applied sciences

MDPI

Article

A Novel Proprietary Internet Video Traffic Dataset Generation Algorithm

Tianhua Chen [1,*], Elans Grabs [1], Aleksandrs Ipatovs [1] and Maria-Dolores Cano [2,*]

[1] Institute of Photonics, Electronics and Telecommunications, Riga Technical University, LV-1048 Riga, Latvia; elans.grabs@rtu.lv (E.G.); aleksandrs.ipatovs@rtu.lv (A.I.)
[2] Department of Information Technologies and Communication, Universidad Politécnica de Cartagena, Plaza del Hospital 1, 30202 Cartagena, Spain
* Correspondence: tianhua.chen@rtu.lv (T.C.); mdolores.cano@upct.es (M.-D.C.)

Abstract: Considering the exponential growth of network traffic, particularly driven by over-the-top (OTT) streaming applications, video category network traffic constitutes a significant portion of overall network traffic. However, most research has focused on the categorization and diversity of network traffic using benchmark datasets, with limited attention paid to video category network traffic. Additionally, there is a lack of proprietary Internet video traffic datasets, and the few proprietary datasets available often lack transparency and interpretability. This paper introduces a novel framework for generating proprietary Internet video traffic datasets, addressing existing gaps in dataset quality and consistency. We propose the nYFTQC algorithm, which enables the creation of fifteen detailed datasets specifically designed for Internet video traffic analysis. The proposed datasets demonstrate superior performance metrics, including completeness, consistency, and transparency. This comprehensive approach enhances the accuracy and interpretability of traffic sample analysis, providing valuable resources for future research in video category network traffic.

Keywords: network traffic classification; proprietary dataset; algorithm; interpretability

check for updates

Academic Editor: Andrea Prati

Received: 3 December 2024
Revised: 3 January 2025
Accepted: 5 January 2025
Published: 7 January 2025

Citation: Chen, T.; Grabs, E.; Ipatovs, A.; Cano, M.-D. A Novel Proprietary Internet Video Traffic Dataset Generation Algorithm. *Appl. Sci.* **2025**, *15*, 515. https://doi.org/10.3390/app15020515

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The volume of network traffic is increasing exponentially, largely driven by the growing use of over-the-top (OTT) media services [1]. These video applications generate significant network traffic characterized by large packet volumes, vast data bytes, and irregular packet arrival intervals. As a result, video traffic has become a critical focus of analysis in Network Traffic Monitoring and Classification (NTMC). The benchmark Internet traffic flow dataset BRASIL: Characterizing Network-based Applications was introduced in 2005 and classified Windows Media Player and Real applications under the multimedia category [2]. Similarly, popular benchmark datasets such as ISCXVPN2016, ISCXTor2016, and CIC-Darknet2020 categorize video traffic under labels like Streaming and Video-Stream [3–5]. However, categories such as Web Browsing, Chat, Transfer, Audio-Stream, and VoIP may include traffic flows related to video, but the boundaries between applications and categories remain unclear, requiring further validation for accurate classification. In modern networks, video traffic resists straightforward categorization, it is intertwined with other traffic categories in diverse characteristics, including network protocols, architectures, application types, and traffic distribution characteristics, creating a complex profile. While some datasets include video-related traffic, these samples typically represent only a small portion of the total dataset. To address this, researchers have developed proprietary datasets

tailored for specific traffic categories. This paper reviews existing network traffic datasets that contain video traffic and introduces a novel algorithm to generate a proprietary Internet video traffic dataset. The proposed approach aims to improve the performance of classification metrics in the video traffic category within NTMC.

2. Related Works

In this section, we will comprehensively compare the characteristics of video-type traffic across various public and proprietary network traffic datasets, including category and application selection, transmission protocol, sample size ratio, etc. The BRASIL dataset contains a total of 14 categories, with over one million traffic flow records; however, multimedia traffic accounts for less than 0.1% [2]. The ISCXVPN2016 and ISCXTor2016 datasets categorize traffic into 14 groups based on VPN and Tor characteristics, respectively. Each dataset includes seven main categories, with the video category featuring streaming applications such as Vimeo, Facebook, and YouTube. Both ISCXVPN2016 and ISCXTor2016 contain over 150,000 flow samples, with video-streaming traffic accounting for less than 5% and 2%, respectively [3,4]. The CIC-Darknet2020 dataset merges the VPN and Tor traffic from the ISCXVPN2016 and ISCXTor2016 datasets into a Darknet category, with video-streaming traffic still accounting for less than 5% of over 150,000 flow samples [5]. The traffic categories in these datasets are often selected based on specific tasks in the NTMC, such as intrusion detection, botnet detection, encrypted traffic analysis, and anonymized network traffic studies.

Bader et al. developed the OSF-EIMTC traffic classification framework, which demonstrated excellent performance and was validated on several public and benchmark datasets, including USTC-TFC2016, ISCXVPN2016, Ariel (BOA2016), MAppGraph2021, MTA (Malware Traffic Analysis), and StratosphereIPS [6]. The USTC-TFC2016 dataset includes 10 categories of malware traffic and 10 categories of benign traffic, with the video-related category containing applications such as FaceTime and Skype. This dataset includes over 75,000 flow and session samples [7]. The StratosphereIPS dataset includes only three categories, benign, malware, and mixed, without application or protocol metadata. The MTA dataset contains two categories, malware and legitimate traffic, both lacking application and protocol metadata. The Ariel dataset contains over 20,000 session samples across 30 categories, with video traffic from YouTube and Facebook accounting for less than 10% of the total dataset [8]. The MAppGraph dataset provides over one million flow records spanning 101 mobile application categories. Its video traffic category includes applications such as Netflix, TikTok, and Twitch, which contribute less than 10% of the total dataset [9]. Liu et al. proposed the Multi-Task Learning Fusion (MTEFU) algorithm, achieving 94.67% classification accuracy on the QUIC dataset, which includes five service categories: Google Docs, Google Drive, Google Music, YouTube, and Google Search. The dataset comprises nearly 7000 flow records [10]. Existing studies show that video traffic samples make up only a small portion of most public and benchmark datasets. Key issues include limited sample sizes, narrowly defined categories, and a lack of metadata descriptions. Additionally, these datasets often employ inconsistent naming conventions for applications, categories, SSL server names, Server Name Indication (SNI), and other elements, complicating the validation of relationships between samples and their naming conventions.

Some proprietary Internet traffic datasets focus on the video traffic category. Salman et al. utilized packet-level and flow-level features such as size, inter-arrival time, and packet flow direction, combined with ensemble deep learning classifiers, achieving 89.77% classification accuracy across four traffic categories: Interactive, Bulk Data Transfer, Streaming, and Transaction. Their dataset included 3612 flow records [11]. Arestrom et al. categorized video traffic into six groups based on Quality of Service (QoS) and Quality of Experience

(QoE) characteristics: video streaming, web browsing, social networking, audio communication, text communication, and bulk download. Their dataset included 7154 session samples [12]. Wu et al. leveraged byte and time sequence features to classify traffic from 20 Google and Apple applications and 58 web services, achieving over 98% accuracy using the TLS Flow Sequence Network (TFSN) model on a dataset of 13,915 TLS flow samples [13]. Xiao et al. proposed the Extended Byte Segment Neural Network (EBSNN) with long short-term memory (LSTM) for traffic classification across 20 websites, achieving 99.96% accuracy. They collected two kinds of datasets, application identification and website identification with 29 and 20 categories, respectively, and also chose a total of 11,607 flow records from 8 application categories for validation [14]. Gabilondo et al. defined five application traffic categories, Best-Effort/Default, Control Data, Video, IoT, and WebData, analyzing bandwidth requirements and QoS performance across 5G network slice scenarios based on service types and QoS-related characteristics [15].

Wang et al. categorized video traffic into five groups based on QoS levels: broadcast video (BV), web video (WV), trade-style video (TSV), barter-style video (BSV), and interactive video (IV). Each category contained 100 flow samples, sourced from platforms such as Tudou, Youku, TVAnt, Skype, and others. Using a modified K-Singular Value Decomposition (KSVD) classification framework, they achieved 98.97% classification accuracy [16]. Dong et al. employed a hierarchical k-Nearest Neighbor (k-NN) algorithm and achieved over 96.77% classification accuracy on their Internet video traffic dataset. This dataset included six categories: asymmetric standard-definition videos (ASD), asymmetric high-definition videos (AHD), HTTP-download videos (HDV), interactive videos (CV), P2P_video, and network live TV (ILV). It comprised 360 flow samples with a total data volume of 13.03 GB [17]. Canovas et al. developed a multimedia traffic classification system based on objective QoE, collecting 2741 video traffic flow samples labeled into five categories: Non-critical, Low-critical, Some-critical, Critical, and Very-critical [18]. Hayashi et al. collected 400 peer-to-peer video streaming (P2PTV) traffic samples and classified them into server-based bursty traffic, stable traffic, and peer-based bursty traffic [19]. Arestrom et al. achieved an F1-score of 0.893 in classifying video traffic across two categories, Video-on-Demand (VoD) and live streaming, using a dataset of 1232 flow samples [12]. Costa Da Silva et al. defined the video category as LowVideo (Lv), AverageVideo (Av), and High-Video (Hv) based on the chance of detection of a potential characterization trained in flows fuzzy classifiers achieved as high as 90% classification accuracy; however, the datasets used in their study were not adequately described [20]. The above works have developed proprietary Internet video traffic datasets and provided in-depth analyses of video traffic. However, several issues persist. Video traffic often originates from different web platforms and applications or relies on privately emulated video traffic. This makes the classification of Internet video traffic into categories more subjective, as it depends on varying video traffic behaviors or attribute information.

Wu et al. enhanced Internet video traffic classification by introducing the SV category, further subdivided into Standard Definition (480P, SD), High Definition (720P, HD), and Ultra-clear Definition (1080P, UD), achieving 95% accuracy across seven categories using a Chain Hierarchical Structure (CHS) scheme. The dataset included 25,200 flow records [21]. In another study, Wu et al. classified streaming video traffic based on raw video resolution, including audio, 144P, 240P, 360P, 480P, and 720P, into six categories. Using transport layer-encapsulated chunk data unit sizes as features with a Convolutional neural network (CNN) model, they achieved 99% accuracy for each category and a minimum F1-score of 95% on a dataset of 11,510 flow samples [22]. Bukhari et al. developed a proprietary YouTube video dataset containing 5000 streams from 100 categories across different channels, achieving 90% classification accuracy by combining the CNN model with

Packets-Per-Second (PPS) features [23]. Ozkan et al. evaluated video traffic classification across seven popular mobile applications: Netflix, YouTube, YouTube Live, Twitch, Spotify, WhatsApp, and Skype, using statistical models such as Mixture of Markov Components (MMC), k-Nearest Markov Component (kNMC), and k-Nearest Markov Parameter (kNMP). Their dataset, approximately 6GB in size, covered four categories: Video on Demand (VoD), Video Live Streaming (VLS), Sharing and Media Streaming (S&MS), and Teleconferencing (TC) [24].

In general, whether using public or proprietary Internet traffic datasets, video traffic is typically treated as a distinct category, though the labeling definitions for this category vary significantly. Researchers consider several factors when defining labels. First, they consider a priori knowledge about the video, including the web platform, application, content labels, source encoding, resolution, and refresh rate. Second is metadata from the video transmission process, such as transport protocols (e.g., Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Quick UDP Internet Connections (QUIC)) and streaming technologies (e.g., Dynamic Adaptive Streaming over HTTP (DASH), HTTP Live Streaming (HLS), VoD, and VLS). Additional factors include QoS and QoE metrics, as well as the utilization of Tor, VPNs, and other technologies. These elements impose higher demands for achieving fine-grained network video traffic classification. However, most existing proprietary Internet video traffic datasets still have some limitations, including non-disclosure of the dataset, small sample sizes, weak correlations between samples and the aforementioned video-related factors, and limited interpretability. To address these gaps, we develop three algorithms for generating proprietary Internet video traffic datasets.

The main contributions of this paper can be summarized as follows:

- A novel and explainable proprietary Internet video traffic dataset generation algorithm is proposed, which generates fifteen corresponding datasets.
- The resulting datasets demonstrate excellent performance metrics, including completeness, consistency, and transparency.

The rest of this paper is organized as follows: Section 3 outlines the research methodology, covering the NFStream preprocessing algorithm, the proprietary Internet video traffic dataset generation algorithm, and the CICFlowmeter preprocessing algorithm. Section 4 provides an analysis of the proprietary Internet video traffic datasets, focusing on imbalance ratios, video traffic histogram density, and feature correlation. Section 5 concludes the paper.

3. Methodology

Our proposed proprietary Internet video traffic dataset generation framework is illustrated in Figure 1. First, this approach involves streaming a single web video on a specific platform for 30 min using Google Chrome (version 100.0.4896.88). During this process, wireless traffic from an Intel(R) Wi-Fi 6 AX200 160MHz network adapter is captured using Wireshark (version 3.4.3) under a Windows 10 (Build 18363.1316) operating system. The resulting Packet Capture (PCAP) files are labeled based on characteristics such as video resolution, frame rate, and VoD, or VLS attributes to facilitate distinction.

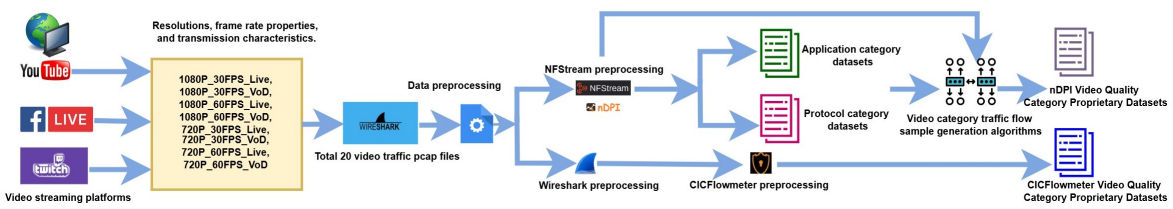


Figure 1. The framework of proprietary Internet video traffic dataset generation.

The next step involves data preprocessing, which includes NFStream preprocessing and Wireshark preprocessing. Sniffed traffic from a wireless adapter interface may include a mixture of traffic from various software or applications, and different analytical tools and methods can further refine the categorization of the dataset. The capabilities of network traffic analytic tools incorporate sniffing and capturing traffic packets and detecting network protocols, which are diffusely applied in cybersecurity monitoring and analysis. Prevalent network protocol and packet capture analysis software include Wireshark and Tcpdump, extensively deployed in network devices and systems at diverse system levels. Another category of traffic analysis tools has the main property of traffic classification. The representative approach is to perform traffic classification based on the results of Deep Packet Inspection (DPI) and generally utilized software tools, including L7-filter, OpenDPI, Libprotoident, nDPI, Zeek, Tstat, and NFStream [25,26]. Highly integrated traffic analytical tools can effectively accelerate the generation of proprietary datasets.

DPI techniques achieve traffic classification by parsing header fields in the application layer of Internet traffic, examining the entire transport layer payload, and extracting keywords from packet signatures or feature strings to identify web applications and protocol types. Typically utilized search and match algorithms include Aho–Corasick (AC) and Knuth–Morris–Pratt (KMP). With the popularity of encrypted traffic, DPI cannot extract the application layer payload information of encrypted traffic. Common solutions to this challenge include the lightweight inspection of limited packet payloads and metadata extraction. Metadata covers information such as bidirectional IP addresses and protocols in TCP/IP packets, and it also includes flow-related information, which, combined with powerful machine learning techniques, tremendously extends the direction of traffic classification. DPI technology is gradually evolving into Deep Flow Inspection (DFI) technology.

CICFlowmeter is a representative DFI tool capable of generating over 76 metadata features, which could be used as a reference for performance comparison [4]. The nDPI tool not only provides detected protocol references but also covers a large number of metadata discriminator conditions to furnish a guarantee for classification results [27]. In this paper, the actual utilization of NFStream [28], a traffic analysis tool based on nDPI and developed in Python language, has the advantage of being convenient to operate and extracting over 86 metadata as training features. We set the flow sample active timeout to 180 s for both tools. This configuration divides a single video traffic-related flow record into multiple records, simplifying further processing operations and increasing the sample volume. Additionally, we compare the differences in flow sample distributions under the same timeout value between the two tools.

In the final step, we integrate the NFStream preprocessing results with video category traffic flow generation algorithms to produce corresponding proprietary Internet video traffic datasets. Details are discussed in the following three subsections.

3.1. NFStream Preprocessing

In the NFStream preprocessing step, the process begins by sending pre-named PCAP files to the NFStream tool for traffic flow metering and exporting to CSV files. Based on the NFStream configuration's accounting mode, statistics are provided across four layers, IP, transport, link, and payload, resulting in over 800,000 flow records. In the second step, the NFStream tool, integrated with the nDPI library, detects the application name and application category for each flow, subsequently splitting them into two separate datasets based on these properties. These datasets require additional noise reduction and cleaning.

Figure 2 illustrates the flow record counts and distributions in the protocol and application category datasets across the four accounting modes. In the application category dataset, the Web category accounts for over 71% of the sample volume, followed by Net-

work, Download, and System. Similarly, in the protocol category dataset, the TLS category dominates with over 72% of the samples, while BitTorrent, DNS, and SSDP constitute the second-largest group. However, only 64% of the parsed flow records are categorized, leaving 36% of samples labeled as Unknown or Unspecified in the Protocol and Application categories, respectively. This occurs due to the 180-s active timeout setting, which truncates metering and may prevent some flows from accumulating enough metadata for accurate classification. Notably, the sample counts and distributions remain consistent across all accounting modes. The nDPI-parsed application name feature encompasses major and minor protocols, with minor protocols typically referring to specific software application names, such as Amazon, Google, Microsoft, etc. This study focuses on major protocols and excludes specific application software or serviced traffic flow samples.

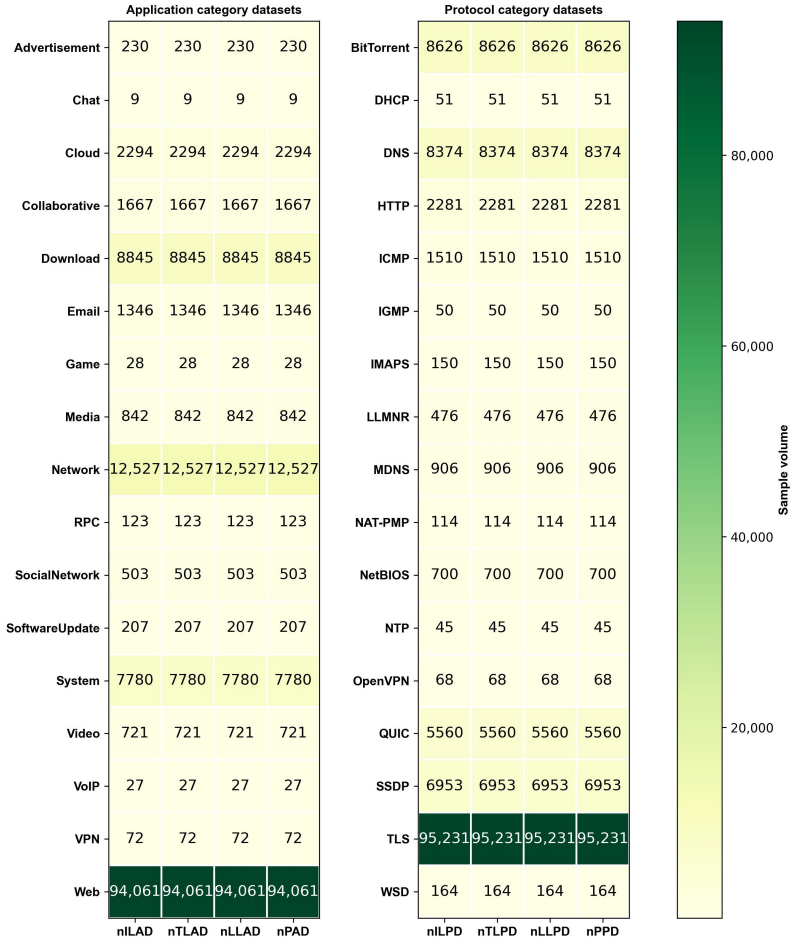


Figure 2. Protocol and application category datasets sample distribution.

Algorithm 1 outlines the nDPI-Detected Application Category and Protocol Category (nDACPC) dataset generation algorithm. A key step in this algorithm involves removing all flow samples where the detected application category is unspecified, as these flows cannot be assigned to a class in the given scenario. The next step counts the flow samples for each class and eliminates any class with fewer than six samples, as such classes cannot undergo further resampling operations.

Although both application category datasets and protocol category datasets detail the distribution of each class and the number of parsed flow samples, they are closely related. For instance, an application named TLS.Microsoft is generally classified under the Cloud category, while TLS.Microsoft365 is placed in the Collaborative category. Since Microsoft365 is a suite of Microsoft Office applications, it is understandable, upon human inspection,

why it falls under the Collaborative class. However, when the major protocol is the same, the subtle differences between minor protocols can make class determination challenging.

Algorithm 1 Pseudo-code for nDPI-Detected Application Category datasets and Protocol Category dataset generation (nDACPC)

```
Input: Each flow sample from segmented CSV files
Output: nILAD, nTLAD, nPAD, nLLAD, nILPD, nTLPD, nLLPD, nPPD dataset
for each CSV file in segmented CSV files do
    for each flow sample in CSV file do
        if Flow sample application name = "Unknown" then
            Delete the flow sample
        end if
        if Flow sample application category = "Unspecified" then
            Delete the flow sample
        end if
    end for
    Save all remaining samples
    for each flow sample in remaining samples do
        Read each sample application name major protocol
        Save read results into a new column 'protocol category'
    end for
    Count flow samples N for each protocol category class
    if N ≥ 6 then
        Keep all flow samples for the specific class
    else
        Delete all flow samples for the specific class
    end if
    Keep all remaining flow samples in the CSV file
end for
Merge different CSV files' samples into each dataset
Output nILAD, nTLAD, nPAD, nLLAD, nILPD, nTLPD, nLLPD, nPPD dataset
```

For example, in the case of DNS.Microsoft, application categories are divided into four main classes, Web, Cloud, System, and ConnCheck, based primarily on the request server names decoded by the nDPI library. One instance is the domain name www.msftconnecttest.com, used by Microsoft systems to check network connectivity, while the domain for Microsoft diagnostic data management services is events.data.microsoft.com. Similarly, the Microsoft account login domain includes login.live.com, and the Microsoft search engine domain is www.bing.com. Each application category can contain multiple domain names due to the diverse nature of the decoded domain names. Despite this variability, the application category can often be determined based on keywords within the domain name. If no domain names are decoded in the flow, the application category is determined by the minor protocol or a single protocol. The major protocol, minor protocol, and application category serve as ground-truth features used as labels for supervised training and interact with and reinforce each other. From this information, it is apparent that some service application traffic samples are mixed with video traffic samples, and there are still low-quality, irrelevant samples in the dataset. By integrating the characteristics of streamed videos and other prior knowledge, the existing proprietary datasets can be further improved and refined.

3.2. Proprietary Internet Video Traffic Dataset Generation

To generate proprietary video quality category datasets for YouTube, Facebook, and Twitch, we incorporated prior knowledge from PCAP files, including attributes such as video streaming platforms, resolution, and video playback modes. This knowledge was used to develop the

YouTube, Twitch, and Facebook video quality category datasets. Each dataset was further divided into four subsets based on four different nDPI accounting modes. Each dataset contains eight categories: 720P_30FPS_VoD, 720P_30FPS_Live, 720P_60FPS_VoD, 720P_60FPS_Live, 1080P_30FPS_Live, 1080P_30FPS_VoD, 1080P_60FPS_VoD, and 1080P_60FPS_Live. Due to limitations in capturing 60 frames per second on the Facebook platform, no packets for relevant 60 FPS videos were collected, resulting in only four 30 FPS classes being included for this platform.

The generation of nDPI-detected YouTube, Facebook, and Twitch video quality category datasets (nYFTQC) is outlined in Algorithm 2. The following are detailed steps.

Initial Filtering: The first step involves retaining all flow samples associated with the transport layer QUIC and TLS protocols, which primarily handle video traffic packets. For instance, the Google application YouTube interacts with Google's Chrome browser and servers using the QUIC protocol. Additionally, network video traffic is typically encrypted, with the TLS protocol widely used by users and video providers to ensure secure and reliable web communication.

Handling Unknown Categories: In the second step, flow samples from unknown application categories that the nDPI library cannot parse should be retained if they display certain characteristics, such as a high volume of bidirectional packets, large data sizes, and long durations of bidirectional flows. This retention specifically includes flow samples related to Google and Facebook.

Removing Irrelevant Flow Samples: Flow samples irrelevant to video content are deleted based on the requested server name. These include samples related to automatic updates, background processes, advertising statistics, account interfaces, and other non-video-related activities. Filtering is performed using the TLS application name and Web application category name.

Filtering Specific Protocols: The fourth step involves filtering out minor protocols associated with non-video traffic, such as those related to Microsoft, Skype, and similar applications, as indicated in the application name column. Using the elimination method, we delete flow records if the application category name is not from the following list: AmazonAWS, Amazon-Video, Azure, Twitch, Facebook, and YouTube. Additionally, flow samples are removed if the application category does not belong to one of these categories: Cloud, Media, Social Network, Video, Unspecified, and Web.

Excluding Inconsistent Flows: The fifth step removes flow samples with bidirectional flow durations of zero milliseconds or zero-count accumulators of bidirectional packets, as these are inconsistent with video traffic characteristics.

Final Compilation: The final step involves saving all remaining flow samples under a new column labeled Video Quality Category. These steps are repeated for all classes, and the remaining flow samples are merged to form a new dataset.

Overall, the process of selecting video-related flow samples using the nDPI tool is complex and meticulous. However, noise flow records that lack sufficient metadata for nDPI inspection or exhibit flow-level time–temporal features similar to normal samples are difficult to clean. Additional potential methods, such as flow aggregation or increasing the flow active timeout threshold, may help address this issue.

Algorithm 2 Pseudo-code for nDPI-detected YouTube, Facebook, and Twitch video Quality Category dataset generation (nYFTQC)

Input: Each flow sample from segmented CSV files
Output: YVAIL, YVATL, YVALL, YVAP, FVAIL, FVATL, FVALL, FVAD, TVAIL, TVATL, TVALL, TVAP dataset
for each CSV file in segmented CSV files **do**
    for each flow sample in CSV file **do**
        Read each PCAP file's prior knowledge
        Update each flow sample label
    end for
    Keep all remaining flow samples in the CSV file
end for
Update each CSV file label according to different platforms
Merger each segment CSV file into three platform categories CSV files
for Each CSV file in platform categorized CSV files **do**
    for Each flow sample in CSV file **do**
        if Flow sample application name major protocol $\neq$ "TLS", "QUIC", or "Unknown" **then**
            Delete the flow sample
        end if
        if Flow sample requested server name $\neq$ "Blanks", application name = "TLS" and application category name = "Web" **then**
            Delete the flow sample
        end if
        if Flow sample bidirectional duration or bidirectional packets = 0 **then**
            Delete the flow sample
        end if
    end for
    Save all remaining flow samples
    for Each flow sample in remaining samples **do**
        if Flow sample application category name $\neq$ "Cloud", "Media"; "SocialNetwork", "Video", "Unspecified", or "Web " **then**
            Delete the flow sample
        end if
        if Flow sample application name minor protocol $\neq$ "AmazonAWS", "AmazonVideo", "Azure", "Twitch", "Facebook", or "YouTube" **then**
            Delete the flow sample
        end if
    end for
    Save all the remaining samples
    Create a new video quality category column
    Read the updated flow sample label
end for
Save each updated CSV file
Output YVAIL, YVATL, YVALL, YVAP, FVAIL, FVATL, FVALL, FVAD, TVAIL, TVATL, TVALL, TVAP dataset

3.3. CICFlowmeter Preprocessing

CICFlowmeter preprocessing is another method for traffic metering and flow exportation. However, this tool lacks an integrated nDPI library function, leading to uninspected flow samples that may include a large number of flows generated by other software applications. To address this, it is necessary to use the Wireshark network analyzer tool. Wireshark's statistics conversations window displays all mutual traffic between two specific endpoints, including information from the link layer, IP layer, and transport layer. Typically, link layer traffic consists of Ethernet or IEEE 802.11 standard traffic [29], the IP layer covers both IPv4 and IPv6 traffic, and the transport layer includes TCP and UDP traffic.

Since the MAC address of the link layer is uniquely determined before packet sniffing, the next step is to sort all conversations in the IP layer by the number of packets, from largest to smallest. Video traffic is generally characterized by a high packet count and large byte size, and web videos delivered via CDNs may involve multiple IP addresses. By comparing the start time and duration of each conversation in seconds, these features can help identify the most prominent one to three conversations.

After selecting the most relevant one to three conversations from the Wireshark IP layer conversation window, these selected packets are then exported as a new PCAP file with the same video attribute category. Using CICFlowmeter, flow samples for each new video attribute category are exported, and a column is labeled with the corresponding name. All flow samples with zero-second durations, which have no practical meaning, are deleted, and the remaining flow samples for each video attribute category are saved. Finally, all video-category-attribute flow samples are merged into a single dataset, and this process is repeated for each of the three web video platform datasets, as outlined in the CICFlowmeter Video Quality Category (CICQC) Algorithm 3.

Algorithm 3 Pseudo-code for CICFlowmeter video Quality Category dataset generation (CICQC)

Input: Each PCAP file from the video attribute category PCAP files
Output: TVQC, YVQC, FVQC dataset
for Each PCAP file in the video attribute category PCAP files **do**
    Filter the most prominent one to three conversations in the Wireshark conversion module
    Save filtered packets as a new same video attribute category PCAP file
end for
Save all updated PCAP files
for Each PCAP file in the updated PCAP files **do**
    CICFlowmeter exports flow samples from each PCAP file
    Create a new column and label with the same video attribute category name for each flow sample
    Save all flow samples as a dataset
end for
Save three datasets according to video attribute categories
for Each dataset in datasets **do**
    for Each flow sample in the dataset **do**
        if Flow duration = 0 **then**
            Delete the flow sample
        end if
    end for
    Save all remaining samples
end for
Update all three datasets
Output TVQC, YVQC, FVQC dataset

4. Analysis of Proprietary Internet Video Traffic Datasets

For the data preprocessing described in the previous section, we used the nDACPC algorithm to generate six application and protocol category datasets: nILAD, nTLAD, nPAD, nLLAD, nILPD, nTLPD, nLLPD, and nPPD. Additionally, we leveraged the nYFTQC and CICQC algorithms to generate a total of fifteen proprietary video quality datasets: YVAIL, YVATL, YVALL, YVAP, FVAIL, FVATL, FVALL, FVAD, TVAIL, TVATL, TVALL, TVAP, TVQC, YVQC, and FVQC. This section analyzes the generated dataset characteristics from three perspectives: the imbalance ratio, flow sample histogram density and cumulative distribution, and flow-level feature correlations.

Figure 3 shows the flow record counts and distributions for YouTube, Facebook, and Twitch video quality category proprietary datasets. Across all datasets, the total number of flow records in the VLS mode consistently exceeds that in the VoD mode under identical resolution and frame rate conditions. Contrary to expectations, higher resolution does not necessarily result in more traffic or flow samples. Video traffic consumption is influenced by various video-related factors, and the number of flow samples reflects these complexities. For the nDPI video quality category datasets, all classes exhibit similar flow record counts within each category. However, Twitch generates more flow records in total and has higher sample volumes per category compared to YouTube and Facebook. In contrast, the CICFlowmeter video quality category datasets show that YouTube produces the most flow records overall. There is significant variation in sample size between categories; for instance, the 720P_30FPS_live category has 13 times more flow samples than the 720P_30FPS_VoD category. Additionally, CICFlowmeter datasets generally have category sample sizes five times larger than the corresponding nDPI categories in VLS mode. In VoD mode, the CICFlowmeter tool generates more flow samples for YouTube and Facebook compared to NFStream. However, on the Twitch platform, its sample volume is less than three times that of the corresponding NFStream dataset. Finally, the sample counts and distributions remain consistent across all accounting modes in the video quality proprietary datasets.

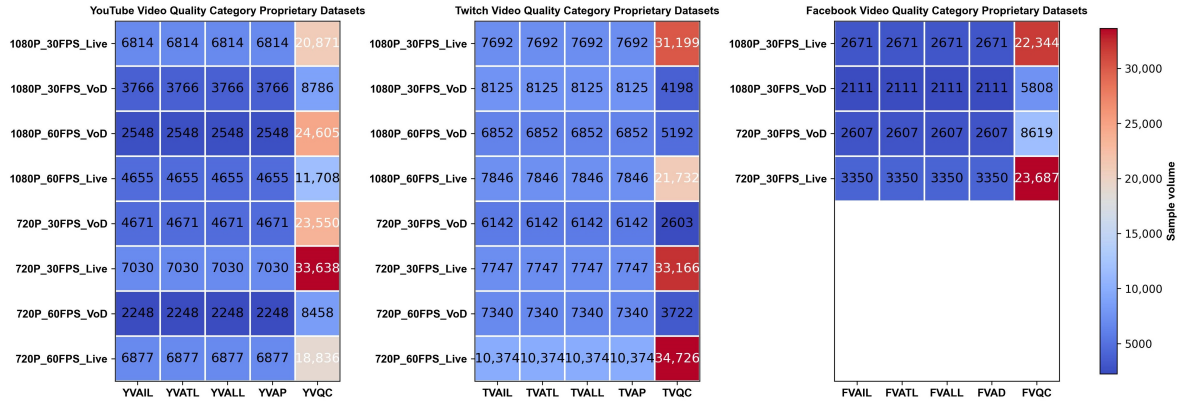


Figure 3. Video quality category proprietary datasets sample distribution.

It is worth noting that the proportions of applications and categories vary significantly after sequencing the selected video-attribute datasets. Table 1 presents the top application and protocol categories by flow sample count across different web video platforms. For the nDPI Facebook and YouTube video quality datasets, unknown flows constitute the majority, accounting for over 95% and 71% of total samples, respectively. TLS and QUIC are the two largest categories of parsed protocol categories. For the Twitch platform, TLS accounts for over 80% of the total flow volume, reflecting the trend of popular streaming platforms adopting HTTP/2 and HTTP/3 standards [30,31]. These standards integrate advanced encryption protocols, making detection and classification increasingly challenging. The flow samples of application categories exhibit varying distribution patterns. In the Facebook and YouTube datasets, the unspecified category represents the highest proportion. However, Facebook traffic is limited to the SocialNetwork category, while YouTube traffic includes Web and Media categories, highlighting the nDPI library's limitations in parsing Facebook platform traffic. For Twitch, the Web category dominates flow volume, followed by Unspecified, Video, and Cloud applications. Table 1 also lists frequently decoded server names based on SNI, including facebook.com, twitch.tv, and youtube.com. While the total number of decoded server names is relatively small, they exhibit distinct characteristics. These request servers can be categorized into direct video streaming URLs, CDN accelerator cache

domains, DNS resolution addresses, API development interfaces, gateway addresses, user login encryption addresses, and chat and conversation component addresses. Despite their diverse functions, all these addresses are related to video flow samples, further enhancing the confidence, explainability, and reliability of the datasets.

Table 1. Top represented application, protocol categories, and requested server names with quantities in three video quality category proprietary datasets.

Dataset	Top Represented Application Categories	Sample Count	Top Represented Protocol Categories	Sample Count	Top Represented Requested Server Names
nDPI Facebook Video Quality Category Proprietary Dataset	Unspecified	41,156	Unknown	41,156	www.facebook.com
	SocialNetwork	1800	QUIC	1696	scontent.xx.fbcdn.net
			TLS	104	gateway.facebook.com
nDPI Twitch Video Quality Category Proprietary Dataset	Web	198,352	TLS	202,108	www.twitch.tv
	Unspecified	46,364	Unknown	46,364	static.twitchcdn.net
	Video	2696			video-weaver.fra05.hls.ttvnw.net
	Cloud	1052			static-cdn.jtvnw.net
	Media	8			gql.twitch.tv
nDPI YouTube Video Quality Category Proprietary Dataset	Unspecified	109,856	Unknown	109,856	www.youtube.com
	Web	41,272	TLS	41,180	r1—sn-5go7yne6.googlevideo.com
	Media	3308	QUIC	3400	signaler-pa.youtube.com

4.1. Imbalance Ratio

To assess the distribution balance of samples across different label categories in the proprietary Internet video traffic datasets, suppose all classes $x_1, x_2, \ldots, x_n$ in the dataset are arranged in ascending order of magnitude and divided into quartiles, with the quartiles being the lower quartile for values in the 25% position and the upper quartile for values in the 75% position. Equation (2) shows the generic quartile estimation formula that allows the first- and third-quartile values to be calculated. Equation (1) gives the dimensionless quartile coefficient of the dispersion (cvq) calculation method based on Equation (2), where p can be either the lower or upper quartile with values of 0.25 and 0.75, respectively, n is the number of all classes in the dataset, k is the location index of a single class, and α is a coefficient determined by p and n. Symbol $\lfloor \cdot \rfloor$ indicates a round downward integer operation. Another dimensionless quantity evaluation metric is the coefficient of variation (cv), which can also be used to measure the degree of dispersion. It is the ratio of the standard deviation and the mean of the number of samples from all classes as shown in Equation (5). We use the following metrics to measure the imbalance rate in the proprietary Internet video traffic datasets.

$$cvq = \frac{q(0.75) - q(0.25)}{q(0.75) + q(0.25)} \tag{1}$$

$$q(p) = x_{(k)} + \alpha(x_{(k+1)} - x_{(k)}) \tag{2}$$

$$\alpha = p(n+1) - \lfloor p(n+1) \rfloor \tag{3}$$

$$k = \lfloor p(n+1) \rfloor \tag{4}$$

$$cv = \frac{\sqrt{n \sum_{i=1}^{n} \left(x_i - \frac{\sum_{i=1}^{n} x_i}{n} \right)^2}}{\sum_{i=1}^{n} x_i} \tag{5}$$

As shown in Table 2, Application category datasets and Protocol category datasets exhibit the highest dimensionless quartile coefficient and coefficient of variation among all datasets. This is attributed to the substantial differences in the number of samples across various labels, as illustrated in Figures 2 and 3. The nDPI Twitch video quality category proprietary datasets exhibit the lowest cvq and cv coefficients among all datasets due to bal-

anced sample volume distribution across categories. In contrast, the corresponding TVQC dataset has *cvq* and *cv* coefficients that are over 11 times and 5 times higher, respectively. For YouTube and Facebook video quality datasets, the *cvq* coefficient is approximately 0.5, indicating a medium–high value. However, the *cv* coefficient for Facebook is more than twice that of YouTube. Overall, the *cvq* and *cv* coefficients in CICFlowmeter video quality datasets are slightly higher than those in the nDPI datasets. Table 2 also presents some datasets' imbalanced rates performance of categories in existing works. Compared to other flow-level datasets, the variation in flow sample counts between categories depends on dataset design. While some datasets achieve perfect balance with *cvq* and *cv* coefficients of 0, our dataset maintains a generally low imbalance rate despite the inherent differences in category volumes.

Table 2. Measurement of the categories' imbalance rate in the proprietary Internet video traffic datasets.

Dataset	***cvq***	***cv***	**Dataset**	***cvq***	***cv***	**Dataset**	***cvq***	***cv***
nILAD	0.9620	2.9209	YVAIL	0.4127	0.3996	TVAIL	0.0719	0.1585
nTLAD	0.9620	2.9209	YVATL	0.4127	0.3996	TVATL	0.0719	0.1585
nPAD	0.9620	2.9209	YVALL	0.4127	0.3996	TVALL	0.0719	0.1585
nLLAD	0.9620	2.9209	YVAP	0.4127	0.3996	TVAP	0.0719	0.1585
nILPD	0.9713	2.9478	FVAIL	0.4983	0.8448	TVQC	0.7896	0.8538
nTLPD	0.9713	2.9478	FVATL	0.4983	0.8448	YVQC	0.4738	0.5037
nLLPD	0.9713	2.9478	FVALL	0.4983	0.8448	FVQC	0.5639	0.6094
nPPD	0.9713	2.9478	FVAD	0.4983	0.8448			
Reference	*cvq*	*cv*	Reference	*cvq*	*cv*	Reference	*cvq*	*cv*
[10]	0.3522	0.3881	[11]	0.8321	1.0149	[12]	0.0	0.0
[13]	0.5534	0.9055	[14]	0.3740	1.2235	[16]	0.0	0.0
[20]	0.4365	1.2184	[21]	0.5	0.4082	[25]	0.9669	2.3018

4.2. Video Traffic Histogram Density

In this section, we analyze the video traffic histogram density and distribution across three key features, bidirectional flow duration, bidirectional packets, and bidirectional bytes, for all video quality category proprietary datasets. To analyze the probability density function (PDF) and cumulative density function (CDF) of a specific video traffic feature vector, a histogram is used as an approximation of the PDF. Each bin represents an interval of the variable, with the normalized height approximating the probability density in that range. Increasing the number of bins while reducing bin width enhances the histogram's accuracy, allowing it to converge to the true PDF. To normalize the histogram, frequencies are scaled so the total area under the histogram equals 1, as defined in Equation (6).

$$p_i = \frac{f_i}{mh} \tag{6}$$

where

- f_i is the frequency of points in the i-th bin of the histogram;
- p_i is the height of the i-th bin of the histogram;
- m is the total number of data points;
- h is the bin width.

In practice, bin width is determined using the Sturges and Freedman–Diaconis rules, selecting the strategy that minimizes bin width. The cumulative density function (CDF) is derived from the PDF. The CDF can be estimated from a histogram-based PDF by summing the normalized heights of all bins up to and including the bin corresponding to a given value. For the bin containing the value, a proportional contribution is added based on the position of the value within the bin. This cumulative sum provides an estimate of the cumulative probability distribution. Figure 4 shows the traffic flow duration histogram

density and cumulative distribution for proprietary datasets across all video quality categories. It reveals that for most nDPI datasets from all three platforms, the majority of flow durations concentrate in the first 25 s, while the CICFlowmeter datasets exhibit the opposite trend, with most flow durations concentrated in the range of 125–180 s. We set the flow metering timer to 180 s, which means the maximum flow duration in the samples will be capped at 180 s. The two DFI tools employ different flow metering strategies. Figure 5 displays the traffic flow bidirectional packet accumulation histogram density and cumulative distribution for all datasets. It shows that most bidirectional packet accumulations fall within the 0–1500 range. However, the nDPI Twitch dataset stands out with a significant concentration of flow samples having very large bidirectional packet accumulations, particularly in the range of 120,000–150,000. Figure 6 presents the traffic flow bidirectional byte accumulation histogram density and cumulative distribution for all datasets. The majority of bidirectional byte accumulations are concentrated within the first 175,000 bytes across all datasets. Similarly, CICFlowmeter datasets consistently show higher byte accumulations than nDPI datasets.

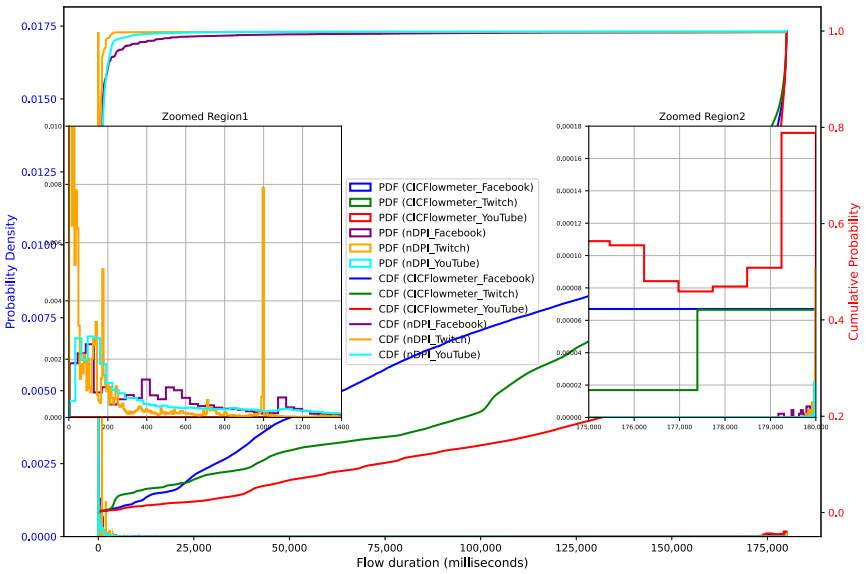


Figure 4. Traffic flow duration histogram density and cumulative distribution.

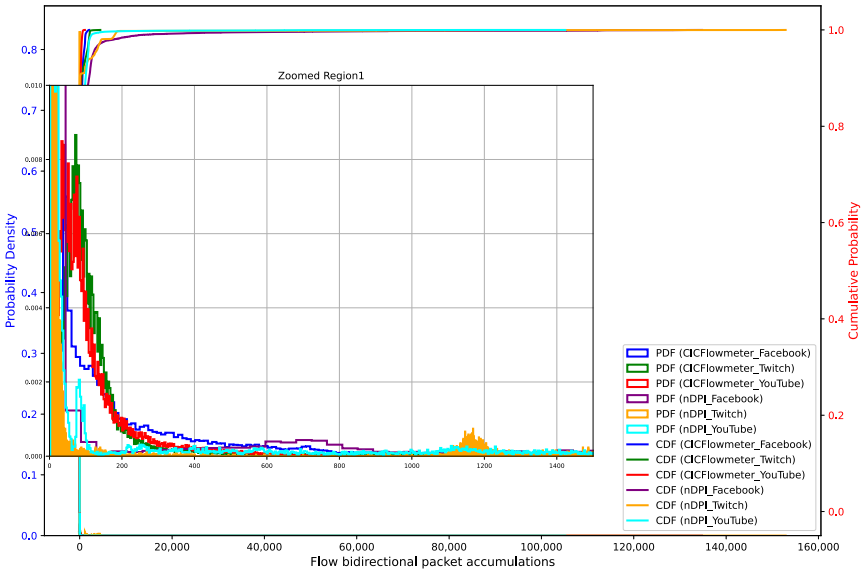


Figure 5. Traffic flow bidirectional packet accumulation histogram density and cumulative distribution.

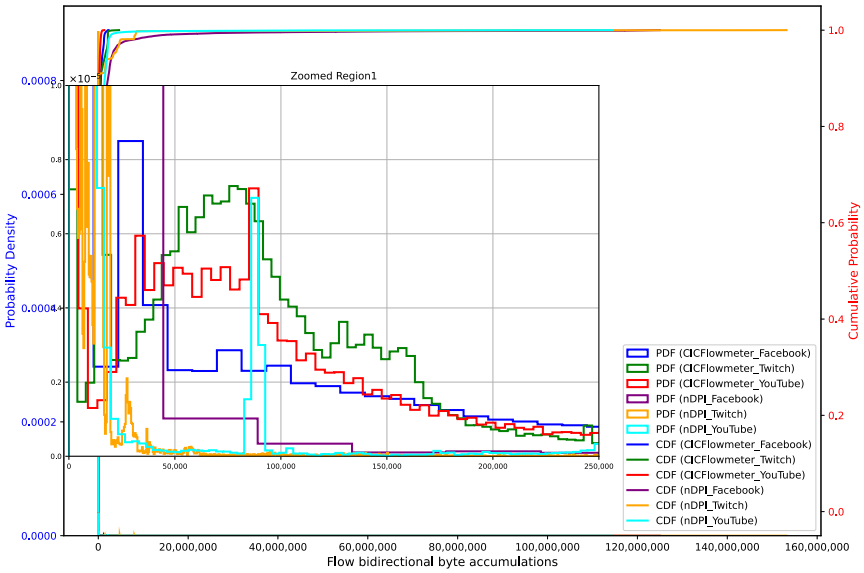


Figure 6. Traffic flow bidirectional byte accumulation histogram density and cumulative distribution.

4.3. Feature Correlation

Correlation analysis between different features in proprietary Internet video traffic datasets is essential. Based on different DFI tool labels, we merged the datasets into two categories as shown in Figure 3, and selected all numerical features for Pearson's Correlation Coefficient computation. The number of numerical features selected under the CICFlowmeter and nDPI tools were 80 and 68, respectively. Given a set of vectors $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$, where each $V_i \in \mathbb{R}^n$, m represents the total number of features. Each pair of vectors (V_i, V_j) with $i \neq j$ is chosen to compute the Pearson's Correlation Coefficient ρ_{V_i,V_j}. The formula for the Pearson's Correlation Coefficient between two vectors $X = V_i$ and $Y = V_j$ is defined in Equation (7).

$$\rho_{V_i,V_j} = \frac{\sum_{k=1}^{n}(V_{i_k} - \bar{V}_i)(V_{j_k} - \bar{V}_j)}{\sqrt{\sum_{k=1}^{n}(V_{i_k} - \bar{V}_i)^2}\sqrt{\sum_{k=1}^{n}(V_{j_k} - \bar{V}_j)^2}} \tag{7}$$

where

- V_{i_k} and V_{j_k} are the k-th elements of vectors V_i and V_j;
- $\bar{V}_i$ and $\bar{V}_j$ are the means of vectors V_i and V_j;
- n is the number of elements in each vector.

Based on the computational results presented in Figure 7, we observe that some features have a value of 0 in all samples. These concentrate on specific TCP flags, unidirectional flow in a certain direction, and some flow-related temporal features, which must be filtered using feature selection in the training model. The Pearson's Correlation Coefficient for most features is around 0, indicating a low correlation between them. In contrast to Figure 7, Figure 8 shows fewer features with a value of 0. However, the nDPI tool employs numerous statistical methods to generate features, which leads to strong correlations among them, primarily focusing on features related to packet interarrival time and packet size. This increases noise and complexity in the training model, necessitating additional feature selection and dimensionality reduction techniques to mitigate the impact of high correlation.



Figure 7. Feature correlation matrix in video quality category datasets created with CICFlowmeter.

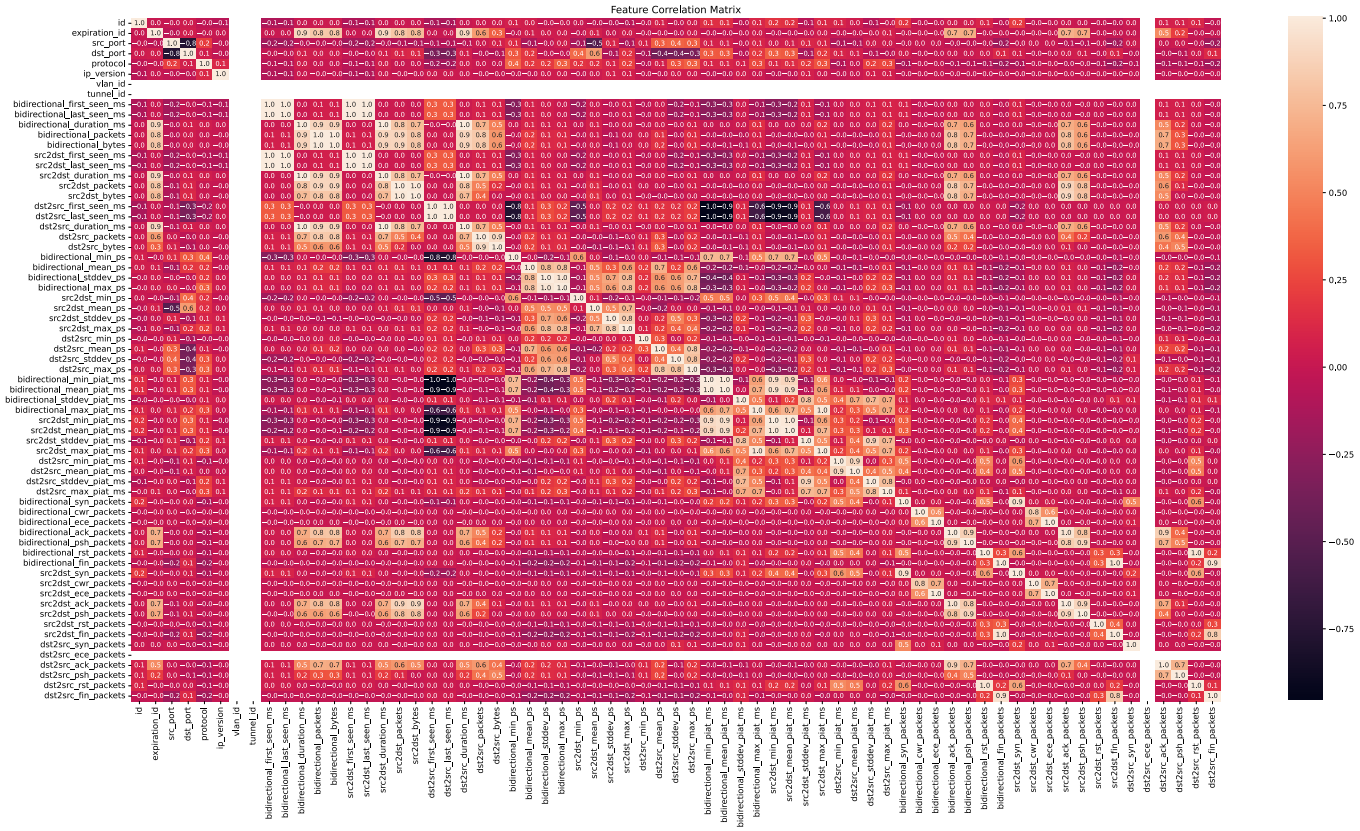


Figure 8. Feature correlation matrix in video quality category datasets created with nDPI.

5. Conclusions

This paper introduces a novel proprietary Internet video traffic dataset generation algorithm (nYFTQC), and we comprehensively demonstrate the fine-grained dataset generation process using various DFI and network traffic analysis tools. The steps involved, including data collection, collation, and label selection for each sample in the nDPI video quality category datasets, are fully transparent. All samples in the datasets are highly interpretable through analyses of imbalance rates, histogram densities, distributions, and feature correlations. Although the imbalance rate of samples across different labels is low, their feature correlation coefficients are high, accompanied by low values in duration, packets, and byte distributions. This allows for the synthesis and processing of multiple individual datasets of the same type according to training model requirements. Additionally, we developed the CICQC algorithm and generated the corresponding CICFlowmeter video quality category datasets. While these datasets contain more samples, their interpretability is limited due to the lack of metadata, and they exhibit a high imbalance rate across different labels; however, the advantage is that the feature correlation coefficient is low. The datasets generated in this study provide valuable insights for the analysis of Internet traffic in the video category.

Author Contributions: Conceptualization, M.-D.C., T.C. and E.G.; methodology, T.C.; software, T.C.; validation, E.G., A.I. and M.-D.C.; formal analysis, T.C.; investigation, T.C.; resources, T.C.; data curation, T.C.; writing—original draft preparation, T.C.; writing—review and editing, T.C., E.G., A.I. and M.-D.C.; visualization, T.C.; supervision, M.-D.C.; project administration, A.I. and E.G.; funding acquisition, M.-D.C. and T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the European Social Fund within the Project No 8.2.2.0/20/I/008 «Strengthening of PhD students and academic personnel of Riga Technical University and BA School of Business and Finance in the strategic fields of specialization» of the Specific Objective 8.2.2 «To Strengthen Academic Staff of Higher Education Institutions in Strategic Specialization Areas» of the Operational Programme «Growth and Employment», and also supported by grant PID2023-148214OB-C21 funded by MICIU/AEI/10.13039/501100011033 and FEDER/EU. This research is being carried out within the framework of the Recovery, Transformation and Resilience Plan funds, financed by the European Union (Next Generation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are publicly available at https://github.com/dossos/Datasets (accessed on 2 January 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kwak, K.T.; Lee, S.Y.; Ham, M.; Lee, S.W. The Effects of Internet Proliferation on Search Engine and Over-the-Top Service Markets. *Telecommun. Policy* **2021**, *45*, 102146.
2. Li, W.; Canini, M.; Moore, A.W.; Bolla, R. Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema. *Comput. Netw.* **2009**, *53*, 790–809.
3. Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Encrypted and VPN Traffic Using Time-Related Features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy, Rome, Italy, 19–21 February 2016; SCITEPRESS—Science and Technology Publications: Rome, Italy, 2016; pp. 407–414.
4. Habibi Lashkari, A.; Draper Gil, G.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Tor Traffic Using Time Based Features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy, Porto, Portugal, 19–21 February 2017; SCITEPRESS—Science and Technology Publications: Porto, Portugal, 2017; pp. 253–262.

5. Habibi Lashkari, A.; Kaur, G.; Rahali, A. DIDarknet: A Contemporary Approach to Detect and Characterize the Darknet Traffic Using Deep Image Learning. In Proceedings of the 2020 the 10th International Conference on Communication and Network Security, Tokyo, Japan, 27 November 2020; pp. 1–13.
6. Bader, O.; Lichy, A.; Dvir, A.; Dubin, R.; Hajaj, C. OSF-EIMTC: An Open-Source Framework for Standardized Encrypted Internet Traffic Classification. *Comput. Commun.* **2024**, *213*, 271–284.
7. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware Traffic Classification Using Convolutional Neural Network for Representation Learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; pp. 712–717.
8. Muehlstein, J.; Zion, Y.; Bahumi, M.; Kirshenboim, I.; Dubin, R.; Dvir, A.; Pele, O. Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application. In Proceedings of the 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017; pp. 1–6.
9. Pham, T.-D.; Ho, T.-L.; Truong-Huu, T.; Cao, T.-D.; Truong, H.-L. MAppGraph: Mobile-App Classification on Encrypted Network Traffic Using Deep Graph Convolution Neural Networks. In Proceedings of the Annual Computer Security Applications Conference, Virtual Event, 6 December 2021; pp. 1025–1038.
10. Liu, L.; Yu, Y.; Wu, Y.; Hui, Z.; Lin, J.; Hu, J. Method for Multi-Task Learning Fusion Network Traffic Classification to Address Small Sample Labels. *Sci. Rep.* **2024**, *14*, 2518
11. Salman, O.; Elhajj, I.H.; Chehab, A.; Kayssi, A. Towards Efficient Real-Time Traffic Classifier: A Confidence Measure with Ensemble Deep Learning. *Comput. Netw.* **2022**, *204*, 108684.
12. Arestrom, E.; Carlsson, N. Early Online Classification of Encrypted Traffic Streams Using Multi-Fractal Features. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April –2 May 2019; pp. 84–89.
13. Wu, H.; Wang, L.; Cheng, G.; Hu, X. Mobile Application Encryption Traffic Classification Based On TLS Flow Sequence Network. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
14. Xiao, X.; Xiao, W.; Li, R.; Luo, X.; Zheng, H.-T.; Xia, S.-T. EBSNN: Extended Byte Segment Neural Network for Network Traffic Classification. *IEEE Trans. Dependable Secur. Comput.* **2021**, *1*, 1.
15. Gabilondo, Á.; Fernández, Z.; Viola, R.; Martín, Á.; Zorrilla, M.; Angueira, P.; Montalbán, J. Traffic Classification for Network Slicing in Mobile Networks. *Electronics* **2022**, *11*, 1097.
16. Wang, Z.; Dong, Y.; Shi, H.; Yang, L.; Tang, P. Internet Video Traffic Classification Using QoS Features. In Proceedings of the 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15–18 February 2016; pp. 1–5.
17. Dong, Y.; Zhao, J.; Jin, J. Novel Feature Selection and Classification of Internet Video Traffic Based on a Hierarchical Scheme. *Comput. Netw.* **2017**, *119*, 102–111.
18. Canovas, A.; Jimenez, J.M.; Romero, O.; Lloret, J. Multimedia Data Flow Traffic Classification Using Intelligent Models Based on Traffic Patterns. *IEEE Netw.* **2018**, *32*, 100–107.
19. Hayashi, K.; Ooka, R.; Miyoshi, T.; Yamazaki, T. P2PTV Traffic Classification and Its Characteristic Analysis Using Machine Learning. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–6.
20. Costa Da Silva, L.; Monks, E.; Yamin, A.; Reiser, R.; Bedregal, B. ω-IvE Methodology: Admissible Interleaving Entropy Methods Applied to Video Streaming Traffic Classification. *Int. J. Approx. Reason.* **2024**, *164*, 109061.
21. Wu, Z.; Dong, Y.; Jin, J.; Wei, H.-L.; Xie, G. Multimedia Traffic Classification for Imbalanced Environment. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1838–1852.
22. Wu, H.; Li, X.; Cheng, G.; Hu, X. Monitoring Video Resolution of Adaptive Encrypted Video Traffic Based on HTTP/2 Features. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10 May 2021; pp. 1–6.
23. Bukhari, S.M.A.H.; Afaq, M.; Song, W.-C. PPS: A Packets Pattern-Based Video Identification in Encrypted Network Traffic. In Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing, Taormina, Messina, Italy, 4 December 2023; pp. 1–6.
24. Ozkan, H.; Temelli, R.; Gurbuz, O.; Koksal, O.K.; Ipekoren, A.K.; Canbal, F.; Karahan, B.D.; Kuran, M.Ş. Multimedia Traffic Classification with Mixture of Markov Components. *Ad Hoc Netw.* **2021**, *121*, 102608.
25. Alcock, S.; Nelson, R. Measuring the Accuracy of Open-Source Payload-Based Traffic Classifiers Using Popular Internet Applications. In Proceedings of the 38th Annual IEEE Conference on Local Computer Networks—Workshops, Sydney, Australia, 21–24 October 2013; pp. 956–963.
26. Rescio, T.; Favale, T.; Soro, F.; Mellia, M.; Drago, I. DPI Solutions in Practice: Benchmark and Comparison. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 27 May 2021; pp. 37–42.

27. Deri, L.; Martinelli, M.; Bujlow, T.; Cardigliano, A. nDPI: Open-Source High-Speed Deep Packet Inspection. In Proceedings of the 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 4–8 August 2014; pp. 617–622.
28. Aouini, Z.; Pekar, A. NFStream A Flexible Network Data Analysis Framework. *Comput. Netw.* **2022**, *204*, 108719.
29. *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*; IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN. IEEE: Piscataway, NJ, USA, 2021; pp. 1–767.
30. Thomson, M.; Benfield, C. HTTP/2; RFC Editor, 2022; p. RFC9113. Available online: https://www.rfc-editor.org/rfc/rfc9113.html (accessed on 2 December 2024).
31. Bishop, M. HTTP/3; RFC Editor, 2022; p. RFC9114. Available online: https://www.rfc-editor.org/rfc/rfc9114.html (accessed on 2 December 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.