# Clustering-based Separation of Media Transfers in DPI-classified Cellular Video and VoIP Traffic

Johan Garcia
Department of Mathematics and Computer Science
Karlstad University, Sweden
Email: johan.garcia@kau.se

Anna Brunstrom
Department of Mathematics and Computer Science
Karlstad University, Sweden
Email: anna.brunstrom@kau.se

***Abstract*—Identifying VoIP and video traffic is often useful in the context of managing a cellular network, and to perform such traffic classification deep packet inspection (DPI) approaches are often used. Commercial DPI classifiers do not necessarily differentiate between, for example, YouTube traffic that arises from browsing inside the YouTube app, and traffic arising from the actual viewing of a YouTube video. Here we apply unsupervised clustering methods on such cellular DPI-labeled VoIP and video traffic to identify the characteristic behavior of the two sub-groups of media-transfer and non media-transfer flows. The analysis is based on a measurement campaign performed inside the core network of a commercial cellular operator, collecting data for more than two billion packets in 40+ million flows. A specially instrumented commercial DPI appliance allows the simultaneous collection of per packet information in addition to the DPI classification output. We show that the majority of flows falls into clusters that are easily identifiable as belonging to one of the traffic sub-groups, and that a surprising majority of DPI-labeled VoIP and video traffic is non-media related.**

## I. Introduction

Machine learning (ML) has established itself as a key methodology for network performance optimization with application in areas such as traffic classification and anomaly detection. However, supervised learning techniques require training of the ML model and the achieved performance is directly dependent on the quality of training data and the accurate knowledge of ground truth for this data. In this paper we address how to provide more accurate ground truth for training, targeting enhanced traffic classification of video and voice traffic in the context of cellular networks.

A large fraction of the traffic sent over cellular networks today is media traffic, with video making up the majority of the transferred bytes. As video and VoIP traffic puts demands on both bandwidth and delay, and its performance also greatly influences user satisfaction, mobile operators commonly perform traffic management for media traffic. In order to perform traffic management of video and VoIP flows, traffic classification is needed to separate these flows from the rest of the traffic. To this end, operators commonly apply Deep Packet Inspection (DPI) to examine the traffic and identify the flows of interest. However, this practice is today being challenged by the increased use of encryption over the Internet. Here, ML based methods that can base the flow classification on flow characteristics that are available also for encrypted traffic, such as packet sizes, packet direction and packet timing aspects, present an attractive alternative.

As mentioned, ML-based traffic classification of video and voice flows does, however, require accurate training of the ML model. Currently available DPI classifiers can be used to establish ground truth for the training data. However, as commercial DPI classifiers can base the separation of traffic on identifying the application, they do not necessarily differentiate between, for example, YouTube flows that emanate from browsing inside the YouTube app, and flows associated with the actual viewing of a YouTube video. Including non-media related flows in the flows labeled as video or voice can add considerable noise to the training data. It is thus beneficial to further separate actual media flows from other flows classified as media by the DPI classifier as a result of them being part of a media application. We address how to perform this separation in our work.

We apply unsupervised clustering methods on DPI-labeled VoIP and video traffic to analyze characteristic features that can be used to separate flows used for media transfer from other flows in media applications. The analysis is based on a large data set captured in the core network of a commercial cellular operator. The data set consists of more than two billion packets transferred in over 40 million flows by 53090 devices. In support of our analysis, a DPI classifier was instrumented to also capture per packet statistics, in addition to the DPI classification output. The results of our analysis show that a majority of flows belong to clusters that are easily identifiable as containing either media or non-media flows. We also find that the majority of flows labeled as video or VoIP by the DPI classifier carry other traffic and are not related to the media transfer. Separating them out in the training data will significantly improve ground truth and lead to enhanced ML models.

## II. Related Work

Related problems have been studied by a considerable body of previous works. Moore et al. [1] is one of the first studies considering ML for traffic classification while Nguyen et al. [2] survey early ML based classification approaches. Finsterbush et al. [3] provide a survey of payload-based classification approaches, and Velan et al. provide an overview of methods for encrypted traffic classification. In [4], Erman et al.

examines traffic classification using K-means and DBSCAN approaches. Xu et al. [5] examines how to select optimal features for use in statistical-based traffic classification. Shbair et al. [6] consider classification of encrypted traffic, and suggest a multi-level approach to identify application classes in HTTPS traffic. Fu et al. [7] study encrypted cellular traffic, focusing in particular on how to classify mobile messaging apps. Correlating HTTPS traffic and DNS requests to allow host-based service classification of encrypted traffic is examined by Mori et al. [8]. Work by Casas et al. [9] uses DPI-labeled flows as ground truth with a semi-supervised learning approach for flow classification.

To allow improved ground truth quality over regular DPI, Gringoli et al. [10] propose the gt toolset to collect information on the process which generates a particular flow and use this to improve ground truth quality. A study by Dusi et al. [11] using gt show that the examined DPI engine (L7-filter) had considerable weaknesses with some traffic types. Bujlow et al. [12] propose a similar process-monitoring system (VBS), and use it to evaluate the classification performance of six DPI-based network classifiers. However, these studies do not consider that the same process may generate different traffic types which is a core consideration in this work. Based on a limited data set, Garcia [13] previously reported initial results on how to distinguish media flows from other traffic in video applications. By analyzing a recent large scale cellular data set our work extends the knowledge regarding video and VoIP flow behavior as observable by mobile operators and their DPI equipment, focusing on features available also to emergent encrypted traffic.

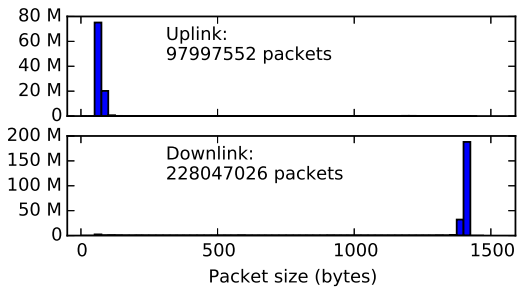
## III. Description of Data Set

### A. Data collection

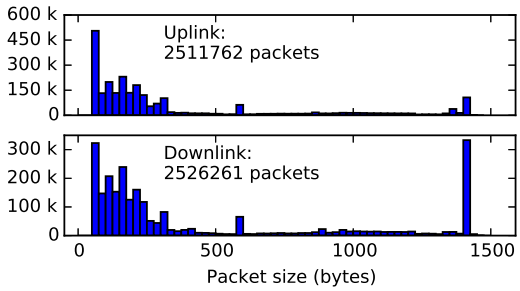
The data set was collected from inside the cellular network backbone of a commercial cellular operator. An operational DPI box was modified to collect anonymized information for each packet in a flow, up to a maximum collection time of one minute per flow. For each packet, the data also included the flow application label as inferred by the DPI engine at that point in time. The DPI engine has over 1000 applications that it differentiates between, and it can update the classification of a flow as more packets are observed. Two lists of applications that were considered video and VoIP applications, respectively, were provided by the DPI vendor.

Data collection was performed during 18 hours in February 2017. The data set consists of 42 million unique flows, which in the captured data set had a total of 2.15 billion packets and represented a transfer of 1.66 TiB.

Data sanitization was employed to remove flows which were started during the last 30 seconds of the capture window, or were initiated before the start of the capture window. As the capture was performed on a single link of a load-balanced connection pair where non-synchronized hashing was used to distribute the load, approximately half of all flows were only observable in a single direction. Flows with traffic observed in only one direction were not included in the further analysis.




(a) Video flows




(b) VoIP flows

Figure 1: Packet size histograms

Flows with only a small number of packets exchanged, 10 or less, were also excluded from this examination.

### B. Data set characterization

As a baseline for our further analysis, we first describe some of the key characteristics of our data set. The packet size distributions for packets in flows that were classified as video and VoIP applications by the DPI engine are illustrated in Figure 1a and Figure 1b, respectively. Looking first at the video flows, we can see that the majority of packets were sent in the downlink direction and that most of those packets were large in size. This is consistent with video flows downloading large amounts of data, leading to a large fraction of full-sized packets in the downlink and to a large fraction of smaller acknowledgement packets in the uplink. It should be noted that the video flows contribute roughly 40% of all packets, and hence the packets size distribution of the data set as a whole is heavily influenced by this traffic. As expected, the packet size distribution for the VoIP flows looks markedly different. Consistent with the communication pattern of VoIP, most packets are smaller in size with a roughly equal number of packets in both the uplink and downlink.

To further characterize the data set we examine the distinct applications that generated the traffic, as identified by the DPI engine. The most common applications, based on the number of observed flows, for the data set as a whole are shown in Table I. The corresponding information for the video applications and VoIP applications are shown in Tables II and III, respectively. We can see in Table I that the most common applications have quite different characteristics. While SSLv3,

Table I: Most frequent application classes overall

| | Application | Nr of flows | Mean DL Packetsize | Observed packets | Transport protocol |
|---|---|---|---|---|---|
| 1 | SSLv3 | 976799 | 973 | 52600232 | TCP |
| 2 | Google | 697102 | 933 | 48770994 | UDP/TCP |
| 3 | HTTP | 501443 | 1175 | 39632523 | TCP |
| 4 | Facebook | 451766 | 1025 | 46896789 | TCP |
| 5 | Youtube | 308266 | 1378 | 257464382 | UDP/TCP |
| 6 | Instagram | 306007 | 1323 | 88351048 | TCP |

Table II: Most frequent video applications

| | Application | Nr of flows | Mean DL packetsize | Observed Packets | Transport protocol |
|---|---|---|---|---|---|
| 1 | YouTube | 308266 | 1378 | 257 Mill | UDP/TCP |
| 2 | HTTP media str. | 39255 | 1389 | 52.0 Mill | TCP |
| 3 | Netflix | 16555 | 1387 | 13.2 Mill | TCP |
| 4 | Kodi | 852 | 1389 | 0.55 Mill | TCP |
| 5 | Flash video | 451 | 1377 | 1.49 Mill | TCP |

the top application in number of observed flows, contributes more than three times as many flows as YouTube, YouTube contributes roughly five times as many packets as SSLv3. Looking at the video applications in Table II, we see that Youtube heavily dominates and that the three most common applications cover almost all of the video traffic. As seen in Table III, the number of flows classified as VoIP in the data set is smaller, with Skype-SSL contributing the majority of the flows. We also see that the average packet sizes between the VoIP applications varies a bit more, as was also reflected in the distribution of packets sizes in Figure 1b.

## IV. Clustering aspects

### A. Employed features

For the clustering task a straightforward feature set consisting of the transport protocol, and 14 statistics-based features were utilized. Seven different features were computed separately for all packets in each flow both for the downlink and uplink directions. The features were the log of the number of packets, and mean, maximum, standard deviation, variance, skew, and kurtosis of the packet sizes. The features were scaled to zero mean and unit variance. All these features can be computed also for encrypted traffic.

### B. Clustering algorithms

A large number of clustering algorithms exists, and for this examination two different clustering approaches based on expectation maximization (K-means) and agglomerative clustering (BIRCH) are employed. The specifics of the current examination necessitates some restrictions on the algorithms. The algorithms should be able to scale well with large number of observations, and it is desirable that the algorithms should be able to perform cluster formation and cluster membership assignment separately. This allows the clusters to be formed using only a smaller subset of a potentially very large data set, while still being able to assign cluster membership to all observations in a large data set allowing for large training sets to be constructed.

Table III: Most frequent VoIP applications

| | Application | Nr of flows | Mean DL packetsize | Observed Packets | Transport protocol |
|---|---|---|---|---|---|
| 1 | Skype-SSL | 25951 | 829 | 945981 | TCP |
| 2 | Skype | 2285 | 253 | 307930 | TCP/UDP |
| 3 | WhatsApp voice | 1998 | 397 | 1438537 | UDP |
| 4 | Viber SSL | 1319 | 290 | 35861 | TCP |
| 5 | Facebook VoIP | 729 | 461 | 908642 | TCP/UDP |

Using two different clustering methods allows a straightforward comparison of their results, which can serve as indication of how algorithm-specific the obtained results are.

For K-means, this evaluation uses the K-means++ variant [14] with Euclidean distances. Each clustering invocation uses ten runs with different centroid starting points, returning the run with the best within-cluster sum-of-squares. K-means have been extensively used in previous literature on flow classification. K-means typically works best for data where the clusters are convex shaped, has equal variance in all dimensions, and are of similar density.

BIRCH [15] is a hierarchical clustering algorithm that builds a clustering feature tree and iteratively splits/merges nodes before employing a hierarchical clustering on the subclusters to arrive at a specified number of clusters.

### C. Cluster evaluation metrics

Both the employed algorithms require the number of clusters as input to the algorithm. In our context this number is not known a-priori, but a suitable number $N$ of clusters needs to be identified. Several clustering metrics are available but many require the ground-truth cluster levels to be known, which is not the case here. The properties of several clustering metrics are discussed by Liu et al. [16], and in this study two such metrics are used. The silhouette score [17] represent the average distance between each sample and all points in the same cluster, and the next nearest cluster. For each sample, the silhouette coefficient is composed from two scores:

$a$: The mean distance between a sample and all other points in the same cluster.

$b$: The mean distance between a sample and all other points in the next nearest cluster.

The silhouette coefficient for a single sample is then given as: $s = \frac{b-a}{max(a,b)}$. The mean silhouette score for all samples can then be used as an overall metric of clustering performance. In addition to the silhouette score the second metric, the Calinski-Harabasz index, is also computed as described in [18].

For each N and each metric we compute a score normalized over the range of N considered. As we need enough clusters to separate out the behaviors of different flow types, but not more clusters than can be readily interpreted we chose a range of $3 < N < 10$. A composite metric is created by equal weighting of the two normalized scores, and the N which has the highest score is chosen.

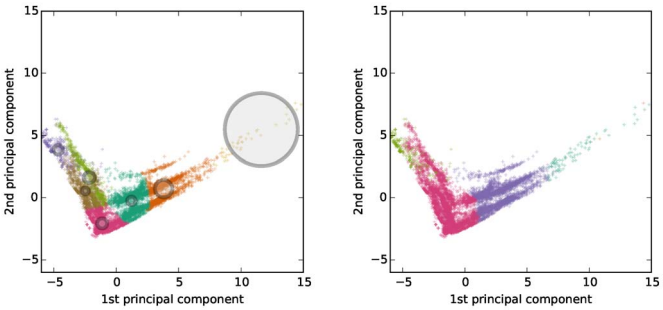
## V. Video flow clustering

The proposed approach consists of the steps: (1) Principal Component Analysis (PCA) transformation, (2) Sample the flow population to obtain a smaller process flow set, (3) Determination of $N$ over the process set, (4) Create clustering model for the process flow set using the selected $N$, (5) Use the obtained cluster model to cluster the complete flow set, (6) Examine cluster characteristics and assign media transfer relevance labels, (7) Calculate per application cluster membership. The approach produces a clustering model that can be used to divide DPI-classified VoIP and video flows into flows that are likely or unlikely to be media-transporting flows. Statistics to provide additional insight into DPI-labeled application behavior can also be extracted. The two-pronged approach with a sampled cluster model creation step, and a later full cluster assignment step, allows a very large number of flows to be used for the ground truth generation.

Unsupervised techniques for reducing the number of dimensions can be useful both to improve clustering performance if the original number of features is large, and to provide a more complete two-dimensional visual representation of higher-dimensional data. Such visual representation provides an overview of the relationship and scale of total variation. Principal component analysis (PCA) is a technique which performs linear transformations to maximize the variance for the resulting principal components. As the number of features in this study is relatively small, PCA here has utility mainly for visualization rather than feature reduction. In step (1), a PCA transformation is performed on the 15-dimensional data set of the employed features, resulting in a 15-dimensional PCA representation.

The flows are sampled in step (2) with a process set size of 10000 flows, and then in step (3) candidate clusterings are computed over the range of $N$, with the composite metric used to select the appropriate $N$ for each clustering method. The clustering is performed in PCA space, here using all components as the number of components (15) is relatively small. If feature sets with higher numbers of features are used, a subset of PCA components would be used to reduce dimensionality. In step (3) the resulting number of clusters is $N = 7$ for K-means, and $N = 5$ for BIRCH.

After performing step (4) a clustering model is obtained, and the resulting clustering can be visualized. The 10000 flows of the process flow set is shown along the two primary PCA components in Figure 2. The sub-figures thus show the projection of the 15-dimensional feature space onto a two-dimensional plane giving maximum variation, which in this case explains 59 percent of the total variance. However, Figure 2 contains no information regarding the spread in the remaining 13 PCA dimensions. For K-means, the resulting clusters are shown in Figure 2a. Also shown are the cluster centroids, and a relative indication of the cluster spread. As seen in Figure 2b, BIRCH produces a quite similar clustering, although fewer clusters will be formed. Due to the use of sub-clusters BIRCH cannot provide a per cluster model centroid.




(a) K-means clustering (b) BIRCH clustering

Figure 2: PCA scatter plots of 10000 video flows

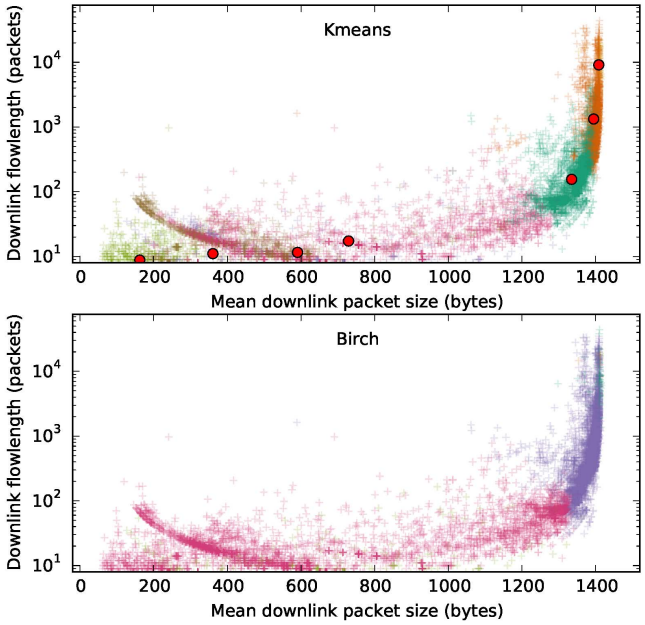



Figure 3: Video clusters in feature space

By inverting the PCA transform and the feature scaling, it is possible to map the obtained clusters back to the original feature space as shown in Figure 3. Contrasting to the PCA space clustering in Figure 2, it can be seen that in PCA space the flows are more spread out compared to what in Figure 3 is a quite cramped upper right corner. For K-means the cluster centroids are also indicated in Figure 3.

Considering only the two features plotted in Figure 3, it would be expected that flows transferring actual video would have a high per flow mean downlink packet size (x-axis), and a high downlink flow length (y-axis). Note that the flow length number is bounded by the one minute per flow data collection limit. For K-means it would seem that the three leftmost clusters could be holding video-transfer related flows, and the remaining clusters likely hold non video-transfer flows. For BIRCH the visual interpretation is harder, but studying Figures 2b and 3 jointly it can be concluded that of the five clusters, two are likely video-transfer dominated and two non

video-transfer related. Additional features can be similarly visualized. The purpose of these figures is however to obtain a visual intuition for the process data set at the resulting clusters, and not a formal part of the overall approach.

The next step in the approach, (5), is to use the created model to cluster the complete data set. The distribution of the number of flows per cluster after applying the K-means clustering model on the complete data set is shown in the top line of Table IV. Next comes the only step in the approach that requires human input, step (6), which is the media transfer relevance assignment. While the earlier figures could provide visual hints, a more quantitative approach is desirable. To perform step 6, a set of quantitative measures are computed for the obtained complete data clusters. These measures could be the centroids of relevant dimensions in the feature space, other statistical aspects of the cluster points, or metrics derived through other processing. The purpose of these metrics is to allow a domain expert to label clusters as likely media related or media non-related. How many, and which metrics to use is up to the discretion of the domain expert. For this example we use the three straightforward metrics: (A) Mean of all per flow downlink-packetsize-mean, (B) Mean of all downlink flow lengths (in packets for the one minute collection limit), (C) Max of all downlink flow lengths. The numerical results for these three metrics are shown in the middle part of Table IV. The numeric results show that clusters 2 and 6 are consistent with what could be expected of a flow transporting video. Clusters 3,4,5 and 7 appear unlikely to represent flows actually transferring video. Cluster 1 has a too small average DL flow length to be dominated by video transfer, but the max flow length shows that the cluster likely holds a number of flows which are indeed video transfers. The video-transporting clusters are shown with boldface type in the tables.

The corresponding BIRCH clustering results are shown in Table V. Examining the metrics, it can be seen that clusters 1,2 and 3 are consistent with transporting actual video, and cluster 4 and 5 are not. Comparing the two clustering approaches we find that K-means classifies 47% of the flows to clusters (which here includes cluster 1) with video transfer characteristics, while BIRCH classifies 37%.

Overall, the results are quite promising as, even though not perfect, they allow a considerable increase in the quality of DPI-based ground truth data used to train flow classifiers which aim to identify actual video transfer flows. It can also be noted that as two separate clustering results are available, it is possible to pool these results and make an improved aggregate clustering result. It is also possible to use the procedure with additional clustering algorithms, or to make a separate (sub-)clustering of clusters which are non-obvious to assign.

The procedure also has a step (7) which can provide further insight into the DPI-label to cluster mapping. The bottom parts of Tables IV and V lists the percentage fraction of flows with DPI-labels according to Table II, that got assigned to each of the clusters. From the tables it can be observed that for YouTube the majority of flows are actually not transporting video information. It can also be seen that while there is some variation, on a large scale the fraction of video transferring flows is showing a level of consistency between the two clustering approaches.

Table IV: Video flow clustering with K-mean.

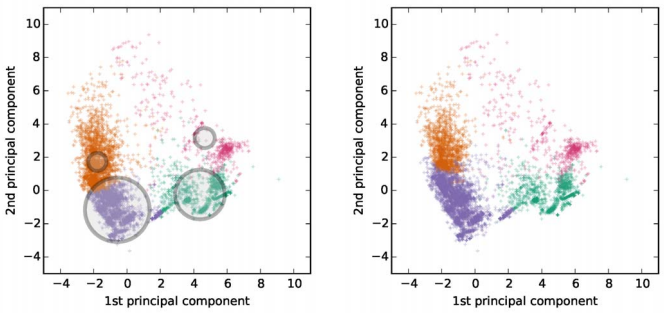
| Cluster label: | **1** | **2** | 3 | 4 | 5 | **6** | 7 |
|---|---|---|---|---|---|---|---|
| Flows in cluster | **104k** | **66k** | 14k | 82k | 26k | **1698** | 73k |
| Mean DL pkt sz | **1335** | **1395** | 579 | 726 | 159 | **1404** | 361 |
| Mean DL flw lgth | **255** | **2710** | 16 | 26 | 10 | **11k** | 15 |
| Max DL flow lgth | **16k** | **73k** | 904 | 4057 | 1383 | **57k** | 419 |
| YouTube | **27** | **13** | 4.7 | 24 | 7.9 | **0.3** | 23 |
| HTTP media str. | **27** | **52** | 0.0 | 14 | 2.5 | **1.8** | 2.7 |
| Netflix | **65** | **29** | 0.0 | 5.4 | 0.5 | **0.4** | 0.1 |
| Kodi | **6.8** | **52** | 0.0 | 6.1 | 35 | **0.0** | 0.0 |
| Flash video | **20** | **65** | 0.0 | 1.3 | 0.9 | **14** | 0.0 |

Table V: Video flow clustering with BIRCH

| Cluster label: | **1** | **2** | **3** | 4 | 5 |
|---|---|---|---|---|---|
| Flows in cluster | **3024** | **144** | **134k** | 218k | 12k |
| Mean downlink packet size | **1403** | **1382** | **1374** | 620 | 627 |
| Mean downlink flow length | **8638** | **18k** | **1444** | 29 | 14 |
| Max downlink flow length | **55k** | **57k** | **73k** | 6193 | 904 |
| YouTube | **0.5** | **0.0** | **30** | 66 | 3.8 |
| HTTP media str. | **3.0** | **0.2** | **72** | 25 | 0.0 |
| Netflix | **0.5** | **0.1** | **76** | 23 | 0.0 |
| Kodi | **0.0** | **0.0** | **57** | 43 | 0.0 |
| Flash video | **15** | **2.8** | **79** | 4.2 | 0.0 |

## VI. VoIP FLOW CLUSTERING

When applied to VoIP flows the approach in step (3) yields $N = 4$ for both clustering approaches. The clusterings of the process flow set in PCA space are shown in Figure 4. The clusterings visually appear quite similar. Figure 5 shows the clustering transformed and scaled into the feature space for K-means only. For step (6), the selected metrics are here slightly different. For this traffic type the size and symmetry of the up and downlink packet sizes are deemed relevant to determine actual VoIP transfer. The K-means clustering results for all flows are shown in Table VI. Considering the size and symmetry between the up- and down-link mean packet sizes, clusters 1 and 4 are consistent with VoIP transfers. Somewhat surprising is the mean downlink flow length for cluster 1 which appears very small. The exact reason for this observed behavior is currently unknown, but some aspect of silence detection could possibly be involved. Examining the DPI-label allocation over the clusters supports the notion that cluster 1 is VoIP transport related. The list of VoIP DPI-labels obtained by the DPI vendor clearly includes labels not transporting actual VoIP data, and these are very consistently separated in the clusters. The results from the BIRCH clustering is very similar with regards to VoIP and non VoIP-transfer separation. The ratios of VoIP flows are 18.2 % for both K-means and BIRCH clustering. It can also be noted that although the employed packet size features provide a clear separation of flow clusters for VoIP, extending the employed features to also include packet timing features could provide additional benefits.




(a) K-means clustering (b) BIRCH clustering

Figure 4: PCA scatter plots of 10000 VoIP flows

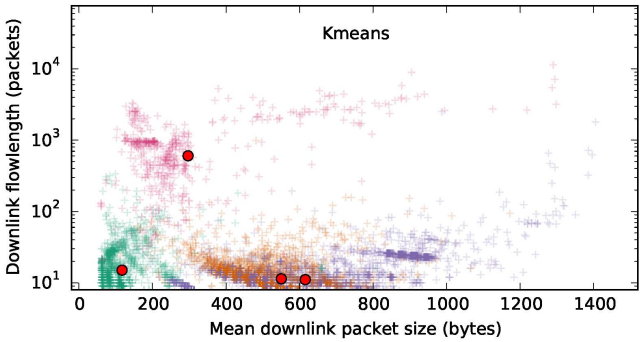



Figure 5: VoIP clusters in feature space

Table VI: VoIP flow clustering with K-mean

| Cluster label: | **1** | 2 | 3 | **4** |
|---|---|---|---|---|
| Flows in cluster | **4235** | 9352 | 17917 | **1842** |
| Mean downlink packet size | **115** | 549 | 619 | **290** |
| Mean uplink packet size | **124** | 334 | 170 | **306** |
| Mean downlink flow length | **22** | 13 | 17 | **1087** |
| Skype-SSL | **0.5** | 36 | 64 | **0.1** |
| Skype | **96** | 0.9 | 0.8 | **2.3** |
| Whatsapp voice | **59** | 3.1 | 0.9 | **37** |
| Viber SSL | **16** | 0.0 | 84 | **0.1** |
| Facebook voice | **32** | 0.0 | 0.1 | **68** |

## VII. Conclusions

We propose a scalable and flexible clustering-based analysis approach for improving ground truth data quality for DPI-labeled video and VoIP flows. Using a large scale data set from a commercial cellular operator, we apply the approach and analyze flows labeled as belonging to video or VoIP applications by the DPI engine used by the operator. Based on flow characteristics available also for encrypted traffic and using both K-means and BIRCH clustering approaches we examine flow clusters in the data set. Our results indicate that flows related to the actual media transfers to a large extent can be readily separated from non-media related flows. Surprisingly, we find that the majority of flows coupled to VoIP or video applications by the DPI engine are non-media related. When accurate classification of media transfers is important, our proposed approach is expected to provide substantial benefits when applied as a preprocessing step before using DPI-labeled flows as ground truth input for training machine learning models.

## Acknowledgments

The authors wish to thank Sandvine for support and for providing the data set. Funding for this study was partly provided by the HITS project grant from the Swedish Knowledge Foundation.

## References

[1] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, 2005, pp. 50–60.

[2] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[3] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[4] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM workshop on Mining Network Data*, 2006, pp. 281–286.

[5] M. Xu, W. Zhu, J. Xu, and N. Zheng, "Towards selecting optimal features for flow statistical based network traffic classification," in *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*, 2015, pp. 479–482.

[6] W. M. Shbair, T. Cholez, J. François, and I. Chrisment, "A Multi-Level Framework to Identify HTTPS Services," in *IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 240–248.

[7] Y. Fu, H. Xiong, X. Lu, J. Yang, and C. Chen, "Service usage classification with encrypted internet traffic in mobile messaging apps," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[8] T. Mori, T. Inoue, A. Shimoda, K. Sato, K. Ishibashi, and S. Goto, "Sfmap: Inferring services over encrypted web flows using dynamical domain name graphs," in *International Workshop on Traffic Monitoring and Analysis*. Springer, 2015, pp. 126–139.

[9] P. Casas, J. Mazel, and P. Owezarski, "Minetrac: Mining flows for unsupervised analysis amp; semi-supervised classification," in *2011 23rd International Teletraffic Congress (ITC)*, Sept 2011, pp. 87–94.

[10] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso *et al.*, "Gt: picking up the truth from the ground for internet traffic," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 12–18, 2009.

[11] M. Dusi, F. Gringoli, and L. Salgarelli, "Quantifying the accuracy of the ground truth associated with internet traffic traces," *Computer Networks*, vol. 55, no. 5, pp. 1158–1167, 2011.

[12] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular dpi tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, 2015.

[13] J. Garcia, "A clustering-based analysis of DPI-labeled video flow characteristics in cellular networks," in *2nd IFIP/IEEE International Workshop on Analytics for Network and Service Management*, 2017.

[14] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07, 2007, pp. 1027–1035.

[15] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," in *Proceedings of ACM SIGMOD Conference*, ser. SIGMOD '96, 1996, pp. 103–114.

[16] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *IEEE 10th International Conference on Data Mining (ICDM)*, 2010, pp. 911–916.

[17] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[18] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.