Throughput Prediction in Real-Time Communications: Spotlight on Traffic Extremes

Throughput Prediction in Real-Time Communications: Spotlight on Traffic Extremes

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

***Abstract*—Amidst the thriving advancement of networks, further catalyzed by the COVID-19 pandemic, we have witnessed a marked escalation in the worldwide adoption of Real-Time Communications (RTC) applications. In this context, there is a compelling necessity to cultivate intelligent and robust network infrastructures and technologies. Real-time throughput prediction emerges as a promising candidate for this purpose to foster network observability and provide preemptive functions, supporting advanced system management, e.g., bandwidth allocation and adaptive streaming. Nonetheless, contemporary solutions grapple with predicting extreme conditions in traffic throughput, notably peaks, valleys, and abrupt changes. To address the challenges, we propose a Transformer-based Deep Learning (DL) Neural Network (NN), leveraging solely packet-level information and adopting a multi-task learning paradigm, to predict short-term throughput, with an emphasis on critical values. In particular, our work is grounded in voluminous traffic traces procured from real video-teleconferencing sessions, and we formulate a time-series regression problem, comparing numerous technologies, from an adaptive filter to Machine Learning (ML) and DL approaches. Conclusively, our methodology exhibits superior efficacy, especially in forecasting traffic extremities.**

***Index Terms*—Real-time communications, Real-time Transport Protocol, throughput, packet level, machine learning, deep learning.**

I. Introduction

In recent years, real-time communications (RTC) have solidified their position as quintessential tools in both professional and recreational domains, ushering in applications such as video-teleconferencing, online gaming, streaming, etc. The intensified demand for enhanced living and entertainment experiences in the post-pandemic era, together with the widespread embrace of remote work [1], has significantly buoyed the prominence of RTC applications. Nowadays, consumers are confronted with a profusion of competing applications [2] as RTC services continue to proliferate, which can be attributed to the augmented availability of bandwidth, the global expansion of network infrastructures, and the cutting-edge developments in 5G technologies. Specifically, Real-time Transport Protocol (RTP) [3] over User Datagram Protocol (UDP) remains the foundational pillar for the majority of these applications, whereas web browsers hinge on the globally recognized standard, WebRTC [4], an open-source framework built atop RTP.

To this end, there is a heightened impetus for developing advanced network technologies aimed to elevate network performance and enhance Quality of Experience (QoE). Notably, bandwidth management emerges as an auspicious prospect, proffering pivotal functionalities encompassing bandwidth allocation, dynamic transmission adjustments, throughput measurement, and traffic prioritization [5], [6], [7], [8]. Within this context, the prediction of traffic throughput assumes paramount potential, supporting an intelligent and proactive system that engenders a manifold of advantages such as optimized bandwidth allocation, an augmented QoE realized through advanced adaptive streaming and transcoding, efficient resource planning, and the alleviation of network congestion. Nevertheless, throughput prediction remains formidable, owing to the dynamic and multifarious nature of networks. Compounding this, existing solutions for time series problem suffer from the prediction of extreme conditions, that are crucial in RTC traffic.

In this paper, we propose a novel Transformer-based DL framework, that exclusively leverages packet-level information for throughput prediction. The sequential progression of packet flows mirrors the problems in Natural Language Processing (NLP) domain, which has been revolutionized by the Transformer [9] architecture. This congruence endows our proposed model with the capability to adeptly discern the dynamic and inherent intricacies of networks. Predominantly, our attention is riveted on peak values, valley values, and abrupt changes in throughput, and for this purpose, we deploy a multi-task learning schema combined with an assortment of trainable and predefined weights, stimulating the neural network to learn patterns of traffic extremes. Moreover, our work is firmly rooted in an extensive dataset of traffic, sourced from client endpoints during multiple video-conferencing calls with diverse network connections. We frame a time series regression problem in two distinct manners, comparing various techniques that either employ historical samples or packet-level information as features. As a result, our model outstrips the baselines, particularly showcasing prowess in forecasting traffic extremities. In order to further validate the enhanced performance, we conduct two ablation studies and briefly examine the practical viability of the model. We also make our dataset and model publicly accessible to foster research reproducibility[1].

II. Related work

In this section, we provide an overview of literature related to throughput and packet-level prediction.

Throughput prediction or bandwidth estimation, has gathered attention in academic research. [10] introduced a Recursive Least Squares (RLS) filter to estimate bandwidth for video calls in cellular networks, while [11] developed

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

[1]https://mplanestore.polito.it:5001/sharing/v9GGTvLhJ

a Random Forest (RF) approach to predict link bandwidth in 4G Long Term Evolution (LTE) networks. The authors in [12] employed general Internet traffic and adopted multiple ML algorithms to predict short-term bandwidth based on features extracted from aggregated packets. Furthermore, a Long Short-Term Memory (LSTM) model was implemented by [13], where mobile bandwidth prediction was performed using Bayes model fusion to enhance the performance. Additionally, the authors in [14], [15] concentrate on Adaptive Bitrate (ABR) for HTTP-based video streaming, by proposing tree-based models or DL technologies to forecast throughput, which were then integrated into ABR algorithm to optimize QoE. Regarding packet-level prediction, there are limited amount of works existing. [16] aimed to predict packet-level characteristics by utilizing packet-level information with 3 predicted and 3 exogenous parameters arranged in a sequential way. The authors investigated multiple DL approaches, implementing a multitask learning algorithm, and compared the performance with respect to Markov chain and RF regressor. Moreover, both [17] and [18] referred to Transformer-based NN. The former study aimed to generalize network dynamics, based on historical packet-level information. The authors added an extra hierarchical aggregation layer before the encoder to condense lengthy sequence, predicting end-to-end delay to pre-train the model, while envisioning a replaceable decoder for other tasks. The latter work proposed FlowFormer, an ensemble architecture with attention-based encoders, attempting to classify real-time network flow types — video, conference, and download. In particular, the authors aggregated packets into different categorized bins, by comparing packet-level information like payload length with predefined thresholds, and then computed the quantities of packets in such bins as features.

To the best of our knowledge, our work represents a pioneering effort in employing Transformer-based architecture with packet-level information to predict throughput in RTC, focusing on traffic extremes to bolster performance. Note that the method in [14] can capture abrupt changes based on chunk size, which, however, is not available in RTP-based traffic. On top of that, our model has a streamlined architecture, efficiently harnessing a minimal suite of packet-level information as features. Consequently, the need for resource-intensive processes such as intricate feature extraction is eliminated.

III. Problem Statement

Herein, we motivate our work, and subsequently, present the problem formulation and an overview of the dataset.

A. Motivation

Why traffic extremes: The throughput extremes constitute critical facets in RTC traffic, affecting prediction performance and epitomizing the nuanced and intricate network dynamism. To provide context, the time series throughput of a sample traffic in our data collection (details in Section III-C) is presented in Figure 1 (top), where extremities are highlighted. Specifically, we underscore the prediction of extreme values for several reasons: *i)* Peak values embody the zenith of transmission rates and often correspond to network

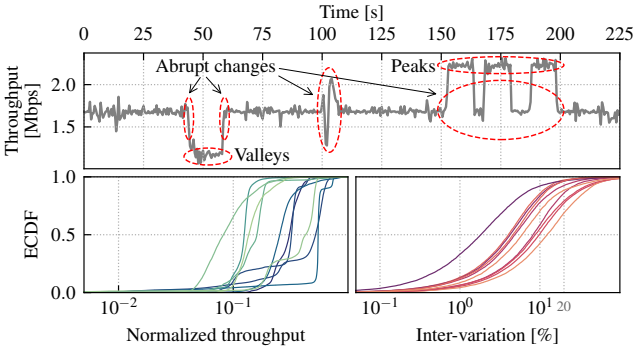

Fig. 1: Traffic patterns.

bottlenecks, thereby offering insights into the prospective bandwidth availability. Precise prediction of peaks facilitate optimal resource allocation, which, in turn, averts packet loss, degraded audio/video, diminished QoE, etc.; *ii)* Valley values denote comparatively idle periods, wherein network resources are underutilized, unveiling opportunities for energy-conservation strategies, resource redistribution, and efficacious load balancing. Furthermore, certain unexpected troughs might herald network irregularities, supporting the detection of traffic anomalies; *iii)* The ability to anticipate abrupt changes, which highlight sudden and transient network fluctuations, could facilitate more agile adaptive streaming and swift bandwidth allocation, ensuring a prompt adaptation of rapid transitions. Additionally, the depicted patterns of critical values from 10 randomly selected traffic in Figure 1 (bottom), which illustrates the Empirical Cumulative Distribution Function (ECDF) of throughput values (left) and the percentage variations between successive throughput samples (right), demonstrate that all throughput values exhibit a steep ascent in the middle, tapering into narrow tails for both ultra-low and high values, and the majority of inter-variations remain below 20%. In fact, 64.8% of inter-variations are less than 10% and 84.1% are less than 20% for all traffic. In other words, the traffic throughput generally experiences a globally stationary evolution, which underlines the significance of comprehending and forecasting traffic extremes, further rendering their prediction an intriguing and substantive endeavor.

Why packet-level information: The rationale underpinning the selection of packet-level information is threefold: First, packets stand as the fundamental unit and constitute the finest granularity within networks, encapsulating the rapidly oscillating dynamics and intrinsic nature of network traffic [19]. Predictive models, when sculpted around such meticulous features, hold a distinct advantage in discerning the underlying patterns, thereby yielding more accurate predictions. Second, the acquisition of packet-level data demands minimal exertion in terms of feature extraction, particularly in the context of RTC with possible temporal and computational constraints. More importantly, our model relies solely on packet header attributes, obviating potential complexities due to packet encryption and enabling a more streamlined workflow with expeditious access to pertinent information. Finally, packets are ubiquitously accessible across the network, transcending the confines of client sides and thus

affording a more comprehensive network observability. This broader vantage point enables the prospect of conducting throughput prediction within the network, contributing to the improvement of overarching network performance.

In this context, we select 7 elements of the RTP packet as features: 1) **Frame length**, the packet total size including all its headers and data; 2) **Inter-arrival time**, the time span between the arrivals of two consecutive packets; 3) **RTP timestamp**, the timestamp field in the RTP header; 4) **Timestamp**, the relative timestamp at which the packet arrives; 5) **RTP marker**, a single-bit field used to indicate the last packet of a specific media unit; 6) **Sequence number**, a 16-bit value that is used to identify and order the RTP packets; 7) **Flow ID**, a unique numerical identifier that is assigned to an individual RTP flow. In this way, we intend to encompass potential impact exerted by various factors, including both spatial and temporal patterns, as well as particular RTP-related incidents, by extracting directly from unadorned text in packets, thereby circumventing the necessity for resource-intensive feature engineering. Moreover, we aim to leverage the innate capabilities of the Transformer architecture with the multi-head attention, to intuitively and autonomously discern the endogenous correlations interweaving the packet-level information and traffic throughput.

B. Problem formulation

The objective is to predict the traffic throughput in a forthcoming time window of duration Δt, and we approach the problem in two ways with different features but same target: *i)* a conventional univariate time series problem with historical samples as features, and *ii)* an irregular multivariate one with preceding packet-level features. Assuming at a time instant t, we formulate a regression problem as follows:

$$\begin{aligned} &\hat{R}_t = f(X) \\ &\text{with } X = \begin{cases} r_{t-\Delta t}, r_{t-2\Delta t}, ..., r_{t-m\Delta t}, ..., & \text{if Problem } i \\ \overline{x}_{t,1}, \overline{x}_{t,2}, ..., \overline{x}_{t,n}, ..., & \text{if Problem } ii \end{cases}, \\ &n \in [1, N], m \in [1, M], \end{aligned} \tag{1}$$

where $\hat{R}_t$ denotes the predicted throughput in the subsequent time window starting at time t and ending at $t + \Delta t$, and the input feature matrix X differs between problems. For problem *i*, M historical samples are considered, and $r_{t-m\times\Delta t}$ is the previous throughput in the time window with a duration of Δt commencing from $t - m \times \Delta t$. For problem *ii*, we refer to a total number of N packets in the past, and $\overline{x}_{t,n}$ represents the feature vector of the n^{th} previous packet antecedent to time t, which includes the corresponding packet information constituted by a tuple of the aforementioned 7 elements. We aim at developing a model, mastering a function $f(\cdot)$ to undertake the regression task and map our input feature matrix X to the estimated throughput that converges with the actual value, R. Additionally, we refer to the uppermost α_p (percentage) and the nadir α_v (percentage) throughput samples during each session as the peak and valley values respectively, and define an abrupt change when the inter-variation surpasses a threshold β (percentage), i.e., a sample with throughput value R_t is deemed an abrupt change if $\frac{|R_t - R_{t-\Delta t}|}{R_{t-\Delta t}} > \beta$.

C. Dataset introduction

We employ two RTC applications, *Webex* and *Jitsi Meet*, to collect traffic traces during 71 real video-teleconferencing calls, each comprising 2 to 6 participants connected to either WiFi, mobile, or Ethernet. We collect the traffic from client sides with a total duration of nearly 70 hours, concentrating on incoming streams and storing the data in *pcap* format. In alignment with the problem formulation, we construct the dataset for each session, calculating traffic throughput in successive time windows following chronological order, by aggregating the frame lengths of all packets within each window. Importantly, each time window (target throughput) is accompanied with the packet-level features of the preceding N packets and the historical throughput samples from the previous M time windows. In our work, we compute and predict the throughput in time windows of $\Delta t = 500\ ms$, and resort to $M = 10$ prior windows (5 s) for problem *i*, while considering previous $N = 1024$ packets, that translates to an average span of roughly 3.8 s for problem *ii*. Furthermore, we devise that $\alpha_p = 10\%$, $\alpha_v = 10\%$, and $\beta = 20\%$. Noteworthily, all these selections are modifiable hyperparameters (e.g., a larger time window of $\Delta t = 1000\ ms$ or considering more large values, $\alpha_p = 20\%$), and our preliminary experiments vouch for consistent performance, for which we earmark the details for future work.

IV. Methodology

In this section, we delineate our proposed DL model and other comparative methodologies. Then, we explain the model development as well as evaluation process.

A. Introduction of proposed model

We propose a Transformer-based DL framework that utilizes packet-level information and incorporates a multi-task learning paradigm, as elucidated in Figure 2. Specifically, the sequence of packet features are injected into a packet embedding layer to enrich the features, creating feature embedding. After superimposing the trainable positional encoding, that automatically assimilates optimal positional insights, the resultant sequence of embedded features is fed into a single layer of Transformer encoder to generate the sequence of encoded features, employing multi-head attention mechanism to unearth latent patterns and apprehend network fate. Afterwards, we compute the mean value of each sequence and apply layer normalization, distilling feature essence and consummating the feature extraction phase.

Imperatively, we elaborately conceive an innovative multi-task learning strategy, augmented with multifunctional weights, to further incentivize the model to discern traffic extremes. In addition to the primary regression module, we integrate two auxiliary learning blocks: a binary classification component and a trainable multiplier. The former block aims to predict and identify whether the target throughput signifies an abrupt change with respect to the preceding sample, a feat deemed attainable due to the granular and domain-specific packet-level features, often absent in conventional time-series scenarios. The latter block operates as a calibrator that amplifies or attenuates the regression output upon the classification outcome, adjusting the final predictions to better

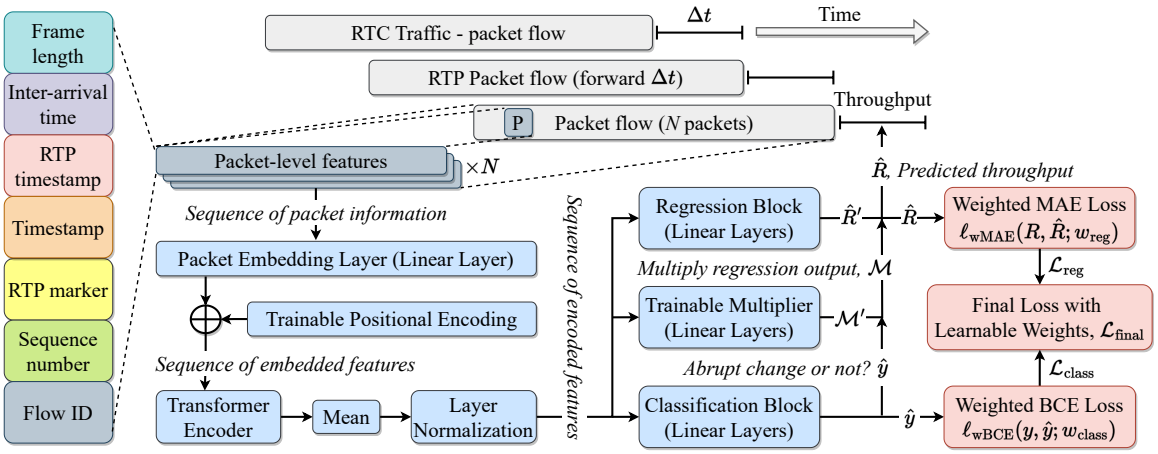

Fig. 2: Workflow, model architecture, and training strategy.

accommodate dramatic variations. As a result, we meticulously train the neural network in a way such that normal values are regularly satisfied, while abrupt changes are methodically compensated. Meanwhile, we implement learnable weights [20] that automatically determine the importance of different tasks to systematically and optimally combine losses output by different blocks as follows:

$$\begin{aligned} \mathcal{L}_{\text{final}} &= e^{-w_1} \cdot \mathcal{L}_{\text{class}} + w_1 + e^{-w_2} \cdot \mathcal{L}_{\text{reg}} + w_2 \\ \text{with } \mathcal{L}_{\text{class}} &= \ell_{\text{wBCE}}(y, \hat{y}; w_{\text{class}}), \\ \mathcal{L}_{\text{reg}} &= \ell_{\text{wMAE}}(R, \hat{R}; w_{\text{reg}}), \\ \hat{R} = \hat{R}' \cdot \mathcal{M}, \quad \mathcal{M} &= \begin{cases} \mathcal{M}', & \text{if } \hat{y} = 1 \\ 1, & \text{if } \hat{y} = 0 \end{cases}, \end{aligned} \tag{2}$$

where $\mathcal{L}_{\text{final}}$ represents the final loss, computed by melding the classification ($\mathcal{L}_{\text{class}}$) and regression ($\mathcal{L}_{\text{reg}}$) losses through trainable weights (parameters) — w_1 and w_2. Moreover, both regression and classification blocks are associated with weighted losses during training phase. On the one hand, the classification loss is calculated by weighted Binary Cross Entropy (BCE) loss function $\ell_{\text{wBCE}}(\cdot)$, with higher weights w_{class} granted to the minority samples of abrupt changes to tackle the problem of class imbalance. On the other hand, the weighted Mean Absolute Error (MAE) loss function $\ell_{\text{wMAE}}(\cdot)$ is employed for regression, with larger weights w_{reg} assigned to peaks and valleys to accentuate the model's sensitivity to such scenarios. Both y and R are ground truths of classification and regression tasks, and $\hat{y}$ symbolizes the label predicted by classification block, while $\hat{R}$ is the final forecasted throughput, ascertained by modulating the regression output ($\hat{R}'$) with the intervention of trainable multiplier ($\mathcal{M} = \mathcal{M}'$). Notably, when the classification indicates a normal transition ($\hat{y} = 0$), the multiplier remains neutral ($\mathcal{M} = 1$), leaving the regression output unaltered.

B. Model comparison, development and evaluation

We also refer to a broad range of domains, implementing multiple other technologies as benchmarks, as listed in Table I. Note that 7 and 3 models are developed for problem *i* and *ii*, respectively, as annotated by the footnotes. Specifically, RLS is a lightweight adaptive filter algorithm that recursively updates its coefficients to minimize the weighted linear least square errors. Both ML methods are tree-based ensemble models, wherein RF constructs parallel trees and XGB relies on sequential trees. MLP, also known as a deep neural network (DNN), is a simple and traditional DL model. LSTM represents one of the most successful sequence modelling NNs, comprising LSTM cells and overcoming the vanishing gradient problem prevalent in conventional recurrent neural network (RNN). LSTNet and N-BEATS are two popular time series DL models, in which the former is a multivariate model that integrates convolutional neural networks (CNNs) with RNNs, and the latter is a univariate one composed of multiple blocks with fully connected layers and residual connections. Moreover, we deliberately segment the 71 *pcap* files into 3 independent groups (50, 10, 11) to construct training (358,386 samples of throughput), validation (62,413), and test (65,698) datasets, aiming to derive a generalized solution and preclude data contamination among traffic. To evaluate the performance, we gauge 4 metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2 score). Alongside, we also measure the identification accuracy for peaks and valleys:

$$\begin{aligned} Acc_{\text{peak}} &= \frac{Num_{predicted\ value > peak\ value \times 0.9}}{Num_{peak}} \times 100\%, \\ Acc_{\text{valley}} &= \frac{Num_{predicted\ value < valley\ value \times 1.1}}{Num_{valley}} \times 100\%, \end{aligned} \tag{3}$$

TABLE I: Model summary

Category	Model
Naive baseline*	Moving Average (MA)[1] [21]
Adaptive filter	Recursive Least Squares (RLS)[1] [22]
ML method	Random Forest (RF)[1] regressor [23] XGBoost (XGB)[1] regressor [24]
DL method	Multi Layer Perceptron (MLP)[1] [25] Long- and Short-term Time-series network (LSTNet)[2] [26] Long Short-Term Memory (LSTM)[1,2] [27] N-BEATS network[1] [28]

* It calculates the average value of past throughput samples as the prediction.
[1] Problem *i*, univariate time series prediction.
[2] Problem *ii*, multivariate packet level prediction.

TABLE II: Model results: performance comparison of overall throughput, peaks, valleys, and abrupt changes.

Problem		Problem *i*							Problem *ii*		
Feature		Historical throughput samples							Packet-level features		
Model		MA	RLS	RF	XGB	MLP	LSTM	N-BEATS	LSTM	LSTNet	Ours
Overall value	MSE ↓	0.0841	0.0735	0.0481	0.0518	0.0519	0.0525	0.0490	0.0503	0.0533	**0.0466**
	MAE ↓	0.1462	0.1284	**0.1121**	0.1183	0.1224	0.1209	0.1122	0.1163	0.1208	0.1128
	MAPE ↓	14.1%	12.1%	11.3%	10.9%	14.6%	14.8%	**10.7%**	11.6%	11.2%	**10.7%**
	R^2 ↑	0.8686	0.8851	0.9248	0.9191	0.9189	0.9179	0.9234	0.9215	0.9169	**0.9273**
Peak value	MSE ↓	0.2362	0.2349	0.1714	0.1898	0.1756	0.1813	0.1759	0.1839	0.1965	**0.1630**
	MAE ↓	0.3036	0.2694	0.2532	0.2710	0.2653	0.2672	0.2558	0.2648	0.2691	**0.2360**
	MAPE ↓	14.7%	13.3%	12.2%	13.3%	13.3%	13.2%	12.7%	13.1%	12.9%	**11.8%**
	R^2 ↑	0.7570	0.7584	0.8237	0.8048	0.8194	0.8136	0.8191	0.8095	0.7964	**0.8311**
	Accuracy ↑	51.3%	60.3%	59.6%	51.6%	57.6%	55.1%	55.5%	59.6%	56.6%	**65.1%**
Valley value	MSE ↓	0.0893	0.0475	0.0324	0.0300	0.0395	0.0384	0.0273	0.0269	0.0285	**0.0216**
	MAE ↓	0.1324	0.0946	0.0877	0.0803	0.0961	0.0945	0.0772	0.0762	0.0784	**0.0708**
	MAPE ↓	27.9%	22.4%	20.5%	18.7%	28.6%	30.2%	19.7%	19.4%	18.5%	**16.1%**
	R^2 ↑	-0.0956	0.4178	0.6026	0.6320	0.5143	0.5287	0.6645	0.6666	0.6458	**0.7316**
	Accuracy ↑	57.8%	**68.1%**	60.0%	61.4%	56.6%	58.2%	59.0%	60.8%	65.7%	67.9%
Abrupt change	MSE ↓	0.3119	0.3507	0.2749	0.2881	0.2846	0.2845	0.2884	0.2742	0.2800	**0.2668**
	MAE ↓	0.3730	0.3965	0.3654	0.3764	0.3823	0.3790	0.3767	0.3677	0.3740	**0.3627**
	MAPE ↓	42.5%	44.0%	41.1%	39.7%	44.9%	45.0%	38.8%	39.5%	38.9%	**37.2%**
	R^2 ↑	0.5883	0.5371	0.6371	0.6197	0.6243	0.6245	0.6194	0.6369	0.6291	**0.6466**

in which, we introduce a tolerance margin of 10% to identify the quantity of predictions proximate to their corresponding ground truths (exceed/beneath the relaxed peaks/valleys). The accuracy proffers an insight into the performance from a classification standpoint, yet for overall values and abrupt changes, such a notion is not available.

V. Experimental result

In this section, we present the experimental outcomes for overall and critical values, as showcased by Table II.

We commence with the overall performance outlined in the first part of the table. Generally, our proposed model manifests commendable efficacy, securing the highest ranks across most numerical metrics. Although certain models (such as RF, XGB, N-BEATS, LSTM-*ii*, LSTNet) produce outcomes that are ostensibly on par, e.g. N-BEATS yields similar MAPE of 10.7%, they invariably fall short in other metrics, like LSTNet's declined R^2 score of 0.9169. Intriguingly, our model does not unilaterally outshine its counterparts in terms of MAE, which could stem from our dedicate pursuit of prioritizing the forecast of edge cases, imposing marginally aggressive predictions for several normal values, and thus resulting in a slightly inferior MAE of 0.1128. However, the degradation is arguably minuscule, being merely 0.0006 greater than the premier MAE of 0.1121, which is further consolidated by the best MAPE, suggesting that predictions deviating from true values do so in a rather benign manner. As for plateaus and troughs, our solution demonstrates significant improvement compared to others. For peak values, our model stands unrivaled, boasting the preeminent R^2 score (the only one above 0.83) as well as unparalleled identification accuracy (4.8% higher than the subsequent best), and lowest errors (e.g., the only MAPE beneath 12%), representing accurate forecasting rather than mere overestimation. Meanwhile, the supremacy becomes even more pronounced for valley values. Our model delivers markedly diminished errors and remarkable coherency, as evidenced by a MAPE that is reduced by 2.4% and an R^2 score augmented by 0.065 relative to their respective second-best values. It is pivotal to underline that RLS attains the highest accuracy and LSTNet claims the third rank, but their numerical metrics are unacceptably deficient, illustrating a propensity for them to simplistically undervalue valleys, while our model with the second-highest accuracy of 67.9% is deemed reliable given other decent numerical indicators. Moving forward to abrupt changes, our model persistently outperform the others with optimal performance across the board, exemplified by being the sole model with an MSE under 0.27. However, it remains intrinsically challenging to precisely predict such rapid and sudden transitions, given the relatively subpar performance regardless of the models. The non-ideal result originates from the inherent complexities entwined within the problem per se. Instantaneous fluctuations of throughput in the context of RTC could be spurred by a plethora of factors, like emergent traffic flows or network disruptions, elements which might not manifest conspicuously in the packet flows received by end-users, rendering them elusive and daunting to be detected by the models. Yet, against this backdrop of challenges, our approach adeptly leverages granular packet-level insights coupled with a meticulously designed model architecture, enabling it to adapt to abrupt changes to a commendable extent, thus proffering results that, while not stellar, remain respectable.

Additionally, Figure 3 provides a vivid visual representation, juxtaposing ground truth values against model predictions with MAE, both for traffic entirety (bottom) as well as extremes (4 zoomed-in blocks). In general, all models seem capable of tracing the basic traffic evolution. Regarding the abrupt changes (*A*), our model can swiftly and precisely adapts to the sudden rises, excelling the others and obtaining the minimal error. However, each model exhibits a latency in response to the initial abrupt transformation, reaffirming the notion that it is barely possible to predict drastic transitions, which remains an open problem in conventional time series prediction. Concurrently, even though the overall traffic patterns are generally stable (Figure 1),

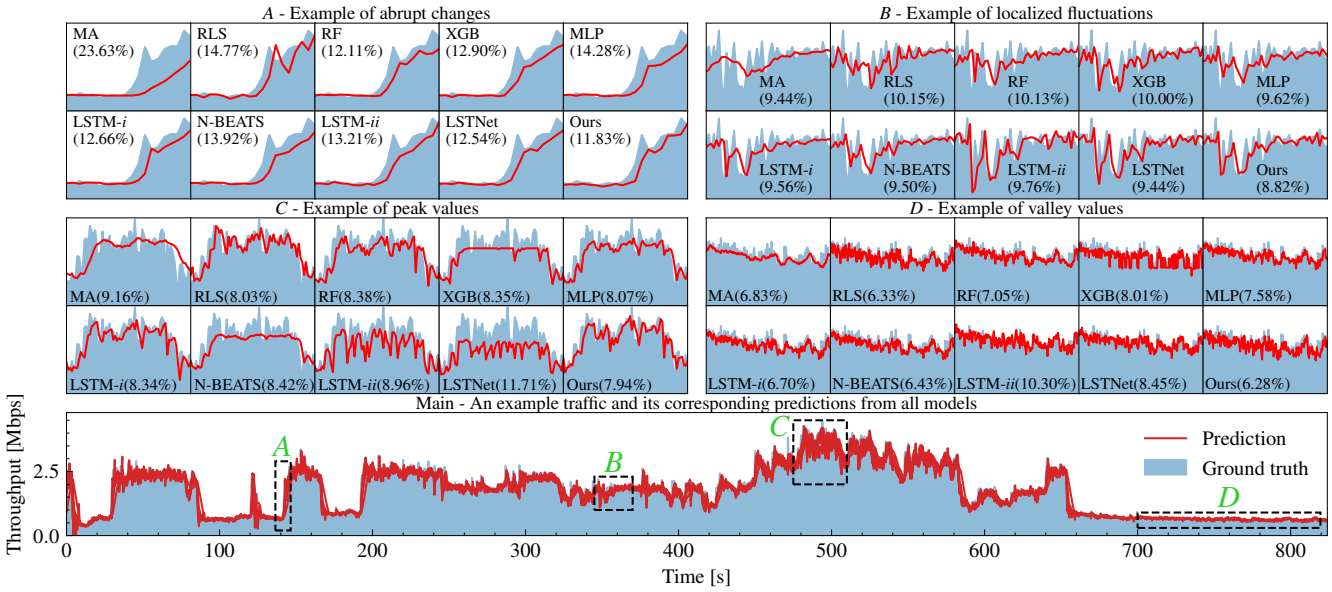

Fig. 3: Throughput Time Series: Example Traffic in Test Dataset and Predictions (use MAE in parentheses for performance comparison when predictions are visually indistinct).

it is inevitable to experience localized perturbations (*B*). Such capricious variations pose formidable challenges, but our solution still generates superior performance, delivering adept predictions that strike a balance between being neither aggressive (LSTM-*ii*) nor conservative (MLP). Furthermore, our model continues to outperform its peers, with the lowest error regarding peaks and valleys. In example *C*, the majority of models tend to underestimate the peaks, while in example *D*, our model presents greater fidelity to the fluctuations, although performance disparities are relatively subtle. It is important to recognize the intrinsic complexity of predicting throughput in RTC, especially when it concerns traffic extremities, because they represent the minority in throughput characteristics, which resembles the dilemmas in imbalanced ML problems [29].

VI. Discussion

In this section, we provide supplementary analyses, performing ablation experiments, and investigating model practicality.

A. Ablation study

To substantiate the efficacy of particular designs within the model architecture, we undertake two ablation tests: 1) we substitute the Transformer-based component with an LSTM neural network to extract features, channelling the output into subsequent multi-task learning blocks, and 2) we omit the multi-task learning schema, exclusively retaining the regression task to forecast throughput.

Table III presents the results, revealing that the original model generally surpasses both scenarios, albeit with several minor exceptions. Firstly, the LSTM-based model in test 1 encounters performance degradation, especially concerning valleys and abrupt changes, while yielding comparable outcomes for peaks, which unequivocally illustrates the superiority of Transformer architecture. On top of that, it also transcends the vanilla LSTM model with packet-level features in Table II, thereby affirming the applicability as well as competence of multi-task learning blocks. Secondly, the removal of multi-task learning paradigm in test 2 results in noteworthy performance drop, as evidenced by a decrement of 0.1217 in R^2 score for valleys, which indicates an underestimation even the identification accuracy remains similar. Interestingly, we obtain a slightly improved result regarding peak values, potentially stemming from the fact that the Transformer architecture with packet features innately advocates for traffic pinnacles.

TABLE III: Result of ablation study.

Scenario		Ablation 1	Ablation 2	Original*
Overall values	MSE ↓	0.0503	0.0500	0.0466
	MAE ↓	0.1170	0.1195	0.1128
	MAPE ↓	10.9%	11.6%	10.7%
	R^2 ↑	0.9216	0.9220	0.9273
Peak values	MSE ↓	0.1584	0.1497	0.1630
	MAE ↓	0.2406	0.2253	0.2360
	MAPE ↓	12.3%	11.5%	11.8%
	R^2 ↑	0.8359	0.8449	0.8311
	Accuracy ↑	65.1%	65.6%	65.1%
Valley values	MSE ↓	0.0251	0.0314	0.0216
	MAE ↓	0.0738	0.0841	0.0708
	MAPE ↓	16.8%	19.8%	16.1%
	R^2 ↑	0.6884	0.6099	0.7316
	Accuracy ↑	63.1%	67.5%	67.9%
Abrupt changes	MSE ↓	0.2802	0.2807	0.2668
	MAE ↓	0.3732	0.3757	0.3627
	MAPE ↓	38.4%	39.6%	37.2%
	R^2 ↑	0.6290	0.6282	0.6466

* The identical result is retrieved from Table II for a straightforward comparison.

B. Model practicality

Although it is challenging to evaluate the real-world implementation of our model at this point, we nonetheless envision its feasibility. To provide context, the execution of a single prediction merely requires $14.8\,\text{ms} \pm 1.05\,\text{ms}$ in

TABLE IV: Result of 512 packets as features.

Scenario	MSE	MAE	MAPE	R^2	Accuracy
Overall values	0.0475	0.1142	10.9%	0.9259	-
Peak values	0.1676	0.2432	12.2%	0.8263	63.3%
Valley values	0.0225	0.0748	17.1%	0.7200	66.3%
Abrupt changes	0.2705	0.3696	37.7%	0.6417	-

a CPU environment (Intel(R) Xeon(R) Gold 6140), devoid of GPU acceleration, which does not even incorporate any possible optimizations, such as efficient Transformer architectures [30] and well-established network pruning technologies [31].

To consolidate the concept, we examine the possibility of employing even less packets as features to further curtail the model complexity. In particular, we adhere to the general architecture while halving the original quantity from 1024 to 512 packets prior to a target, leading to a reduction of roughly 70% in the overall parameter count. As delineated by the result in Table IV, our model equipped with fewer features continues to generate decent performance with only marginal decline versus the original one, still boasting the highest rank in Table II with respect to other models, and thus reaffirming the practicality and feasibility.

VII. Conclusion

In this paper, we intend to predict the RTC traffic throughput with the emphasis on traffic extremes, namely peaks, valleys, and abrupt changes. We propose a novel DL solution with Transformer-based architecture, which incorporates a multi-task learning approach and exclusively utilizes RTP packet-level information. To reinforce the model universality and resilience, our work anchors on ample RTC traffic collected under various scenarios, and we 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, we obtain satisfactory overall performance with preeminent outcomes for traffic extremes, highlighting the salience of packet-level information and illustrating the feasibility of modelling traffic dynamics. Future work could consist of reducing the complexity and introducing explainability for our model. Furthermore, we remain open to incorporating exogenous factors, such as the router queue length when predictions are executed at the edge node, potentially improving the performance.

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] S. Loreto and S. P. Romano, *Real-time communication with WebRTC: peer-to-peer in the browser*. " O'Reilly Media, Inc.", 2014.

[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] 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.

[9] 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.

[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] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

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

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

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

[24] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

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

[26] 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.

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

[28] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[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.

[30] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, dec 2022.

[31] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.