Towards the Detection of Unobservable Losses in Real-Time Communications

Towards the Detection of Unobservable Losses in Real-Time Communications

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

***Abstract*—Packet loss, an omnipresent issue that degrades the QoE in Real-time Transport Protocol (RTP)-based real-time communications (RTC) applications, serves as a pivotal indicator for gauging network performance. Conventionally, loss detection hinges on sequence number irregularities. However, many contemporary applications incorporate customized mechanisms that diverge from the standard, confounding loss identification. Although the actual losses are transparent to applications themselves, they remain unobservable to other entities such as network operators, hampering the prospect of overall network management and performance optimization. To address this challenge, we investigate multitudinous RTC traffic gathered across various locations and times. Consequently, we uncover two types of anomalous patterns pertaining to sequence numbers. To discern between factual losses and aberrations in RTP flows, i.e., to detect the unobservable losses, we curate three distinct datasets, aggregating packets into time bins and calculating multiple traffic statistics. Subsequently, we leverage Machine Learning (ML) technologies, training the algorithm on one dataset while testing the remaining two, to classify the loss presence in a bin. Despite the inherent hurdles posed by class imbalance and intricate traffic dynamics, we achieve decent outcomes (0.64 F1-score), effectively identifying the majority of lossy bins (0.64 recall) while guaranteeing the performance for lossless scenarios (0.94 recall).**

***Index Terms*—Real-time communications, RTP, packet loss, machine learning.**

I. Introduction

Real-time communications (RTC) have experienced an unprecedented development and consolidated their position as indispensable tools in recent years, supporting services such as videoconferencing, streaming, and online gaming, to play a profound role in remote work, recreation, education, and beyond. As RTC applications continue to proliferate, observing and understanding the Quality of Experience (QoE) are of paramount significance for stakeholders including end-users, network operators, and service providers. Standing out as one of the key indicators for QoE metrics, packet loss is an inevitable and adverse event for traffic of Real-time Transport Protocol (RTP) [1], which underpins the vast majority of RTC services, encompassing the globally ubiquitous framework for web browsers — WebRTC[1]. A packet loss/losses could diminish the QoE, resulting in information corruption, fuzzy audio/video fidelity, transmission interruption, etc,. Hence, it becomes increasingly imperative to properly mitigate the repercussions caused by packet loss, which evidently necessitates the detection of losses as a foundational step.

This work was supported by the SmartData@PoliTO center on Big Data and Data Science, and funded by Cisco Systems Inc. and the European Union under NextGenerationEU. PRIN 2022 Prot. n. 2022MWBFEE.

[1]https://webrtc.org/

Conventionally, the identification of packet loss entails a straightforward examination of sequence numbers embedded in packet headers, adhering to default protocol specifications. In the modern landscape, a myriad of RTC applications have emerged in the market [2], and many of them invent unique and bespoke mechanisms atop RTP tailored to their own requirements [3], [4]. Nevertheless, certain customized features deviate from the original RTP paradigm, such as yielding a deliberate discontinuity between sequence numbers of consecutive packets. Thus, discerning packet loss in traffic generated by such applications becomes mired in ambiguity and unreliability, attributable to the intentionally and artificially engineered "defects". While the occurrence of actual losses remains detectable by the respective software implementations, they are not perceptible to external observers due to the proprietary nature of these alterations, commonly safeguarded under closed-source environments inaccessible to third parties. Amidst the thriving advancement of network technologies, and further catalyzed by the escalating consumer demands for high QoE, measuring QoE metrics like loss rate merely for the application level proves insufficient and antiquated. First, granting end-users' equipment access to loss conditions empowers auxiliary programs to intervene, optimizing configurations to better accommodate lossy events, given the typically limited control exerted by RTC applications over device settings. Second, monitoring packet loss within network nodes, such as edge routers, offers insights into localized performance issues, facilitating the discovery of network anomalies and enabling optimization efforts from a particular vantage point. Third, the network operators and service providers that are notified with loss metrics might be able to undertake diagnosis and enact responsive countermeasures, thereby effectively managing network systems and alleviating QoE degradation.

Traditional approaches of active or passive measurement for network losses can estimate the loss rate, by sending probe packets or leveraging Management Information Base (MIB) via Simple Network Management Protocol (SNMP) [5], [6]. However, they fall short in revealing actual losses within specific end-to-end RTP flows, not to mention the inadequate accuracy of the estimation in numerous scenarios [7] and the additional resource overhead incurred, such as the injection of extra packets. Authors in [8] has introduced a novel technology to measure losses for video streaming, but the applicability of the proposed measurement mark inserted into the user data field is questionable for existing RTC applications in the market. On top of that,

the assessment of QoE metrics concerning packet loss in RTC remains relatively understudied. The works of [9]–[11] have endeavored to map Quality-of-Service (QoS) metrics to video QoE, and [12], [13] have conducted extensive evaluation campaigns on videoconferencing performance, but they presuppose the availability of packet loss knowledge. [14], [15] have tried to predict packet loss, whereas the works are valid only for regular RTP flows. Meanwhile, recent contributions by [16]–[20] have developed multiple sophisticated methodologies to estimate QoE metrics in common videoconferencing applications, yet all of them have cast a veil over the measurement of packet loss, acknowledging such a limitation in their works.

To this end, we aim to fill in the gap by analyzing and detecting the unobservable losses in RTC. We scrutinize an enormous amount of RTC traffic sourced from different vantage points, including border routers featuring both generic and anonymous RTP traces, as well as edge-nodes capturing real video-teleconferencing sessions, spanning a period of approximately three years. Consequently, we unveil two types of peculiar patterns incurred by dynamic payload type and video frame segmentation. The involved RTP flows exhibit intermittent gaps in sequence numbers that are not generated by packet loss. These unforeseen gaps, when assessed within the context of the original RTP, are mistaken as losses, obscuring the identification of genuine losses entwined within these flows. In order to tackle the problem, we craft three independent datasets, aggregating packets into 500-ms time bins and computing numerous traffic statistics. Each bin is categorized based on the existence of packet loss, with actual labels assigned to traffic conforming to RTP, and artificial labels created for unusual traffic. Subsequently, we formulate a supervised classification problem, examining multiple Machine Learning (ML) algorithms grounded in statistical features, and conducting a meticulous evaluation process. As a result, our best solution demonstrates a commendable performance, boasting class recalls of 0.94 and 0.64 for lossless and lossy time bins in anomalous traffic with unobservable losses. Our work represents the inaugural attempt to investigate, unravel and identify such anomalous traffic, thereby contributing to the holistic network management and full-stack observability. Additionally, our method does not seek to substitute the basic approach for loss detection; rather, it is conceived to operate as a supplementary tool to enhance the QoE metrics measurement.

II. Problem Context

In this section, we start with a brief introduction of the collected traffic. Then, we characterize the loss phenomenon and formulate the problem.

A. Collected traffic

The RTC traffic in our possession with a temporal span of roughly 3 years are collected from two distinct vantage points:

- Campus router — The border router is situated in our university campus to handle inbound and outbound network traffic. Specifically, we rely on the TCP STatistic and Analysis Tool (Tstat) [21] with the Cryptography-based Prefix-preserving Anonymization (Crypto-PAn) algorithm [22] to selectively capture and filter anonymous traffic, retaining only RTP packets for the sake of our analysis. Although the provenance of the collected traces is undisclosed, we anticipate a diverse spectrum of traffic in terms of applications, mediums, locations, etc., owing to the global origins of these RTP flows frequenting the university's network at different times.
- End-user device — The users' devices, referred to as edge-nodes, actively generate traffic during multiple real video conferences (71 calls and 70 hours in total) with 2 to 6 participants connected to either Ethernet, mobile, or WiFi. We adopt 2 video conferencing applications (VCAs), namely *Jitsi Meet*[2] and *Webex*[3]. Notably, it is confident that both applications comply with RTP, thus ensuring the reliability of loss identification based on sequence numbers. Additionally, we employ Wireshark[4] to capture transmitted RTP packets during a session.

Notably, all traffic is archived into *pcap* format, and we only focus on the inbound traces, since they traverse the entire network path, reflecting the overarching network patterns affected by various factors, e.g., congestion, during transmission.

B. Packet loss

According to RTP specifications, the packet loss can be detected using sequence numbers on a per-flow basis. An RTP flow is determined by a tuple composed of (IP_{src}, IP_{dst}, $Port_{src}$, $Port_{dst}$, *SSRC*, $Type_{payload}$). For each flow, the first sequence number is initialized randomly, and an increment of one is added for each RTP packet produced. Therefore, a lost packet results in a missing number in the monotonically increased sequence, i.e., a loss/losses are identified if an RTP flow manifests a gap within successive sequence numbers. Conversely, we investigate the traffic from the campus router, discovering two types of anomalies, as portrayed in Figure 1. In particular:

- Dynamic payload type — The type of payload denotes the specific media format encapsulated in a packet. The descriptor "dynamic" means that the payload type is not consistently tied to a particular content category, e.g., audio, but rather is supposed to remain fixed within an individual RTP flow. Unusually, certain traffic occasionally alter the payload type of some packets within the same flow, while other attributes, including sequence numbers, remain compliant. When examining the presence of losses, such packets with modified payload types but ordered sequence numbers are segregated from the flow, leading to deficiency in the sequence numbers and thus mistakenly identified losses.
- Video frame segmentation — Due to the relatively large size of a video frame, it is not efficient to encapsulate an entire frame within a single packet. Modern codecs

[2]An open source framework, https://meet.jit.si/
[3]A commercial application, https://www.webex.com/
[4]https://www.wireshark.org/

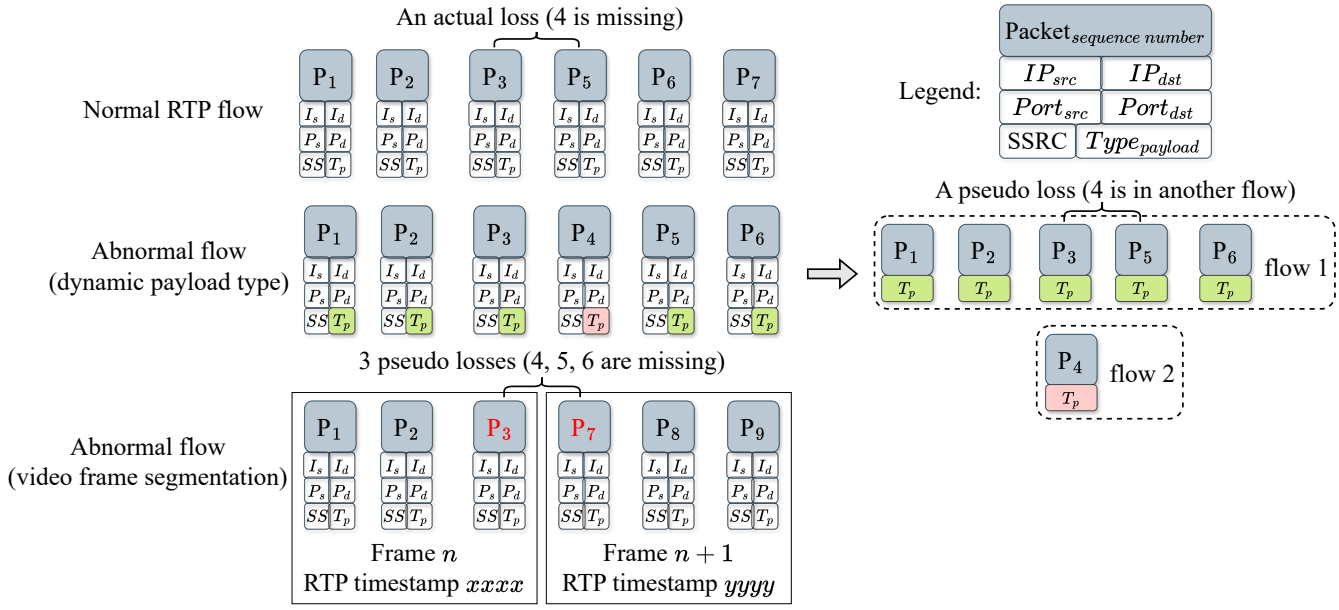

Fig. 1: The pattern of abnormal RTP flows.

typically segment an individual video frame into multiple fragments, each packetized separately. In essence, a series of continuous packets are collectively decoded as a single frame for video flows, discernible through the identical RTP timestamp shared by packets from the same frame, thanks to their simultaneous generation. In general, there is no exception for sequence number in video traces, but some flows exhibit irregular gaps in sequence numbers when RTP timestamp changes from one series of same numbers to another. This indicates that the generators of such flows intentionally introduce a transition between two consecutive video frames, i.e., the sequence number difference between the last packet of a preceding frame and the first packet of a subsequent frame exceeds one, engendering indistinguishable fake losses.

The actually losses occurred in these abnormal RTP flows become unobservable, overshadowed by pseudo losses induced by the aforementioned factors, significantly impairing the effective detection of real loss rate and hindering the reliable observability of QoE metrics. Noteworthily, while the traces are collected in an application-agnostic manner because of the traffic anonymization, we manage to glean insights into the possible software origins through the reverse DNS lookup. As a result, we are able to decipher a portion of domains pointing to certain applications, including *Google Meet*, *Microsoft Teams*, etc. However, the analysis regarding the pattern and reason behind these applications to introduce peculiar mechanisms lies beyond our current scope, and we intend to investigate pertinent facets in the future.

Importantly, although we already comprehend the underlying patterns of abnormal traffic, we refrain from simply applying a rule to eliminate these anomalies for several reasons: *i)* The elucidated mechanisms merely represent current solutions adopted by certain applications. As RTC services and network technologies continue to prosperously evolve, it is foreseeable that more advanced applications tend to emerge, probably introducing novel techniques that further complicate loss detection, not to mention the possible updates of present technologies that may change the rules again. Thus, imposing multiple rules becomes untenable, especially considering the challenge of staying abreast of both existing and emerging technologies. *ii)* Our understanding of the impediments to loss detection stems from having full access to all relevant packet information post-traffic collection, which is, however, impossible in reality, due to the trend of increasingly prevalent and strict packet encryption [23], [24]. Therefore, applying rules might be infeasible in the first place because of the unavailability of certain attributes necessary for their implementation. *iii)* The employment of rules carries the risk of rendering loss detection unreliable. For instance, a loss/losses may coincidentally occur exactly between two video frames, where an intentional gap exist in sequence numbers, but is ignored alongside the actual loss if the rule is applied. In such cases, we may inadvertently overlook the actual loss, leading to inaccuracies in detection.

C. Problem statement

As long as RTC applications adopt RTP under the hood, we posit that the traffic characteristics could provide insights into the patterns of packet loss, e.g., a loss might elongate the inter-arrival time. Given the massive and heterogeneous nature of traffic at our disposal, we assert that it is plausible to extract insights regarding packet loss embedded in the attributes of RTP flows. To this end, we aim to utilize these patterns to identify the existence of losses in an aggregated manner. Specifically, we refer to a new concept of an RTP link, denoted by a tuple of (IP_{src}, IP_{dst}, $Port_{src}$, $Port_{dst}$, *SSRC*), that is the definition of an RTP flow devoid of $Type_{payload}$. For all packets associated to a certain link, we

TABLE I: Dataset information.

Dataset	Campus-2020	Campus-2023	VCA-2020
Collection period	2020-01 → 03	2023-03	2020-04 → 2021-01
#total packets	116,230,197	96,047,339	56,269,893
#total time bins	4,478,085	4,468,413	2,008,073
#lossy bins	58,819 (1.3%)	23,812 (0.5%)	32,734 (1.6%)
#bins (abnormal)	387,043	286,451	–
#lossy bins	4,065 (1.1%)	1,122 (0.4%)	–

aggregate them into consecutive and chronologically ordered 500-ms time bins, in which we discern the presence of packet loss and compute various statistics to represent traffic patterns. Generally, our goal is to develop ML algorithms capable of classifying time bins into two categories: lossless (class 0) and lossy (class 1) based on the computed statistics.

III. Methodology

Herein, we introduce the dataset with a focus on statistical features. Following this, we delineate considered ML models and the performance evaluation process.

A. Dataset introduction

Dataset. Based on different periods and locations for the traffic collection, we construct three independent datasets, namely Campus-2020, Campus-2023, and VCA-2020. Traces are segmented accordingly, with packets associated with individual links extracted and grouped. For the packet flow of each link, we conceive sequential time bins, aggregating packets into 500 ms intervals[5]. As a result, each dataset comprises time bins with their corresponding attributes including timestamp, statistical features, class label, etc,. Table I lists the related information. For the Campus datasets, abnormal traces account for roughly 6% to 9% of time bins, which assume a substantial proportion, given the comprehensiveness and diversity of the overall traffic. Additionally, we also offer insights into the occurrence of packet loss by inspecting the duration between two adjacent lossy bins for each dataset. According to the Empirical Cumulative Distribution Function (ECDF) plots in Figure 2, around 40% of lossy bins experience another loss or losses in its subsequent 500 ms (duration between two bins is 0 s), and the majority (nearly 80%) of durations are less than 10 s, indicating a high likelihood of encountering a series of losses. Such bursts of losses are common in RTC, as the primary causes of packet loss, e.g., network congestion, normally are not instantaneous events, but endure, exerting a lingering influence. This further underscores the impetus of effectively measuring losses, as successful observations may herald extra future losses, enabling proactive interventions to mitigate issues and potentially prevent loss occurrence.

Statistical features. Utilizing all packets in each time bin, we calculate multiple traffic statistics inspired by [3]. Differently, we solely refer to attributes available in IP/UDP headers to circumvent the potential complication arising from the packet encryption [16]. In particular, we derive 29 features, encompassing statistics for UDP length, inter-arrival time (IAT), number of packets, and bitrate. Detailed descriptions are provided in Table II. For example, iat_{mean} represents the mean value of IATs of all packets within a time bin. On top of that, we extend our horizon beyond a current time bin, also incorporating information from its preceding bin as additional features, since we postulate that a lossy bin could introduce certain variations comparing to its neighborhood. Consequently, we arrive at a total of 58 features for a single time bin (target).

[5]At first glance, the 500 ms may appear to be an empirical choice; however, our preliminary results demonstrate consistent performance for different window sizes, a topic we will elaborate on in future works.

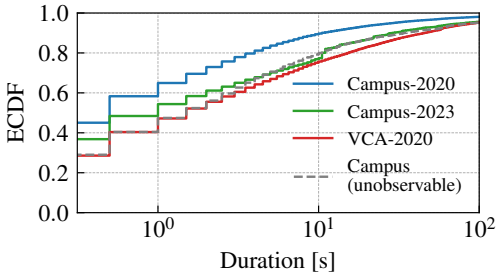

Fig. 2: ECDF of duration between adjacent losses (unobservable losses in both Campus datasets are highlighted to confirm an unified pattern).

Class labelling. Although the datasets are constructed on a per-link basis, the class label —whether a time bin is lossless (0) or lossy (1)— is still assigned on a per-flow basis. Class labelling for normal RTP flows in all datasets is straightforward by examining the missing sequence numbers. Notably, we employ the default jitter buffer size of 50 for WebRTC [25], considering also out-of-order packets that arrive too late as losses. As for abnormal flows present in both Campus datasets, we need to artificially inspect the traffic, applying the rules to eliminate pseudo losses — locating the missing packet with a different payload type in another flow and disregarding gaps among sequence numbers caused by frame transitions. While we remove abnormalities for labelling purposes, we still annotate the corresponding bins with marks for future reference.

B. ML models & Evaluation process

In order to solve the supervised binary classification problem, we resort to multiple ML classifiers: k-Nearest-Neighbors (kNN) [26], Gaussian Naive Bayes (GNB) [27], decision tree (DT) [28], random forest (RF) [29], and eXtreme Gradient Boosting (XGB) [30]. Crucially, our problem is compounded by an inherent challenge of class imbalance — only around 1% of time bins are lossy, resulting in the domination of the majority class (lossless bins) and consequent poor performance of the algorithms. To mitigate the issue, striking a balance between minimizing misclassifications for class 0 and ensuring acceptable performance for class 1, we leverage a simple under-sampling strategy with more but not overmuch lossless bins, randomly selecting twice as many lossless samples as lossy bins during training phase instead of using all available data.

TABLE II: Traffic features computed in each time bin.

Category	Shared features	Unique feature
UDP length	mean, max, min, std, kurtosis, skewness, $3^{rd}/4^{th}$ moment, max-min difference, m-m ratio[1], %max	#unique length, %unique length
Inter-arrival time		#IAT ($> 50^{th}$ percentile)[2]
Others	–	#packets, kbps

[1] $max\text{-}min\ ratio = max/(max+min); min\text{-}max\ ratio = min/(max+min)$.
[2] Number of IATs that are greater than the 50^{th} percentile of all IATs in Campus-2020 dataset.

More importantly, we devise an elaborate performance evaluation process to select the most proficient model, justify the generalizability, and assess the suitability for abnormal traffic. In general, we consider the earliest Campus-2020 dataset as a basis, with VCA-2020 serving as a reference, and Campus-2023 used for evaluating the ultimate performance. Initially, we perform 50 trials of random segmentation for the Campus-2020 dataset, partitioning the data into 70% for training and 30% for testing in a stratified manner. This process resembles cross-validation, but instead of shuffling the entire dataset, we opt to shuffle the links to ensure that samples from individual links are exclusively allocated to either the training or test sets. Throughout each trial, we train and assess all candidate models, recording the evaluation metrics to identify the best-performing option. Noteworthily, due to the severe class imbalance in our dataset, not all evaluation metrics for binary classification are useful, e.g., precision is biased towards the majority class. Hence, we adopt class recalls to examine each individual class, and the macro-average F1 score for overall performance. After discovering the optimal choice, we retrain the model on the entire Campus-2020 dataset and then apply it to the remaining two datasets, with the aim of demonstrating the versatility and generalizability under various scenarios, including times, locations, applications, etc,. On one hand, with the VCA-2020 dataset composed of traffic from known applications that adhere to RTP, we investigate the model to verify its competence, given the reliability of class labels in this case. On the other hand, with the Campus-2023 dataset, not only do we further consolidate the assessment on normal traffic, we also focus on the anomalous traffic, examining the model's efficacy for time bins with ambiguous gaps among sequence numbers.

IV. Experimental Result

In this section, we showcase the experimental findings throughout the evaluation process.

To commence, the bar plots in Figure 3 present the performance metrics in terms of class recalls and macro-average F1-score attained by all considered models across the complete evaluation trials. All models share a similar behaviour of lossless time bins outperforming lossy ones. GNB turns out to be the worst choice with both class 1 recall and F1-score below 0.5, while kNN and DT yield intermediate performance levels. Evidently, RF and XGB excel the others with recall for class 1 surpassing 0.8 and F1-score hovering around 0.6. Given the relatively lightweight and highly configurable nature of XGB, we opt for the model for the subsequent analysis.

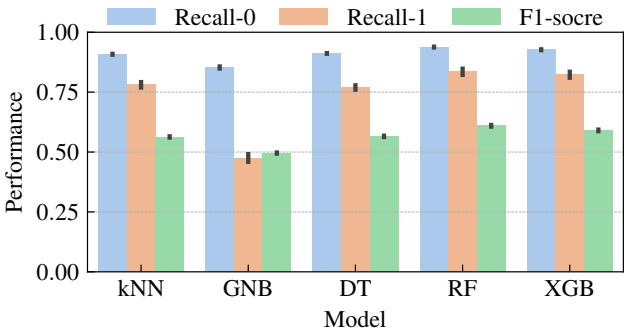

Fig. 3: Overall performance with 95% confidence interval of all models after all evaluation trials.

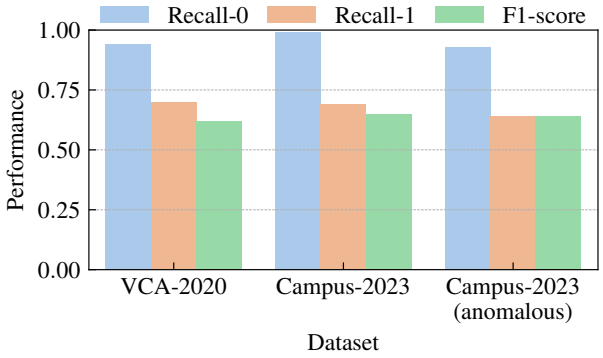

Fig. 4: Final performance evaluation.

Subsequently, we proceed to retrain the selected XGB model on the entire Campus-2020 dataset[6], intending to encompass various facets from diverse traffic, whether anomalous or not, and evaluate the performance from three different perspectives, as illustrated in Figure 4. Starting with the VCA-2020 dataset, our solution bolsters a commendable outcome, successfully identifying 70% of lossy time bins, which substantiates the reliability, given that the model functions efficaciously for traffic with assured presence of packet loss. Moving forward to the Campus-2023 dataset, the performance escalates, showcasing an exceptional class recall of 0.99 for lossless bins, consequently yielding the highest F1-score of 0.65. While these results affirm the model's proficiency, our primary focus lies on time bins in which discontinuities among sequence numbers exist, because the unobservable loss can only occur in such circumstances.

[6]Besides all lossy bins, we still refer to the under-sampling strategy for lossless bins.

Therefore, we use the aforementioned marks to identify and isolate abnormal bins from the Campus-2023 dataset, evaluating the model to ascertain the final performance. Although we observe a marginal decline for both classes, the overall performance remains respectable, boasting an F1-score of 0.64 with recalls of 0.94 and 0.64 for classes 0 and 1, respectively. Given that the labels of such time bins are artificially generated, the performance degradation might be induced by mislabelling, and thus the model overfitting is acceptable to a certain degree. In general, due to the intrinsic difficulty posed by the serious class imbalance and the multifaceted characteristics from diverse traffic, it is arduous to improve both classes simultaneously. Nonetheless, our efforts still obtain satisfactory and generalized outcomes, adeptly recognizing the majority of unobservable losses without compromising the detection of normal traffic. Additionally, the model can be tuned to favor a certain class, but this results in trade-offs such as excessive misclassifications for class 0 or inferior performance for class 1, while our finalized model achieves a judicious balance between both classes.

V. Conclusion

In this paper, we methodically examine ample RTP-based RTC traffic amassed under various conditions, revealing two types of anomalies that obfuscate the identification of packet losses and render the actual losses unobservable in such traffic. To surmount the challenge, we refer to an aggregated approach to derive traffic statistics in time bins and formulate a supervised classification problem to predict the presence of packet loss in a bin. We create three independent datasets and explore multiple ML technologies. Moreover, we conduct a meticulous evaluation process, selecting the optimal model based on one dataset while comprehensively assessing the performance on the other two. Consequently, we achieve decent performance, identifying a significant portion of lossy samples, whilst maintaining a moderate level of misclassifications for lossless bins. Our proposed solution stands poised to serve as a supplementary instrument for packet loss measurement in RTC, contributing to a transparent and dependable observability of network performance. Future work could aim to investigate thoroughly the source applications of abnormal traffic.

References

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

[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] G. Perna, D. Markudova, M. Trevisan, P. Garza, M. Meo, and M. M. Munafò, "Retina: An open-source tool for flexible analysis of rtc traffic," *Computer Networks*, vol. 202, p. 108637, 2022.

[4] Y. Dodis, D. Jost, B. Kesavan, and A. Marcedone, "End-to-end encrypted zoom meetings: Proving security and strengthening liveness," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 157–189, Springer, 2023.

[5] A. Miyamoto, K. Watanabe, and K. Ikeda, "Packet loss rate estimation with active and passive measurements," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–4, IEEE, 2012.

[6] P. Barford and J. Sommers, "A comparison of probe-based and router-based methods for measuring packet loss," *submission,(see http://www. cs. wisc. edu/pb/publications. html)*, 2003.

[7] J. Sommers, P. Barford, N. Duffield, and A. Ron, "A geometric approach to improving active packet loss measurement," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 307–320, 2008.

[8] Z. Hu and Q. Zhang, "A new approach for packet loss measurement of video streaming and its application," *Multimedia Tools and Applications*, vol. 77, pp. 11589–11608, 2018.

[9] A. Nikravesh, D. K. Hong, Q. A. Chen, H. V. Madhyastha, and Z. M. Mao, "Qoe inference without application control," in *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pp. 19–24, 2016.

[10] S. Yan, Y. Guo, Y. Chen, F. Xie, C. Yu, and Y. Liu, "Enabling qoe learning and prediction of webrtc video communication in wifi networks," in *Proceedings of the ICC*, vol. 2017, 2017.

[11] G. Carofiglio, G. Grassi, E. Loparco, L. Muscariello, M. Papalini, and J. Samain, "Characterizing the relationship between application qoe and network qos for real-time services," in *Proceedings of the ACM SIGCOMM 2021 workshop on network-application integration*, pp. 20–25, 2021.

[12] 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.

[13] M. Varvello, H. Chang, and Y. Zaki, "Performance characterization of videoconferencing in the wild," in *Proceedings of the 22nd ACM Internet Measurement Conference*, pp. 261–273, 2022.

[14] T. Song, D. Markudova, G. Perna, and M. Meo, "Where did my packet go? real-time prediction of losses in networks," in *ICC 2023-IEEE International Conference on Communications*, pp. 3836–3841, IEEE, 2023.

[15] A. Giannakou, D. Dwivedi, and S. Peisert, "A machine learning approach for packet loss prediction in science flows," *Future Generation Computer Systems*, vol. 102, pp. 190–197, 2020.

[16] T. Sharma, T. Mangla, A. Gupta, J. Jiang, and N. Feamster, "Estimating webrtc video qoe metrics without using application headers," *arXiv preprint arXiv:2306.01194*, 2023.

[17] A. Choi, M. Karamollahi, C. Williamson, and M. Arlitt, "Zoom session quality: A network-level view," in *International Conference on Passive and Active Network Measurement*, pp. 555–572, Springer, 2022.

[18] O. Michel, S. Sengupta, H. Kim, R. Netravali, and J. Rexford, "Enabling passive measurement of zoom performance in production networks," in *Proceedings of the 22nd ACM Internet Measurement Conference*, pp. 244–260, 2022.

[19] K. MacMillan, T. Mangla, J. Saxon, and N. Feamster, "Measuring the performance and network utilization of popular video conferencing applications," in *Proceedings of the 21st ACM Internet Measurement Conference*, pp. 229–244, 2021.

[20] H. Chang, M. Varvello, F. Hao, and S. Mukherjee, "Can you see me now? a measurement study of zoom, webex, and meet," in *Proceedings of the 21st ACM Internet Measurement Conference*, pp. 216–228, 2021.

[21] M. Mellia, A. Carpani, and R. Lo Cigno, "Tstat: Tcp statistic and analysis tool," in *Quality of Service in Multiservice IP Networks: Second International Workshop, QoS-IP 2003 Milano, Italy, February 24–26, 2003 Proceedings*, pp. 145–157, Springer, 2003.

[22] J. Fan, J. Xu, M. Ammar, and S. Moon, "Cryptography-based prefix-preserving anonymization," *URl: http://www. cc. gatech. edu/computing/Telecomm/projects/cryptopan/(cit. on p. 58)*.

[23] J. Uberti, C. Jennings, and S. Murillo, "Rfc 9335: Completely encrypting rtp header extensions and contributing sources," 2023.

[24] B. Marczak and J. Scott-Railton, "Move fast and roll your own crypto: A quick look at the confidentiality of zoom meetings," April 2020.

[25] Y. Cinar, P. Pocta, D. Chambers, and H. Melvin, "Improved jitter buffer management for webrtc," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–20, 2021.

[26] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[27] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, vol. 18, no. 60, pp. 1–8, 2006.

[28] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

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

[30] 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.