From Pixels to Packets: Traffic Classification of Augmented Reality and Cloud Gaming

Alireza Shirmarz, Fábio Luciano Verdi
Department of Computer Science
Federal University of São Carlos (UFSCar)
Sorocaba, Brazil
ashirmarz@ufscar.br, verdi@ufscar.br

Suneet Kumar Singh, Christian Esteve Rothenberg
School of Electrical and Computer Engineering
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil
ssingh@dca.fee.unicamp.br, chesteve@dca.fee.unicamp.br

***Abstract*—Augmented Reality (AR) real-time interaction between users and digital overlays in the real world demands low latency to ensure seamless experiences. To address computational and battery constraints, AR devices often offload processing-intensive tasks to edge servers, enhancing performance and user experience. With the increasing adoption and complexity of AR applications, especially in remote rendering, accurately classifying AR network traffic becomes essential for effective resource allocation. This paper explores two methods based on Decision Tree (DT) and Random Forest (RF) to classify network traffic among AR, Cloud Gaming (CG), and other categories. We rigorously analyze specific features to precisely identify AR and CG traffic. Our models demonstrate robust performance, achieving accuracy rates ranging from 88.40% to 94.87% against pre-existing datasets. Moreover, we contribute with a novel dataset encompassing AR and CG traffic, curated specifically for this study and made publicly available to facilitate reproducible research in AR network traffic classification.**

***Index Terms*—Augmented Reality, Traffic Classification, ML.**

I. Introduction

Extended Reality (XR), which includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), aims to enhance human interaction with digital and real-world environments. VR immerses users in entirely digital landscapes, whereas AR supplements the real world with digital overlays, and MR facilitates interaction between real and virtual elements [1, 2]. The application of XR technologies spans diverse fields, e.g., gaming, entertainment, healthcare, and education, with projections indicating that the mobile AR market will expand to four times by 2026 [3]. HMDs (Head Mounted Displays) are pivotal in XR, offering visual, audio, and sensory feedback. VR headsets deliver a completely immersive experience by isolating the user from the physical world, while AR glasses enhance real-world interactions with digital information. Current AR glasses are available in two categories: phone-powered, reliant on smartphones for computational tasks, and stand-alone, which are self-sufficient in computing [4–8]. Advancements in offloading AR processing and remote (game) rendering to edge servers are aimed at managing the computational demands by leveraging the servers' superior processing capabilities and leveraging advances in network connectivity such as 5G [9–15]. This strategic shift enhances Quality of Service (QoS) and Experience (QoE) through efficient edge cloud processing for XR and Cloud Gaming (CG) applications. Classifying network traffic for effective resource allocation remains imperative to accommodate the varying demands of different applications [16, 17]. However, challenges behind traffic classification (TC) are compounded by encryption and the use of dynamic ports, making application identification through network traffic analysis more complex. Current strategies for traffic classification include port-based, payload-based, and machine learning (ML)-based methods. By handling the hazards of payload encryption and dynamic port allocation, ML approaches are gaining prominence in accurately classifying network traffic at high performance and affordable costs across various hardware platforms, including GPUs, SmartNICs equipped with CPUs, and FPGAs [16, 18].

In this paper, we present a solution for traffic classification of AR, CG, and other applications (e.g., web-based network traffic) using flow-based features in DT and RF models. Compared with previous work on CG traffic classification [19], as far as we know, this is the first work that jointly classifies AR and CG. The contributions of this paper are:

- We propose an algorithm to classify the AR and CG from other applications based on the network traffic behavior in Uplink (UL) which is the data transmitted from the User Equipment (UE) (e.g., AR glasses, game controller) to the edge server, and Downlink (DL) i.e., the data sent from the edge server to the UE;
- We select the key features for the network traffic classification by analyzing the different possible combinations of the features. Hence, the most effective set of features is exploited to classify the network traffic in AR, CG, and other applications with high accuracy;
- We propose a DT and RF model to classify the network traffic into three classes: AR, CG, and other applications based on network flow features. The models are trained, tested, and improved with real traces of AR and CG applications;
- Finally, we collect AR and CG network traffic to test and improve the model. All the collected PCAP files and Jupyter notebooks for reproducibility are publicly available.

The rest of the paper is organized in the following sections. Section II discusses the background of VR/AR, and CG network traffic models. Section III addresses the related works. Section IV shows the proposed algorithm to classify network traffic in AR, CG, and others. Section V presents the hyperparameters tuning for improvement and the feature effectiveness analysis to select the most effective features for classification. Section VI details the AR and CG datasets that were collected in this work. Section VII reports the performance of classification examined on pre-existing and the datasets collected in this work. Then, the challenges are discussed in Section VIII. Finally, section IX concludes this paper.

II. Background

This section outlines the distinctive features of network traffic associated with VR/AR and CG. Understanding these characteristics can aid in effectively identifying these applications through network traffic analysis.

VR/AR Network Traffic. VR/AR traffic comprises two types of data flow directions: UL where data is sent from the HMD to the edge server, and DL in the other direction. In VR/AR, the HMD initiates communication with the server, generating significant UL and DL traffic due to immersive data. In VR, initial HMD messages convey position and orientation, guiding the server to select video segments matching the HMD's Field of View (FoV). In VR, UL traffic is minimal, while DL traffic is significant, attributed to the HMD's FoV. In AR, if rendering occurs within the glasses, traffic patterns mirror VR. However, when rendering is offloaded to the edge server, both UL and DL traffic intensify. AR requires significant UL bandwidth to transmit the frames of the scene, and DL bandwidth increases for returning digital objects and scene frames to the HMD, leading to considerable traffic both ways [20]. In AR, when rendering occurs within the HMD, network traffic patterns resemble those in VR, yet distinct differences set AR apart. Offloading VR/AR functions to enhance flexibility emphasizes the network's critical role in connecting HMDs and rendering servers, crucial for managing XR traffic and minimizing delay [15, 20, 21].

XR/CG Network Traffic. The study of XR and CG traffic by 3GPP, detailed in [13], highlights the distinct roles of UL and DL in traffic models. UL primarily handles 'pose and control' traffic, characterized by its light and frequent nature. Conversely, DL facilitates multimedia streaming, which is divided into two models: (a) single-stream and (b) multi-stream. The single-stream model merges video frames, audio, and data into a unified flow, while the multi-stream model separates these data types into distinct flows. The multi-stream model supports transmission over two or three distinct flows, allowing for the separation of video, audio, and data into individual streams. Alternatively, it offers the option to distribute video frames—specifically I-frames, P-frames, and B-frames—across different streams, providing flexibility in how traffic is managed. This technical report outlines statistical parameters, e.g., Periodicity (ms), Frame Rate (FPS), Data Rate (Mbps), Packet Size (Byte), Packet Delay Budget (PDB) (ms), and Packet Success Rate (%) for constructing generic VR, AR, and CG traffic models under both single and multi-stream models. However in [13], the authors note that CG's network traffic in a single-stream setup deviates from the generic model, unlike its multi-stream counterpart. According to [13], in the single-stream model, AR and VR share similarities in DL traffic but differ in UL. Conversely, in the multi-stream model, both exhibit similar DL patterns and align with the generic model in UL traffic.

III. Related Work

Research on network traffic classification, including port-based, DPI, and ML techniques, has been extensively reviewed in [16]. While many studies, e.g. [22, 23], have applied ML algorithms to identify applications and services, few have addressed CG and VR/AR classification due to the lack of labeled datasets. In [19], the authors stand out by classifying CG and non-CG traffic, collecting and publishing datasets under normal and disturbed conditions[1], with DT achieving the highest accuracy at 98.5%. However, this binary classification does not extend to multi-class classification involving CG, VR/AR, and other categories. The absence of VR/AR labeled data necessitates modeling VR/AR network traffic to generate or extract relevant traffic. The work in [24] published an AR dataset for DL traffic analysis and modeling, used in [15] for an AR scenario where a user navigates a street with AR glasses. This scenario, however, overlooks network conditions. Another approach in [21] introduces a statistical model based on Johnson's S_U distribution for generating XR traffic, validated against collected IP traffic. Our work will leverage this AR traffic model, combining it with CG and non-CG datasets for a comprehensive training dataset. We aim to train and test a classification model, then validate its performance with unseen AR, CG, and non-CG data, and also collect and publish AR and CG datasets for further research. Therefore, as far as we know, this paper is the first paper that addresses the AR network traffic classification and differentiates AR network traffic from VR, CG, and other applications to take a step forward to improve the AR network traffic prioritization and resource allocation.

IV. Methodology

In XR, the HMDs and the servers are connected with the network components (e.g. switches, routers, SmartNIC). Hence, network devices in the path between the user and server are good places to differentiate the flows and allocate the appropriate resources aiming at improved QoS and QoE.

The features chosen for classification are discussed in Subsection IV-A. Then, we propose a general classification algorithm in Subsection IV-B. The training dataset is organized in Subsection IV-C. We detail the pre-processing of the dataset in Subsection IV-D. Finally, the ML algorithm is selected to

[1] https://cloud-gaming-traces.lhs.loria.fr/data.html

train the model based on the training dataset and evaluated as mentioned in Subsection IV-E.

A. Feature selection

The network traffic in VR/AR and CG are expected to be based on a one-stream model [13]. The sensors of VR/AR glasses, e.g., camera, Inertial Measurement Unit (IMU), and audio data are fused (multiplexed) in one stream [3–6, 8]. The majority of the data belongs to the XR glasses sending or receiving high-quality video frames compared to other sensors; hence, in this paper, the audio and IMU traffic are neglected to simplify the network flow analysis. Unlike CG traffic, AR traffic is almost symmetric [20], so we can use the data rate to differentiate CG from AR traffic on the UL. Hence, one key feature that can represent the data rate in the packet flow is the *Inter-Packet Interval (IPI)*. This feature can extract the traffic behavior to differentiate the AR network traffic from CG and other applications based on network conditions. However, classifying DL traffic presents challenges due to the similarities between AR, VR, and CG traffic and other video streaming traffic, unlike the more distinguishable differences in UL. The other two key features for identifying VR/AR traffic are the resolution and frame rate. The resolution can be inferred by using the *Frame Size (FS)* and the frame rate can be obtained by calculating the *Inter Frame Interval (IFI)*. Therefore, three features are considered in this research to classify the network traffic into three classes: IPI, FS, and IFI.

IPI. It is obtained by subtracting the timestamp of the current packet from the timestamp of the previous packet as indicated in Eq. (1).

$$\begin{aligned} IPI_i &= Pkt_i[\text{Time}], \quad i = 1 \\ IPI_i &= Pkt_i[\text{Time}] - Pkt_{i-1}[\text{Time}], \quad i \in \{2, \ldots, n\} \end{aligned} \tag{1}$$

FS. FS indicates the size of each frame, which is influenced by the glasses' resolution and compression ratio. A Marker bit in the 12-byte Real Time Protocol (RTP) header marks the start of each frame. Typically, a frame's size exceeds that of a single UDP payload, necessitating multiple UDP packets for transmission. The final packet, carrying the leftover bytes, is often smaller than its predecessors. The calculation of FS is detailed in Eq. (2). In the equation, $Pkt_i[\text{Len}]$ indicates the payload size of the *i-th* UDP packet, with *i* ranging from *1 to n*, where *n* is the total number of packets for a frame.

$$FS = \sum_{i=1}^{n} Pkt_i[\text{Len}] \tag{2}$$

IFI Before explaining how to obtain this metric, we need to highlight that a frame is composed of consecutive IP packets. So, we need to identify where a frame starts and ends. To solve this, we calculate the time interval between consecutive frames, determined by subtracting the timestamp of a frame's first packet from that of the next frame's first packet, as illustrated in Eq. (3). In this equation, $F(t)$ denotes the start time of a frame, with the subscript representing the frame's sequence number.

Algorithm 1: XR/AR and CG Traffic Identification

```
Input: Ingress flow
Output: Flow Applications (XR/AR or CG or others)
/* (Step 1) Extract Flow Features                */
Extract features: IPI, IFI, FS
/* (Step 2) Determine Flow Direction             */
Identify if the flow is UL or DL using the features
/* (Step 3) Preliminary Classification           */
Classify the flow into preliminary categories: AR, CG, or
  Others based on extracted features
/* (Step 4) Confirm TC using flow UL/DL          */
if the flow is UL then
    Classify UL flows as AR, CG, or Others
else
    Classify DL flows as XR (VR or AR), CG, or Others
end
/* (Step 5) Final Decision                       */
Based on the preliminary classification and identification,
  label the flow accurately.
```

$$\text{IFI } (N) = F_N(t) - F_{N-1}(t) \tag{3}$$

The first packet of a frame can be identified through specific headers, e.g., the RTP header, where the Marker flag bit is set to 1 for the initial packet of the frame and reset to 0 for subsequent packets. Additionally, the last packet carrying the frame can also be distinguished either by a reduction in its size compared to preceding packets or by its distinct behavior.

B. General Algorithm

According to the similarities and differences among VR, AR, and CG with other applications mentioned in [13, 20], Algorithm 1 is proposed to designate which flow belongs to XR/AR, CG, or others based on the UL and DL behavior.

Algorithm 1 effectively categorizes ingress network flows into application types (VR/AR, CG, or others) by analyzing the flow's unique characteristics through a structured five-step process. The initial step extracts essential features: IPI, FS, and IFI. Next, the algorithm evaluates the flow's direction in Step 2 by analyzing its symmetry or asymmetry, with a particular focus on the asymmetric nature of CG traffic (player's commands in one direction and multimedia flow in the other), which distinguishes it from XR/AR traffic (both directions are multimedia) [19]. In Step 3, the flow is classified into one of the categories: AR, CG, or others. Step 4 advances this classification by examining and confirming the UL and DL directions by observing the IP address, transport protocols (TCP/UDP), and port numbers of the server, given that these data are well-known and determined. This step is considered based on the 3GPP network traffic model for VR/AR/CG [13] and is crucial as it helps make the model deterministic for final implementation and deployment, addressing the inherent non-determinism in ML model accuracy. The final step, Step 5, consolidates these insights to make a definitive decision, thereby enhancing the algorithm's accuracy and reliability in identifying between AR, CG, and other forms of network traffic, streamlining the methodology for this investigation.

C. *Dataset for model training*

A precisely labeled dataset is crucial for ML-based AR and CG classification, ensuring the model trains on accurate data for reliable predictions. Three essential datasets related to AR, CG, and others are needed for training.

AR Dataset. Despite the growing interest in AR, the lack of publicly available datasets specifically tailored for AR research poses a significant challenge, particularly in the area of AR network traffic classification. A viable approach to generating network traffic datasets for AR involves utilizing statistical distributions, e.g. Johnson, Poisson, Normal, and Exponential distributions. Among many existing statistical models, the Johnson $\mathrm{S_U}$ distribution model stands out [21]. This model, which includes parameters for Location (Γ), Scale (σ), Shape-A (a), and Shape-B (b) as proposed in [21], has been shown to closely match XR traffic and has been adopted for model training. Despite evidence confirming the model's ability to replicate real-world XR traffic [21], we further validate its effectiveness by testing it against pre-collected AR-specific data mentioned in [24]. This step ensures the model's accuracy from a data perspective.

Moreover, leveraging the similarity between VR and AR network traffic in DL as outlined in the 3GPP specifications [13] and XR traffic characteristics [20], we extend our training to include publicly available VR datasets [25], with additional VR datasets [26, 27] utilized for DL testing of AR traffic. This approach enhances our model's robustness in accurately classifying AR network traffic.

CG & Other Dataset. The CG traffic dataset, documented in reference [19], comprises approximately 35 GB of data and is available in both PCAP and CSV formats. The Non-CG traffic dataset, distinct from the CG dataset mentioned in [19], comprises 2 GB and includes high-bitrate UDP applications such as Video Conferencing (VC) via platforms like Discord and Google Meet, Video Streaming (VS) on YouTube and Facebook Watch, Live Video Streaming (LV) on YouTube Live, and Facebook Navigation (FN) for activities like feed browsing and profile checking. Although referred to as Non-CG in [19], this 2GB dataset is labeled as 'other' in our work. It features a diverse range of small network flows, making it ideal for representing the 'other' class in our classification training.

We process three distinct datasets—each representing a specific class (AR, CG, or others)—sourced from PCAP or CSV files. We extract relevant features from these datasets and compile them into separate, newly created CSV files, categorically labeled according to their respective classes. These labeled datasets are then merged and their samples are shuffled to prepare for model training. For the training phase, 90% of the combined dataset is utilized, while the remaining 10% is set aside for evaluation purposes.

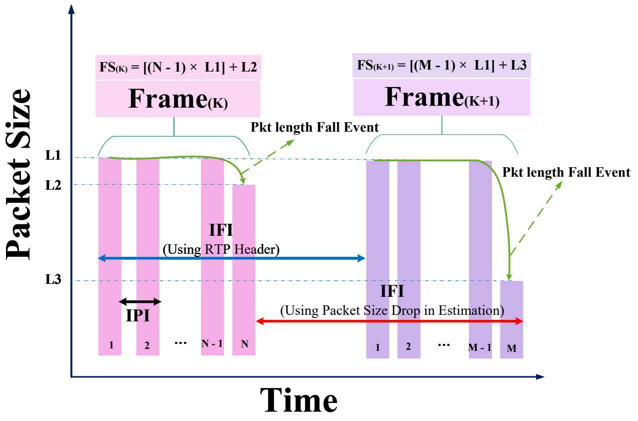


Fig. 1: IP packet flow in which frame size, inter-frame interval, and inter-packet interval time are illustrated.

D. *Pre-processing*

In this study, given the scarcity of datasets for AR and CG network traffic, we employed two approaches to prepare our training dataset: (a) generating data using the statistical model from [21], and (b) extracting features from existing PCAP or CSV files. We utilized Python and the Scapy package to parse PCAP files, while also leveraging Python to process features from CSV files[2]. In the PCAP files, we utilize packet data, e.g., source and destination IPs, source and destination ports, and transport protocol to identify network flows. Additionally, we use timestamps and packet lengths to extract the key features, IPI, FS, and IFI, as outlined in Eq. (1), (2), and (3), respectively. Extracting the FS from existing datasets presented challenges due to the occasional absence of the RTP header or the application not utilizing RTP. To overcome this, we adopted an estimation algorithm to determine FS from UDP packets, which are likely fragmented to fit the Maximum Transmission Unit (MTU) constraints. This approach assumes that a series of fully utilized UDP packets is followed by a shorter packet, forming a recognizable pattern. By analyzing the timing and length of received UDP packets, we accurately estimated FS, as illustrated in Fig. 1.

Unlike traditional methods that identify a frame's start with the RTP Marker flag, our approach focuses on pinpointing a frame's ending packet. We calculate the IFI using Eq. (1), which can either be the time difference between the end packets of two successive frames (indicated by a red arrow in Fig. 1). This technique offers a reliable means of estimating frame characteristics in the absence of RTP data. Fig. 1 demonstrates that the start and end of a frame are identifiable by the RTP Marker header and drop packet size estimation, respectively, with IFI measured by blue and red arrows. The CSV dataset files, e.g., [24], provide details including time, source and destination IPs, packet length, and the header and info fields,

[2] https://github.com/dcomp-leris/XR-AR-NTC.git

which contain source and destination ports along with payload length. From these datasets, we extract IPI, FS, and IFI.

For model training, we methodically organize, label, merge, and shuffle the samples across datasets. The training dataset contains 1200 AR samples, 128 CG samples, and 900 samples belonging to other applications. While the AR dataset benefits from augmentation via statistical models, the datasets for CG and other applications face limitations, posing a risk of imbalance. This could unfairly tilt the ML model towards over-represented classes. To counteract this, we adopt a weighted class strategy to preserve dataset balance and guarantee impartiality among all categories.

E. ML training algorithm selection

In this study, DT and RF with selected features namely IPI, IFI, and FS are considered to train the model to satisfy the future hardware requirements [17, 19, 23, 28]. In this implementation, we use Python programming language with Scikit-learn, Pandas, and Matplotlib packages. We utilize the DT classifier for its clear interpretability and strong performance with categorical data, which is further enhanced by its adaptability to programmable hardware[3]. Additionally, we employ RF, an ensemble of DTs where decisions are aggregated through voting, offering improved classification accuracy. The deployment of both DT and RF models benefits from straightforward hardware implementation [19, 23, 28]. To train our model, we meticulously prepared three datasets by first loading and labeling them. We then combined and shuffled these datasets to ensure that the data order does not introduce biases, which is vital for preventing the model from inadvertently learning sequence patterns that could skew its predictions.

Our methodology involves training the DT model with both ‘Gini’ and ‘Entropy’ criteria to gauge their effectiveness in handling impurities in tree nodes. Despite ‘Gini’s’ speed and resilience to imbalanced datasets, ‘Entropy’ demonstrated superior accuracy for multi-class classification in our context [29], prompting its selection despite the computational considerations. To counteract dataset imbalance, we adjusted class weights inversely proportional to the class sample sizes, ensuring a balanced model training approach. The model underwent training with seven possible combinations of three key features, IPI, FS, and IFI, to identify the most relevant feature set based on True Positive (TP) and True Negative (TN) rates for AR, CG, and other classes. Ultimately, the combination yielding the highest classification accuracy was selected for the final model, underscoring our commitment to optimizing performance.

V. Hyperparameters Tuning & Feature Analysis

A. DT & RF hyperparameters

To enhance the performance of the classification models, we fine-tune the hyperparameters of the DT and RF algorithms.

TABLE I: DT and RF Hyperparameters & Search Space

Parameters	**DT-UL**	**RF-UL**	**DT-DL**	**RF-DL**	**Search Range**
max_depth	20	20	None	20	[None, 10-50]
min_samples_leaf	1	1	2	1	[1-10]
min_samples_split	5	2	5	2	[2-20]
max_features	‘Sqrt’	‘Sqrt’	‘Sqrt’	‘Sqrt’	[‘Auto’, ‘Sqrt’]
criterion	’Entropy’	‘Entropy’	‘Entropy’	‘Entropy’	[‘Entropy’, ‘Gini’]
n_estimators	-	100	-	200	[10-300]

Various techniques for identifying optimal hyperparameters include Grid Search, Random Search, Bayesian Optimization, Gradient-based Optimization, and Evolutionary Algorithms. Although Grid Search is resource-intensive, we selected it for optimizing our DT and RF models due to its effectiveness and the manageable scope of our search space. Details of the search space and outcomes are documented in Table I.

In this study, we focus on key hyperparameters for the DT and RF models that are crucial for managing model complexity and preventing overfitting, without imposing any constraints. While hardware deployment of these optimized models is inherently limited by resource constraints such as computation and memory, potentially affecting performance, the examination of these limitations is beyond the scope of our current research.

Table I reveals that setting ‘None’ as the `max_depth` for the DT in DL (DT-DL) indicates the necessity of deeper trees to model the complex relationships within the data. The smaller values for `min_samples_leaf` and `min_samples_split` suggest the dataset is sufficiently large to support detailed segmentation without risking overfitting. Variations in these parameters between UL and DL data may reflect differing noise levels or variability, necessitating tailored tree growth criteria. The choice of ‘sqrt’ for `max_features` across models implies a broad feature set, where feature reduction aids in avoiding overfitting and enhancing computational efficiency. The distinct approaches for UL and DL models indicate unique data characteristics that require bespoke modeling strategies. To mitigate overfitting and maintain uniform training quality, our research employs 10-fold cross-validation for the expedient DTs, enabling extensive hyperparameter tuning, while opting for 5-fold cross-validation with RFs, balancing depth of evaluation against their higher computational demands. The effective parameters are the important part of hyperparameters for classification which are discussed in the next subsection.

B. Features effectiveness analysis

To examine the model, we use TP and TN metrics. These metrics require labeled data to measure. TP measures that the network traffic which was classified into AR class by the model belongs to AR. On the other hand, TN measures that the network traffic that was classified as non-AR does not belong to AR. This examination is done for different combinations of features while considering three classes (AR, CG, and others) separately. The results are shown in Fig 2. Analysis from Fig. 2a and Fig. 2b suggests that the accuracy of DL and

[3] The proposed models are intended to be implemented and deployed in hardware, e.g., Tofino switches or SmartNICs in the near future.

UL traffic classification using DT heavily relies on the chosen features. Specifically, 'IPI', 'FS', and 'IFI' together yield high TP rates for AR traffic in DL, underscoring their effectiveness. Yet, this combination doesn't equally improve TN rates for other classes, hinting at a potential trade-off between class-specific detection and overall accuracy. Simpler feature sets, especially those including 'FS', provide more balanced TP and TN rates, indicating better generalization.

For UL classification, 'FS' is pivotal for high TP rates in both AR and CG traffic, but it does not guarantee high TN rates, emphasizing the importance of using a diverse feature set for accurate 'Others' class identification. The 'IPI', 'FS', and 'IFI' combo is recommended for balanced TP and TN rates in UL by DT. RF model in DL, as shown in Fig. 2c, achieves consistent performance across classes with 'FS' and 'IFI', effectively identifying AR traffic while maintaining solid TN rates for non-CG and 'Others'.

In UL classification (Fig. 2d), RF models demonstrate that 'FS' is crucial for high TP rates in CG traffic, with a mix of 'FS', 'IFI', and 'IPI' optimizing classification of the 'Others' category, indicating that a multifaceted feature approach boosts classification efficacy. Overall, 'FS' emerges as a key factor in classifying AR, CG, and 'Others' in both DT and RF models. However, its efficacy is maximized when combined with 'IFI' and 'IPI', offering the best balance of TP and TN rates across UL and DL classifications, despite variable trends observed.

VI. AR & CG Network Traffic Dataset

In this section, we detail the collection process for our AR and CG network traffic dataset, which is available in PCAP file format. Towards research reproducibility, the dataset was captured using Tshark as illustrated in Fig. 3.

A. AR dataset

To effectively train, test, and enhance the ML model for AR application traffic classification, the dataset derived from AR glasses traffic is crucial due to varying resolutions and refresh rates among different AR glasses. Given the high cost of AR glasses and considering that frame size and refresh frequency are key features, we propose generating traffic based on these attributes. This approach simplifies the process and initiates a comprehensive AR network traffic dataset, contributing to research in this field. For traffic generation and collection, we are inspired by ITU-T standard for AR [4] and utilized two wireless connected computers through an Access Point (AP), as depicted in Fig. 3a, with hardware specifications detailed in Table II.

We leverage seven scene video frames datasets from Microsoft[5] and the FFmpeg tool to create videos at specific resolutions and refresh rates, as outlined in Table III, simulating the perspective of a person wearing AR glasses. In this scenario, the glasses capture scenes, encoding frames in H.264 and transmitting them to an edge server, which processes and returns the video at compatible resolutions for the glasses. The focus here is on analyzing AR network traffic; thus, the latency and processing overhead at the edge server (e.g. rendering) are not considered. We design separate UL and DL streams to execute this scenario, detailed in Table III, reflecting the glasses' specifications. Video streams are created and encoded in H.264 using Gstreamer tools, based on the profiles specified in Table III, and transmitted over Wi-Fi using RTP. On the receiving end, GStreamer[6] processes the stream without saving or displaying it to minimize overhead. Network traffic is captured using Tshark and saved as PCAPs.

TABLE II: Specifications of devices

Device	Features	Model
PC1	CPU	Intel(R) Core$^{\text{TM}}$ i7-5500U
	RAM	12 GB
	Storage	1 TB
	Wi-Fi	Wi-Fi 4E (802.11n)
	OS	Microsoft Windows 10 Pro
PC2	CPU	Intel Core$^{\text{TM}}$ i7-13700T
	RAM	16 GB
	Storage	500 GB
	Wi-Fi	Wi-Fi 6E (802.11ax)
	OS	Linux Ubuntu 22.04.3 LTS
AP	Standard	802.11a/b/g
	Speed	600 Mbps - 2400 Mbps (2.4GHz-5GHz)

TABLE III: Streams Information for Generating AR Dataset

DL/UL	Name	Resolution	fps
UL	Stream1[21]	1280 x 480	60
	Stream2 [4, 5, 8]	1920 x 1080	90
	Stream3[6]	1440 x 936	60
	Stream4[7]	2064 x 2208	120
	Stram 5[7]	1832 x 1920	120
DL	Stream6 [21]	3840 x 1920	72 or 90

The experiment was run two times for each stream and each run took 10 minutes, totaling a 120-minute long collected AR network traffic dataset in PCAP format as publicly available.

B. Cloud Gaming

The CG dataset from [19] was expanded by incorporating data collection from various games on Xbox Cloud Gaming, including Forza Horizon 5, Fortnite, and Mortal Kombat 11, each played for 15 minutes and repeated twice. The setup, depicted in Fig 3b, involves a PC (Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 4GB RAM) for gaming and a Raspberry Pi wired to the Internet and acting as the Access Point (AP) for the PC wireless connectivity, where traffic is captured with Tshark, resulting in a total of 90 minutes of gameplay PCAP data[7].

VII. Model Evaluation

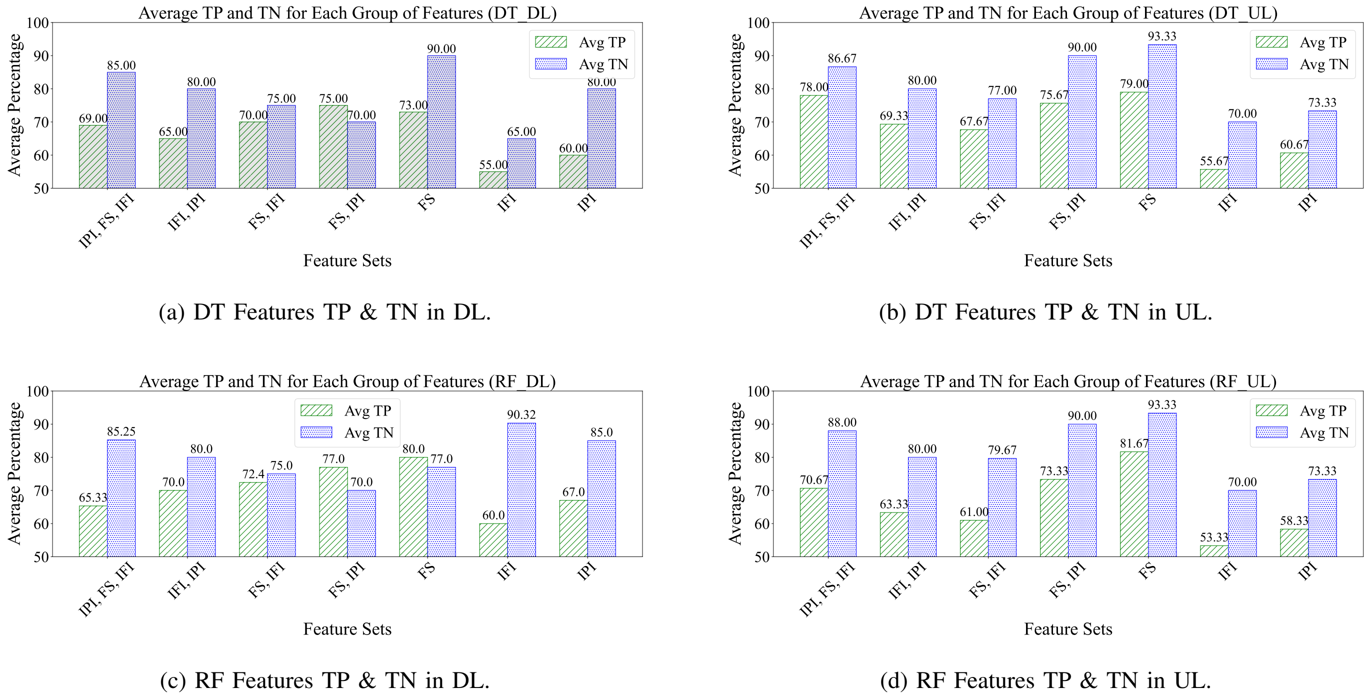
The proposed DT and RF classification models' performance is evaluated using accuracy, precision, recall, and f1-score

[4] https://www.itu.int/ITU-T/recommendations/rec.aspx?id=14419

[5] https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/

[6] https://gstreamer.freedesktop.org/

[7] We are also collecting In-band Network Telemetry (INT) such as queue occupancy in this device. This data will be made available soon as well.



(a) DT Features TP & TN in DL.

(b) DT Features TP & TN in UL.

(c) RF Features TP & TN in DL.

(d) RF Features TP & TN in UL.

Fig. 2: TP & TN of DT & RF algorithms for different feature combinations.

TABLE IV: Pre-existing Dataset (DS) for Testing the Models

ID	Name	App	Ref.	Files	DL/UL
1	XR traffic	AR	[21]	Model	UL,DL
2	AR traffic	AR	[15]	CSV	DL
3	VR DS	AR	[26, 27]	PCAP, CSV	DL
4	CG DS	CG	[19]	PCAP	UL,DL
5	Non-AR/Non-CG	Other	[19]	PCAP	UL,DL

TABLE V: DT & RF Model Performance

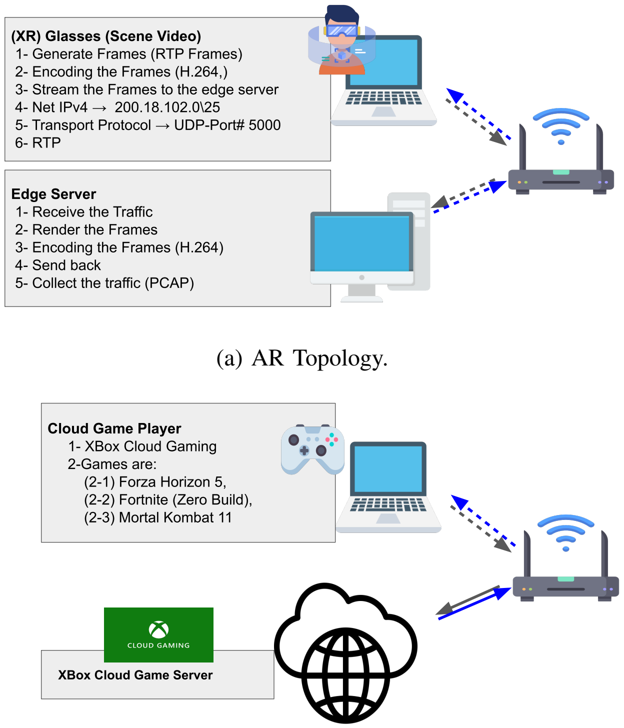
Model	Datasets	Accuracy	Precision	Recall	F1-score
DT-UL	Training Eval	96.4	96.6	96.4	96.4
	Pre-existing DS	94.8	95.5	94.87	94.84
	Collected DS	95.27	96.21	95.27	95.42
DT-DL	Training Eval	95.00	95.1	95.00	95.00
	Pre-existing DS	90.74	89.37	90.74	89.58
	Collected DS	94.87	94.87	94.87	94.87
RF-UL	Training Eval	95.4	95.6	95.4	95.4
	Pre-existing DS	94.87	94.87	94.87	94.87
	Collected DS	94.75	94.09	94.75	94.81
RF-DL	Training Eval	95.1	95.5	95.1	95.00
	Pre-existing DS	88.40	90.78	88.40	87.93
	Collected DS	91.14	89.24	91.14	89.68

metrics. To test the model, we use the dataset collected from pre-existing datasets as mentioned in Table IV whose samples are unseen for the dataset we trained our model. Furthermore, we test the model with the dataset created in this research (Section VI) for AR and CG as well.

The datasets are merged and the samples are shuffled to create a comprehensive test dataset for model evaluation. This evaluation is conducted in three phases: (a) assessing the model's performance during the training phase which used 10% of the dataset, (b) evaluating the model using pre-existing datasets as mentioned in Table IV, and (c) testing the model against datasets specifically collected in this research at our lab. The outcomes of these evaluations are detailed in Table V.

Table V reveals the DT-UL model's high efficiency, achieving top accuracy (96.4%), precision (96.6%), and F1-score (96.4%) in the Training Evaluation dataset. This highlights DT-UL's strong performance in classifying training data and maintaining effectiveness across both existing and new datasets. In contrast, the Decision Tree Downlink (DT-DL) model, while slightly less effective than DT-UL, delivers notable results, particularly with training and using our lab datasets. However, it experiences a drop in precision (89.37%) and F1-score (89.58%) with the existing dataset, indicating potential difficulties in consistent prediction across diverse datasets. The Random Forest Uplink (RF-UL) model demonstrates consistent and strong performance, closely matching DT-UL in all tests, suggesting its dependability across different datasets. The Random Forest Downlink (RF-DL) model, however, shows reduced performance in the pre-existing dataset, with lower accuracy (88.40%), precision (90.78%), and F1-score (87.93%), hinting at possible overfitting or reduced generalizability.

The classification performance results suggest that the observed equivalence between accuracy and recall can be attributed to the weighted multi-class approach, which effectively normalizes class imbalance. The evaluation suggests that DTs outperform due to their fit with specific dataset traits, though they're sensitive to data variation, while RFs' ensemble na-


(XR) Glasses (Scene Video)
1- Generate Frames (RTP Frames)
2- Encoding the Frames (H.264,)
3- Stream the Frames to the edge server
4- Net IPv4 → 200.18.102.0\25
5- Transport Protocol → UDP-Port# 5000
6- RTP
Edge Server
1- Receive the Traffic
2- Render the Frames
3- Encoding the Frames (H.264)
4- Send back
5- Collect the traffic (PCAP)
Cloud Game Player
1- XBox Cloud Gaming
2-Games are:
(2-1) Forza Horizon 5,
(2-2) Fortnite (Zero Build),
(2-3) Mortal Kombat 11
CLOUD GAMING
XBox Cloud Game Server

(a) AR Topology.

(b) CG Topology.

Fig. 3: Network topology for AR & CG traffic collection where dash lines indicate wireless and solid lines wired connectivity.

ture provides stability but demands careful tuning to avoid overfitting. This highlights the necessity of meticulous model calibration to the dataset's nuances, enhancing classification precision, as demonstrated by the slight superiority of DTs, likely owing to optimal tuning for the dataset's characteristics.

VIII. Discussion

AR traffic dataset scarcity. In this study, we addressed the AR traffic dataset scarcity by leveraging a statistical model [21] to create an AR training dataset for our DT and RF models. We enhanced our dataset by incorporating real VR network traffic from [25], exploiting the similarities between AR and VR in DL scenarios as indicated in [13, 20], and further evaluated our models' DL performance using datasets from [26, 27]. Our comprehensive testing across both existing and newly developed datasets underscored our models' capability to accurately classify emerging network traffic, marking a pivotal step forward in AR network traffic classification research.

Performance of DT and RF. In the experiment results, the DT and RF models demonstrated a notable decrease in classification performance when applied to pre-existing datasets. Upon further investigation into the limitations of these models, it was discovered that DT and RF achieved a classification accuracy of 95.36% and 96.35% respectively on instances from these datasets. A critical observation was made regarding the AR dataset collected in a local host environment, as mentioned in [24], which did not account for network conditions. Consequently, this oversight led to inaccuracies in classification using the 'IPI' feature, impacting the expected performance.

Overfitting. Cross-validation and hyperparameter tuning were employed to mitigate overfitting, a potential issue given the limited size of the datasets. Additionally, a combination of statistical modeling and real data collection was utilized to enhance the models' generalizability. RF models, especially in the UL (RF-UL) scenario, demonstrated superior stability across various datasets compared to DT models, likely due to RF's inherent ability to reduce overfitting by averaging predictions from multiple DTs. The evaluation of models using unseen datasets revealed high accuracy and f1-scores, suggesting successful generalization beyond mere training data memorization.

Accuracy and Reliability. Confidence intervals for DT and RF models in classifying AR, CG, and other network traffic types affirm their accuracy and reliability across both UL and DL directions. Specifically, DT models achieve a 95% confidence interval ranging from 95.8% to 99.2%, while RF models show a slightly higher range from 96.5% to 99.4%. Such tight confidence intervals highlight the models' robustness and precision in classifying diverse datasets. Evaluation using both pre-existing and newly collected datasets validates the models' anticipated performance, positioning them as effective tools for network traffic classification.

IX. Conclusions and Future Work

In this paper, we introduced DT and RF models to classify network traffic into AR, CG, and other application categories. These models were evaluated using both pre-existing and newly collected datasets to validate their efficacy in distinguishing AR and CG traffic. However, the potential for bias and overfitting remains a concern due to the limited diversity of the training datasets, highlighting the need for a more extensive collection of AR and CG data to enhance model generalization. To address this, we have compiled and made a comprehensive AR and CG traffic dataset available.

While the models developed so far demonstrated high accuracy in traffic classification, deploying them in real-world scenarios may present challenges related to hardware constraints. Our next steps involve deploying the models on programmable network equipment including Tofino switches, SmartNICs, and DPUs. We will also increase the size of our data sets and make them available along with INT metrics (e.g., queue occupancy, queue delay, and others) collected from a 5G setup and experiment with real-world traffic scenarios for varying AR glasses and applications. We suspect that the differences in terms of flow patterns may vary significantly depending on the AR manufacturer as well as the specific AR application and user behavior. We are also calling for a joint effort from the community to make publicly available rich datasets for VR/AR applications for a better understanding of commercial applications under real networking conditions to support impact research in terms of novel classification models, QoE estimation as well as orchestration loops to improve QoE/QoS.

Acknowledgment

This work was supported by the R&D and Innovation Center, Ericsson Ltda, and by the São Paulo Research Foundation (FAPESP), grant 2021/00199-8, CPE SMARTNESS.

References

[1] M. Speicher, B. D. Hall, and M. Nebeling. "What is mixed reality?" In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. May 2019, pp. 1–15.

[2] A. Villegas, P. Pérez, and E. González-Sosa. "Towards a distributed reality: a multi-video approach to xr". In: *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems*. June 2019, pp. 34–36.

[3] T. S. Perry. "Look Out for Apple's AR Glasses: With head-up displays, cameras, inertial sensors, and lidar on board, Apple's augmented-reality glasses could redefine wearables". In: *IEEE Spectrum* 58.1 (2020), pp. 26–54.

[4] Xreal Air 2 Pro, [Online]. Available: https://www.xreal.com/air2, [Accessed: 2024-01-12].

[5] Rokid Max Pro, [Online]. Available: https://ar.rokid.com, [Accessed: 2024-01-12].

[6] Hololens2, [Online]. Available: http://www.microsoft.com/en-us/hololens/hardware#document-experiences, [Accessed: 2024-01-12].

[7] *Meta Quest 2 & 3*. https://www.meta.com. Accessed: 2024-01-12.

[8] XVarjo XR-3, [Online]. Available: https://varjo.com/products/varjo-xr-3, [Accessed: 2024-01-12].

[9] D. G. Morín, P. Pérez, and A. G. Armada. "Toward the distributed implementation of immersive augmented reality architectures on 5G networks". In: *IEEE Communications Magazine* 60.2 (2022), pp. 46–52.

[10] Ericsson Technology Review, [Online]. Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/future-network-requirements-for-xr-apps, [Accessed: 2024-01-12].

[11] Nokia, [Online]. Available: https://www.nokia.com/blog/5g-advanced-will-power-mobile-xr-experiences-virtually-anywhere, [Accessed: 2024-01-12].

[12] *5G architecture support for XR and media services*. https://www.3gpp.org/technologies/xr-sa2. Accessed: 2024-01-12.

[13] *3GPP Specifications and Technologies, Spec No:38.838*. https://portal.3gpp.org. Accessed: 2024-01-12.

[14] https://www.pewresearch.org/internet/2022/06/30/the-metaverse-in-2040. Accessed: 2024-01-12.

[15] P. Schulz et al. "Analysis and Modeling of Downlink Traffic in Cloud-Rendering Architectures for Augmented Reality". In: *2021 IEEE 4th 5G World Forum (5GWF)*. Oct. 2021, pp. 188–193.

[16] Ahmad Azab et al. "Network traffic classification: Techniques, datasets, and challenges". In: *Digital Communications and Networks* (2022). ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2022.09.009.

[17] Qianqian Wu et al. "P4SQA: A P4 Switch-based QoS Assurance Mechanism for SDN". In: *IEEE Transactions on Network and Service Management* (2023).

[18] N. Shah. *The challenges of inspecting encrypted network traffic*. Fortinet [Internet]. Accessed: 2024-01-12. 2020. URL: https://www.fortinet.com/blog/industry-trends/keeping-up-with-performance-demands-of-encrypted-web-traffic.

[19] P. Graff et al. "Efficient Identification of Cloud Gaming Traffic at the Edge". In: *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. May 2023, pp. 1–10.

[20] A. Alnajim et al. *Traffic Characteristics of Extended Reality*. arXiv preprint arXiv:2304.07908. 2023.

[21] D. Gonzalez Morin et al. *An eXtended Reality Offloading IP Traffic Dataset and Models*. arXiv e-prints, arXiv-2301. 2023.

[22] T. Shapira and Y. Shavitt. "FlowPic: A generic representation for encrypted traffic classification and applications identification". In: *IEEE Transactions on Network and Service Management* 18.2 (2021), pp. 1218–1232.

[23] Aristide Tanyi-Jong Akem, Guillaume Fraysse, Marco Fiore, et al. "Encrypted Traffic Classification at Line Rate in Programmable Switches with Machine Learning". In: *IEEE/IFIP Network Operations and Management Symposium*. 2024.

[24] Andreas Traßl, Nick Schwarzenberg, and Philipp Schulz. *Augmented Reality Streams for Cloud-Based Rendering*. IEEE Dataport. 2021. URL: https://dx.doi.org/10.21227/jjan-tj96.

[25] Seyedmohammad Salehi. *Motivation: Video rendering on the Oculus Quest vs. Edge server*. https://www.eecis.udel.edu/~salehi/vr.html. Accessed: 2024-01-12.

[26] M. Polupanova. "VR Traffic Dataset on Broad Range of End-User Activities". In: *Data* 8.8 (2023), p. 132. DOI: 10.3390/data8080132. URL: https://doi.org/10.3390/data8080132.

[27] Sihao Zhao et al. "Virtual Reality Gaming on the Cloud: A Reality Check". In: *IEEE Global Communications Conference (GLOBECOM 2021)*. Madrid, Spain, 2021.

[28] Shie-Yuan Wang and Ying-Hua Wu. "Supporting Large Random Forests in the Pipelines of a Hardware Switch to Classify Packets at 100 Gbps Line Rate". In: *IEEE Access* (2023).

[29] P. E. Hart, D. G. Stork, and R. O. Duda. *Pattern classification*. Hoboken: Wiley, 2000.