Jansen, T. Goodwin, V. Gupta, F. Kuipers, and G. Zussman, "Performance evaluation of webrtc-based video conferencing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 45, no. 3, pp. 56–68, 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[10] E. Kurdoglu, Y. Liu, Y. Wang, Y. Shi, C. Gu, and J. Lyu, "Real-time bandwidth prediction and rate adaptation for video calls over cellular networks," in *Proceedings of the 7th International Conference on Multimedia Systems*, pp. 1–11, 2016.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[12] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, "Linkforecast: Cellular link bandwidth prediction in lte networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1582–1594, 2017.

[13] M. Labonne, J. López, C. Poletti, and J.-B. Munier, "Short-term flow-based bandwidth forecasting using machine learning," *arXiv preprint arXiv:2011.14421*, 2020.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] L. Mei, R. Hu, H. Cao, Y. Liu, Z. Han, F. Li, and J. Li, "Realtime mobile bandwidth prediction using lstm neural network and bayesian fusion," *Computer Networks*, vol. 182, p. 107515, 2020.

[16] A. Lekharu, K. Moulii, A. Sur, and A. Sarkar, "Deep learning based prediction model for adaptive video streaming," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 152–159, IEEE, 2020.

[17] G. Lv, Q. Wu, W. Wang, Z. Li, and G. Xie, "Lumos: Towards better video streaming qoe through accurate throughput prediction," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 650–659, IEEE, 2022.

[18] J. Yin, Y. Xu, H. Chen, Y. Zhang, S. Appleby, and Z. Ma, "Ant: Learning accurate network throughput for better adaptive video streaming," *arXiv preprint arXiv:2104.12507*, 2021.

[19] A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapè, "Packet-level prediction of mobile-app traffic using multitask deep learning," *Computer Networks*, vol. 200, p. 108529, 2021.

[20] R. Babaria, S. C. Madanapalli, H. Kumar, and V. Sivaraman, "Flowformers: Transformer-based models for real-time network flow classification," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 231–238, IEEE, 2021.

[21] A. Dietmüller, S. Ray, R. Jacob, and L. Vanbever, "A new hope for network model generalization," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, pp. 152–159, 2022.

[22] A. Dainotti, A. Pescapé, P. S. Rossi, F. Palmieri, and G. Ventre, "Internet traffic modeling by means of hidden markov models," *Computer Networks*, vol. 52, no. 14, pp. 2645–2662, 2008.

[23] R. J. Hyndman, "Moving averages.," 2011.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[25] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

[26] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[27] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[28] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22419–22430, 2021.

[29] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–36, 2019.