IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 27, 2020, accepted November 9, 2020, date of publication November 16, 2020, date of current version November 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037971

Smart User Consumption Profiling: Incremental Learning-Based OTT Service Degradation

JUAN SEBASTIÁN ROJAS[1], ADRIAN PEKAR[2], ÁLVARO RENDÓN[1], (Senior Member, IEEE), AND JUAN CARLOS CORRALES[1]

[1]Telematics Engineering Research Group, Universidad del Cauca, Popayán 190003, Colombia
[2]Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, 1117 Budapest, Hungary

Corresponding author: Adrian Pekar (apekar@hit.bme.hu)

This work was supported by the National Research, Development and Innovation Office (NKFIH), Hungary, under Grant BME IE-NAT TKP2020. The work of Juan Sebastián Rojas was supported by the Administrative Department of Science, Technology, and Innovation of Colombia (MINCIENCIAS) through the Scholarship Granted in the Call for National Doctorates under Grant 727-2015.

ABSTRACT Data caps and service degradation are techniques used to control subscribers' data consumption. These techniques have emerged mainly due to the growing demands placed on the networking stack created by the continuous increase in the number of connected users and their feature-rich, bandwidth-heavy Over-the-Top (OTT) applications. In the mobile network's scope, where traditional operators offer user data plans with limited resources, service degradation is a standard mechanism used to throttle consumption. Limiting user data usage helps to utilize resources better and to ensure the network's reliable performance. Nevertheless, this degradation is applied in a generalized way, affecting all user applications without considering behavior. In this paper, we propose a reference model aiming to address this constraint. Specifically, we attempt to personalize service degradation policies by providing a guideline for users' OTT consumption behavior classification based on Incremental Learning (IL). We evaluated our model's viability in a case study by investigating the efficacy of several IL algorithms on a dataset containing real-world users' OTT application consumption behavior. The algorithms include Naive Bayes (NB), K-Nearest Neighbor (KNN), Adaptive Random Forest (ARF), Leverage Bagging (LB), Oza Bagging (OB), Learn++, and Multilayer Perceptron (MLP). The obtained results show that ARF and a composition between LB and ARF achieve the best performance yielding a classification precision and recall of over 90%. Based on the obtained results, we propose service degradation policies to support decision making in mission-critical systems. We argue the strong applicability of our model in real-world scenarios, especially in user consumption profiling. Our reference model offers a conceptual basis for the tasks that need to be performed when defining personalized service degradation policies in current and future networks like 5G. To the best of our knowledge, this work is the first effort in this matter.

INDEX TERMS Over-the-top application, classification, incremental learning, service degradation, decision making.

I. INTRODUCTION

Data caps and monthly traffic limits are techniques used to control subscribers' data usage [1], [2]. They have emerged mainly due to the growing demands placed on the networking stack created by the continuous increase in connected users, new applications, and services. Historically, these techniques have been used by Internet Service Providers (ISP). Today, they are more commonly seen among mobile operators [3]. Australia, Canada, South Africa, the UK, and the US are just a few countries to name where data caps are regularly used to assist mobile operators [4].

A typical application of data caps is service degradation policies. Traditional mobile network operators offer user data plans with consumption limits. In such networks, service degradation is a standard mechanism used to limit the amount of information that can be transferred by the users over a specific period [2], [5]–[7]. This mechanism is usually applied when a user exceeds the predefined consumption limit to save resources and ensure the high performance of the network.

The associate editor coordinating the review of this manuscript and approving it for publication was Shagufta Henna.

Nevertheless, this degradation is applied in a generalized way, affecting the performance of all user applications.

Service degradation is heavily based on traffic analysis, especially in the investigation of Over-the-Top (OTT) media and communications services. OTT refers to applications that deliver audio, video, and other media over the Internet by leveraging the infrastructure deployed by network operators but without their involvement in its management or distribution of the content [8]. These applications are well-known for their large consumption of network resources to offer their different functionalities, generating a massive impact on the network's operation and management. While the implementation of service degradation policies seems straightforward, the analysis of users' consumption behavior is rarely considered. However, by including such analysis, operators can take well-based decisions to maintain a network performance while affecting the users' perception of the service provisioning as least as possible.

Machine Learning (ML) is regularly used in users' consumption behavior analysis [9]–[13]. However, most of the models used are based on traditional batch learning, which is well-known for several limitations. (*i*) Depending on the algorithm and the dataset's size, the training phase might take a considerable amount of time and demand high computational resources. (*ii*) When training with a small dataset with no representative number of samples, the algorithm might present a poor performance. (*iii*) Once an algorithm has been trained and implemented, it cannot acquire new knowledge from new samples. Consequently, if there is a change in the data's statistical properties (i.e., a concept drift), a new model must be generated.

Incremental Learning (IL) constitutes an alternative to traditional ML. It is built under the premise of continuous learning and adaptation. Its main advantage is that it does not require previous perception data. Therefore, it appears to be a suitable solution to address the OTT service degradation limitations where consumption behaviors can change over time in an indeterministic manner.

This paper proposes a reference model that can serve as a guideline for users' OTT consumption behavior classification based on IL. We evaluated our model's viability in a case study by investigating the efficacy of several IL algorithms on a dataset containing real-world users' OTT application consumption behavior. We rigorously assessed the NB, KNN, ARF, LB, OB, Learn++, and MLP algorithms. The obtained results show that ARF and a composition between LB and ARF achieve the best performance yielding a classification precision and recall of over 90%. Subsequently, we proposed service degradation policies based on the users' consumption classification and the 5G technical specification [14]. Our reference model offers a conceptual basis for smart personalization of service degradation policies.

The rest of this paper is organized as follows. Section 2 presents the related work in the area of user profiling. In Section 3, we describe our reference model explaining its actors, components, and workflow. Then, Section 4 presents the case study focused on the proposal of personalized service degradation policies. This section includes the experimental scenarios and the analyses that were obtained through the implementation of the reference model. Finally, in Section 5, we conclude this paper and sketch future work.

II. RELATED WORK

We performed a systematic mapping [15] of existing works to identify related research. In this mapping, we analyzed recent works published between 2015 and 2020 in four scientific databases – Scopus, ScienceDirect, IEEE Xplore, and Google scholar. Figure 1 shows the obtained map with six different research contexts and six research types. The former was self-defined while the methodology itself specified the latter (note, we did not modify the predefined types). A general observation (also evident from Figure 1) is that most works are related to the domain of the social network having a solution proposal type. No evaluation research, philosophical or experience works were identified. Hence, neither new proposals nor experiences gained, lessons learned papers were included in this study.

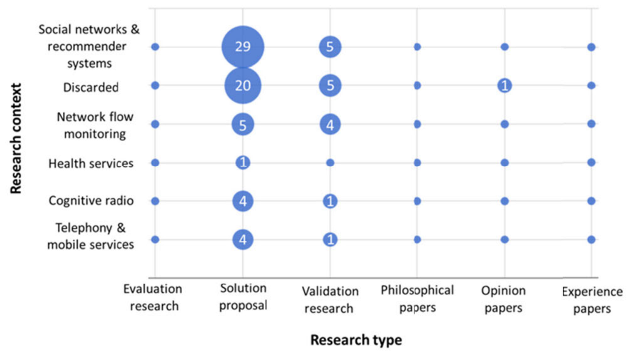


FIGURE 1. Systematic map – User profiling.

The primary focus of interest in this study is network traffic flow measurement and analysis. The obtained knowledge is used for smart user consumption profiling and service degradation policy development. Therefore, we investigated those works that were published in the domain that includes this area – information and telecommunication technologies. Additionally, selected works related to social networks and mobile services were also included in our list as we found some derived conclusions informational. In what follows next, we briefly discuss these related works.

Vinupaul *et al.* [16] utilized network flow analysis as a viable method for user identification. They introduced the concept of flow-bundle-level features to characterize and identify users. Four different ML models were validated using a dataset of flow records that contained 65 users. The authors achieved an accuracy of 83% when identifying users using the random forest algorithm.

Hess *et al.* [17] defined a profiling model that characterizes the user behavior and temporal dynamics. The spatial locations during an hour showed that roughly 80% of users visit six cells daily. Also, 22% of users did not revisit a site, and around 30% revisited the same sites between 6-10 days. Finally, holiday users showed a higher number of upload/download rates. The knowledge of such regularity in mobility can help predict high traffic loads and achieve better network management.

Rajashekar *et al.* [18] studied the extent to which phone behaviors can be associated with users. To achieve this, they used a multilayer perceptron configured to act as an autoencoder. They applied a self-organizing map (SOM) to unveil unique characterizations of user behavior. The application, cell tower, and website usage logs were used to generate user behavior profiles and isolate a single user for a specific device based on the behavior.

Bakhshandeh *et al.* [19] showed that identifying the users based on their behavior instead of IP addresses is vital since the dynamic assignation of addresses can easily avoid security mechanisms. The authors proposed a method for efficient user identification method that is based on NetFlow traffic flows. This method can extract a set of features from the network flows and use a random forest model to classify users. Their model achieved a precision of 94% in user identification. The results showed that such a method is suitable for forensic science. It does not require examining the total traffic (header with payload) for user identification and characterization.

Zhao *et al.* [20] investigated the options for inferring users' gender based on application snapshots installed on mobile devices. Their main goal was to make smartphone systems more user-friendly and provide better personalized services and products. They observed that it is possible to infer gender from usage habits. They also identified significant differences between application types, functionalities, and icon designs through an empirical study. Their prediction model achieved 76.62% accuracy.

Paul *et al.* [21] characterized the network of verified users on Twitter. Their analysis extracted verified users on Twitter and obtained 231,246 user-profiles and 79,213,811 connections. Their analysis confirmed that the sub-graph of verified users mirrors the full Twitter user graph in some aspects, such as possessing a short diameter. They also obtained interesting differences, such as the possession of a power-law out-degree distribution, slight dissortativity, and a significantly higher reciprocity rate. Finally, the authors observed stationarity in the time series of verified user activity levels that differs from traditional Twitter users.

In summary, when compared with our approach, related works discussed above are aimed at user profiling in a different context. Accordingly, they fall short of addressing the classification domain of OTT user consumption behavior. None of them provides a reference model proposal similar to ours. They neither aim to obtain an overview of the users' OTT consumption behavior to support personalized service degradation policies. Furthermore, the proposed models in related research are based on traditional ML. Lastly, related works do not offer a guideline on how to perform user profiling either.

III. REFERENCE MODEL FOR IL-BASED USER OTT CONSUMPTION CLASSIFICATION

This section presents our reference model that is depicted in Figure 2. The description of its main components, actors, and workflow is presented as follows.

A. ACTORS

We identify four actors with the following roles:

- A network user is any person that has access to a device capable of connecting to the network (e.g., smartphone, laptop, and tablet). Their main activity is network traffic generation through the consumption of OTT applications or any other activity that can be performed through the Internet.
- The network expert represents the person or staff who knows about network architecture and devices' technical activities. This actor can execute network devices' configuration, gather and summarize network information, and configure environments for network experiments.
- The data analyst is in charge of developing all the activities related to data analysis and ML models. Those activities include data cleaning and preprocessing, data transformation, clustering and pattern recognition, training, selection, and deployment of ML models.
- Finally, the network analyst is the person or group of people in charge of the decision-making process related to network maintenance, network policies application, resource allocation, and business strategies. This actor can leverage the data analyst's valuable information and the IL model to make strategic decisions. In some cases, depending on the available staff, the network analyst and network expert's activities can be handled by the same person or group. In our case study, the decision-making process is related to the proposal of personalized service degradation policies.

B. COMPONENTS

We also define three macro components. These are depicted in Figure 2. Except for the network user, each actor is responsible for one specific component. The description of the components is presented as follows.

- Data Gathering & Flow Generation comprise all the processes that must be performed by the network expert. Typically, the network expert holds all the knowledge related to network equipment and architecture. These processes (packet persistence and flow generation) involve all the activities related to configuring network devices, gathering and storing IP packets, packet aggregation for flow generation, and application labeling.
- Data Preprocessing and Model Selection comprises all the processes related to the data analyst's skills and

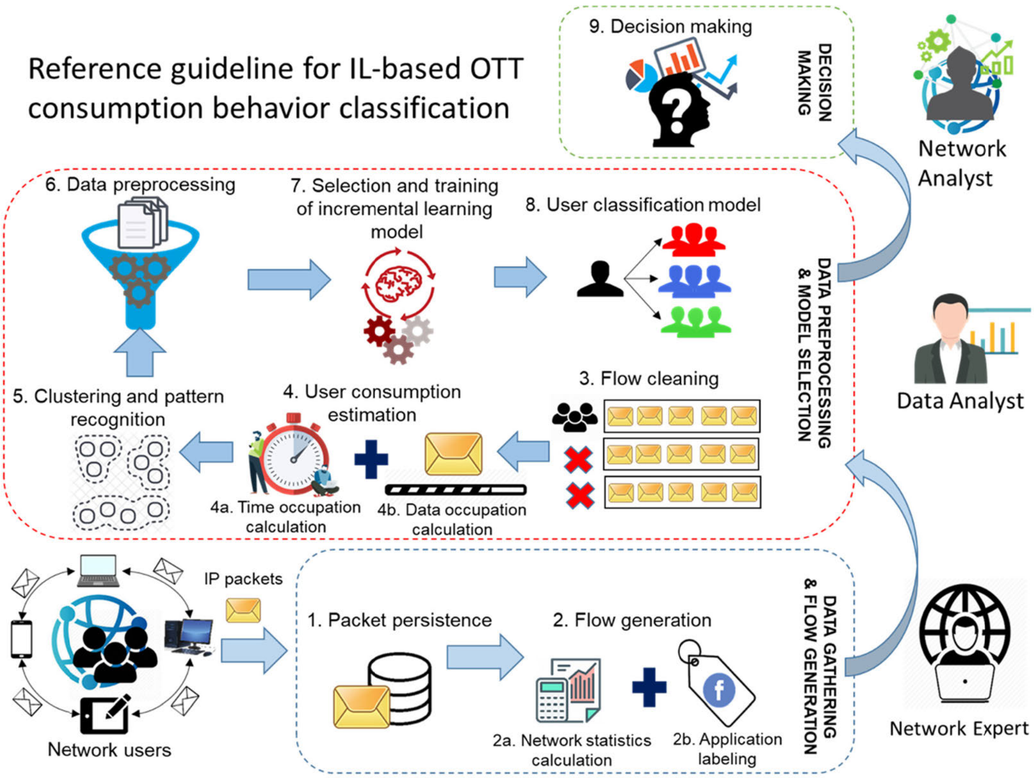


FIGURE 2. Reference model.

knowledge. All raw data preprocessing is carried out in this component. It includes flow cleaning, user consumption estimation, clustering and pattern recognition, and data cleaning. Furthermore, the data analyst is also responsible for the IL algorithms' comparison and evaluation to obtain the best model that maintain its performance consistency and knowledge retention.

- Finally, the Decision-Making involves the processes that the network analyst must perform based on their knowledge and the user information obtained from the IL model. This way, the network analyst can make better decisions related to network resource allocation, QoS policies, service pricing, and service offering. In our case study, decision-making is used for proposing service degradation policies based on users' classification according to their OTT consumption behavior.

C. WORKFLOW

IP Packets are the fundamental elements that describe the communications established by network users. Therefore, packet persistence is the *first step* in our workflow. Packet capture is usually achieved by a device that observes all the packets being transferred in the network. Once the packets are captured, they are stored in a specific format known as packet capture (PCAP) files. Depending on the traffic, the size of PCAPs can grow to several hundreds of GB. An infrastructure capable of storing large amounts of data is therefore highly recommended.

In the *second step*, flow generation is performed. Network flows are a statistical representation of the communications generated by user devices. They carry information related to the amount of transferred data and the communication duration, among many others. Flows are obtained by aggregating packets into flows [22]. Aggregation is performed based on a 5-tuple, precisely the source and destination IP addresses and port numbers, and the protocol identifier. Two tasks are required to be carried out in this context (*steps 2a and 2b*):

- Network statistics calculation: as packets belonging to traffic flows are observed, a set of statistics can be calculated to obtain a statistical representation of the communication. Common network statistics include the packet sizes, number of packets, and inter-arrival times.
- Application labeling: to obtain an overview of network users' consumption behavior, it is necessary to know the applications that generate the flows. Consequently, all the network flows are labeled with the respective application name that is being consumed. The most common approach for flow labeling is Deep Packet Inspection (DPI) [23]. Through this approach, the payload of network packets is inspected to obtain the application-specific information.

The resulting flow records with their respective application labels are typically stored in a CSV file. However, other storage formats can also be considered to store flow measurement data. CSV files are a popular storage format among the ML community.

The *third step* is concerning the flow cleaning process. In this step, unnecessary information is removed. Such unnecessary data is, for example, flows that do not belong to user devices' communications. This process usually requires a collaboration between the data analyst and the network expert. By leveraging their knowledge, the network experts can ensure that the flows to be removed are not data with high information values.

Then, in the *fourth step*, information regarding user consumption estimation is determined. This is achieved by summarizing each user's behavior, which requires performing two main tasks (steps 4a and 4b):

- Occupation time calculation: the amount of time spent by each user consuming OTT applications offers essential insights for the network operator. Knowing this aspect enables the identification of their preferred applications and the high demand for network resources. Therefore, the amount of time that the user spent consuming each application must be calculated by leveraging the network flows' time-related information.
- Occupation data calculation: similarly, a further feature that offers essential insights on users' behavior is the amount of data (bytes) exchanged by each application. As in the previous case, this aspect allows observing if the user demands significant network resources and if a different data plan might not be proper to their needs. Therefore, data occupation is also calculated for each OTT application consumed by the users by leveraging the number of bytes exchanged in the network flows.

Next, clustering and pattern recognition and data preprocessing are performed in the *fifth* and *sixth steps*. These steps aim to identify the best approach to organize the users based on their consumption behavior. This process yields a dataset labeled with the set of groups that can be used for user classification.

Subsequently, IL model selection and training are performed in the *seventh step*. IL is built on the premise of continuous learning and adaptation, enabling the autonomous incremental development of complex skills and knowledge. In ML context, IL aims to smoothly update the prediction model to account for different tasks and data distributions while still being able to re-use and retain knowledge over time. Figure 3 illustrates a typical IL workflow. IL builds its prediction capabilities iteratively on top of what was previously learned. This way, it can amend previous errors and shortcomings while efficiently adapting to new environmental conditions as new data becomes available. As a result, the systematic re-trainings typical for traditional batch learning can be avoided. Consequently, time and computational resources can be saved while maintaining the model's performance.

As shown in Figure 3, IL provides a continuous iterative process. First, after feature extraction and instance labelling applied to raw data, a consolidated dataset is obtained. This dataset is obtained via *Steps 1-6*, as shown in the reference model (see Figure 2). Then, the dataset is used to build the IL model through training and testing. It is not uncommon to see that more than one IL algorithm is used. This results in a set of models.

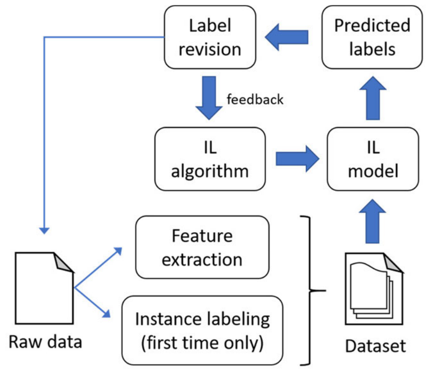


FIGURE 3. IL algorithms workflow.

Next, each algorithm's classification result is reviewed on every iteration in a label revision process, and feedback is provided to the algorithm to maintain the IL setting. This way, every algorithm is tested and compared, aiming at selecting the one showing the best classification performance.

The model capable of performing user classification based on consumption behavior is prepared for application in the *eighth step*. It maintains its adaptiveness, scalability, and efficiency. Furthermore, the model is built in a way so that it provides useful information about the users.

Once all the previous steps have been completed, the network analyst can finally take appropriate actions supporting the decision-making process in the *ninth step*. With the users' information provided by the classification model, the network analyst can develop new strategies to apply network policies, manage resource allocation, and issue data plans.

IV. PERSONALIZED SERVICE DEGRADATION – A CASE STUDY

We assess the usefulness and real-world applicability of our reference model in a case study. Specifically, we evaluated our approach using prototype-level evaluations when evolving towards proposing personalized service degradation policies. In what follows, we first describe the architecture used for both data measurement and decision-making. Then we discuss ground truth development as one of the key enablers of our approach. This is followed by the performance evaluation of IL algorithms when investigated in OTT consumption behavior classification. Finally, we elaborate on developing service degradation policies as an end goal for the decision-making process.

A. ARCHITECTURE FOR SMART DECISION MAKING

We evaluate our approach in the context of the Knowledge-Defined Networking (KDN) paradigm [24], [25].

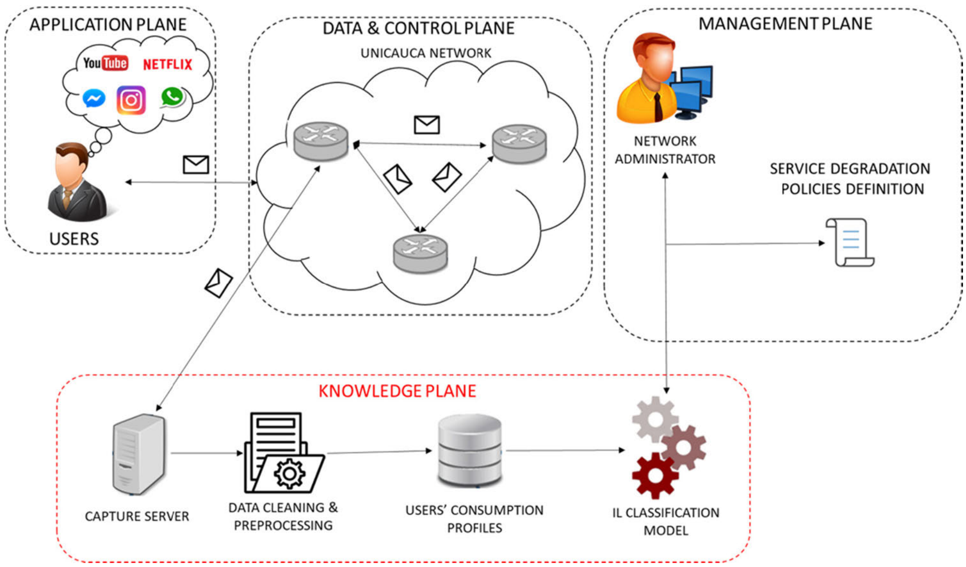


FIGURE 4. Data extraction environment.

Our architecture shown in Figure 4 aims to extend the network with a knowledge plane by gathering information relevant to IL-based users' OTT consumption behavior classification. With such an extension, we aim to support the network administrators in defining personalized service degradation policies. The description of its components is as follows:

- Application Plane: this plane contains the network users that connect through different kinds of devices and consume the OTT applications.
- Data & Control Plane: includes all the devices, network topology, exchanged data, and functional configuration set up by the network administrator.
- Knowledge Plane: contains the elements that offer valuable knowledge about the network to the administrator. The main elements include:
  - (i) The *capture server* configured to gather the network traffic containing the information about OTT consumption behavior;
  - (ii) *Data cleaning & preprocessing* where the raw data is processed and analyzed;
  - (iii) The *users' consumption profiles* dataset containing OTT consumption behavior data;
  - (iv) The *IL classification* model used for 1) classifying the users into their corresponding consumption profiles, and 2) administrator support during the definition of personalized service degradation policies.
- Management Plane: represents the plane where the network administrator makes all the decisions to define the service degradation policies based on the users' behavior analysis.

B. GROUND TRUTH DEVELOPMENT

Following *Steps 1-6* of our reference model's workflow (see Section 3.C), we systematically developed a ground truth. The tasks associated with these steps are summarized as follows.

Step 1: packet persistence. We started with capturing packet traces in the network of Universidad del Cauca, Colombia. The packets were captured using Wireshark on a server configured as a mirror port of a core network switch throughout ten days in 2019. All the packets were stored in PCAP files whose total size was roughly 3 TB.

Step 2: flow generation. Several applications can organize IP packets into flows while obtaining flow statistics. However, the majority of the widely recognized tools cannot perform layer seven (application) labeling. We developed our application, named Flow Labeler [26], to perform network statistics calculations and application labeling to overcome this challenge. Flow Labeler is based on FlowRecorder [27] and NFStream [28] network traffic flow metering tools. It follows the architectural principles of the traditional network traffic measurement paradigm [22]. Using Flow Labeler, we processed all the captured PCAP files. Eventually, we obtained a dataset containing 2,704,839 records with 50 attributes and 141 application labels. The features measured by Flow Labeler are shown in Table 1.

Step 3: flow cleaning. During flow cleaning, we first discarded the flow records that were not generated by user devices. This was achieved by a filter that retained only those flows that belonged to a specific range of IP addresses holding user devices. Then a selection process was applied to preserve only the most representative OTT applications. The original dataset contained flows generated by 141 different applications. After the selection process, flows belonging to 56 OTT applications were retained. These flows were generated by popular applications such as Facebook, Netflix, Spotify, Twitter, Dropbox, WhatsApp, and YouTube.

Step 4: user consumption estimation. After flow cleaning, we proceeded with user consumption estimation.

TABLE 1. Attributes generated by flow labeler.

Feature name	*Feature description*	*Direction*		
		FWD	**BWD**	**Bidirectional**
flow_key	Flow identifier through a hash algorithm	N.A.	N.A.	✓
src_ip_numeric	Source IP in decimal format	N.A.	N.A.	✓
src_ip	Source IP in network format	N.A.	N.A.	✓
src_port	Source port number	N.A.	N.A.	✓
dst_ip	Destination IP in network format	N.A.	N.A.	✓
dst_port	Destination port number	N.A.	N.A.	✓
proto	Transport protocol number according to IANA (e.g., 1 for ICMP, 6 for TCP, 17 for UDP)	N.A.	N.A.	✓
pktTotalCount	Total number of packets in both directions	✓	✓	✓
octetTotalCount	Total of bytes exchanged focusing on the IP payload only	✓	✓	✓
min_ps	Minimum packet size on the flow	✓	✓	✓
max_ps	Maximum packet size on the flow	✓	✓	✓
avg_ps	Average packet size on the flow	✓	✓	✓
std_dev_ps	Packet size standard deviation	✓	✓	✓
flowStart	Flow start time in seconds using UNIX time format	✓	✓	✓
flowEnd	Flow end time in seconds using UNIX time format	✓	✓	✓
flowDuration	Total flow duration in seconds using UNIX time format	✓	✓	✓
min_piat	Minimum packet interarrival time on the flow	✓	✓	✓
max_piat	Maximum packet interarrival time on the flow	✓	✓	✓
avg_piat	Average packet interarrival time on the flow	✓	✓	✓
std_dev_piat	Packet interarrival time standard deviation	✓	✓	✓
flowEndReason	The reason why the flow was expired and sent to the final array that will be converted to CSV file - 0 inactive timeout expired - 1 active timeout expired - 2 forced expiration due to end of pcap file or live captured stopped - 3 FIN flag detected on both directions - 4 RST flag detected - 5 FIN Flag detected on one direction only and timer expired	N.A.	N.A.	✓
category	Category of the communication as delivered by nDPI	N.A.	N.A.	✓
application_protocol	Application protocol for the flow (e.g., TLS, HTTP, DNS, etc) detected by nDPI	N.A.	N.A.	✓
web_service	Web service detected by nDPI (e.g., Facebook, WhatsApp, Google, etc)	N.A.	N.A.	✓

We considered only the attributes directly related to users' consumption behavior. Eventually, OTT service consumption was measured using the amount of time a user spent consuming the application and the amount of data exchanged through the network. This led to a dataset with 716 OTT consumption profiles with 114 attributes for each instance, whose information is provided in Table 2.

Step 5: clustering and pattern recognition. Subsequently, since the users' data remained unlabeled, we performed clustering to classify the users. First, we focused on finding the optimal number of clusters for the data. We determined the ideal number of clusters for the dataset supported by the well-known elbow and average silhouette methods [29].

In the elbow method, the bend (knee) location in the plot is generally considered an indicator of the appropriate number of clusters. Whereas in the silhouette method, the location of the maximum is considered as the appropriate number of

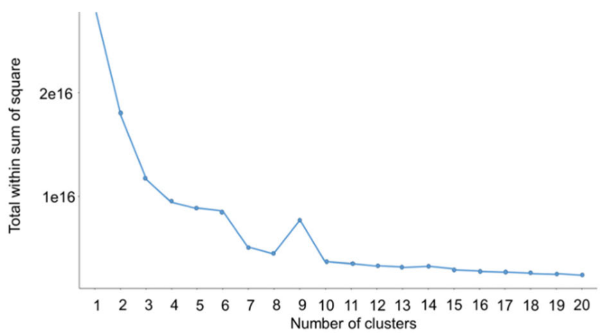


FIGURE 5. Elbow method - the curve of within-cluster sum of square according to the number of clusters *k*.

clusters. Figures 5 and 6 show the plots obtained with the two methods. Based on the figures, the optimal number of clusters in our dataset appears to be 4 and 7.

TABLE 2. Users' consumption dataset – 114 attributes.

Attribute name	Description
src_ip_numeric - 1 attribute	This attribute holds the users' IP in decimal format. It is mainly used as a user identifier.
ApplicationName_time_occupation - 56 attributes	These attributes hold all the information related to the amount of time the user spent using the application measured in seconds.
ApplicationName_data_occupation - 56 attributes	These attributes hold all the information related to the number of bytes the user exchanged while using each application.
User group - 1 attribute	This attribute represents the target class of the dataset i.e., the users that are classified in three different groups (high, medium and low consumption).

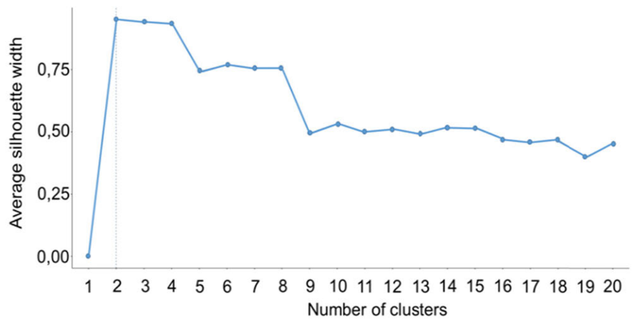


FIGURE 6. Silhouette method – the curve of average silhouette according to the number of clusters k.

With these two values determined as the optimal number of clusters, we proceeded with the clustering process. In our rigorous process, we analyzed our dataset using several clustering algorithms. Specifically, we used Cobweb, Hierarchical (connectivity-based algorithms), Expectation-Maximization (Gaussian mixture), DBSCAN (density-based), and K-means (centroid-based) clustering [30], [31].

Among all the algorithms tested, K-Means yielded the most reasonable results regarding the division of instances. Specifically, we achieved the best results with 4 clusters. After analyzing the centroids, the distributions, and the total time and data occupation, we eventually developed a dataset whose samples were divided into (labeled with) four classes, as shown in Table 3. Since data occupation is considered the most critical feature, the group of users who occupied more data was labeled as the highest consumption users.

Finally, by implementing a correlation study between the dataset attributes and the target classes, the most correlated attributes were determined. Specifically, we determined that AppleIcloud data occupation and Gmail time occupation exhibited the highest correlations. Although the applications are different and several examples can be observed through different correlation, we focused mostly on the target class for visualization purposes. Consequently, in Figure 7, a plot of these two attributes illustrates the cluster distribution.

TABLE 3. Clusters labeling.

Cluster	Total time occupation	Total data occupation	Number of users	Label
Cluster 1	136,534 hours	3,135 Mb	50 users	Medium consumption
Cluster 2	1275,462 hours	11,2 Mb	1 user	Very high consumption
Cluster 3	18,46 hours	2,242 Mb	582 users	Low consumption
Cluster 4	36,545 hours	3,482 Mb	83 users	High consumption

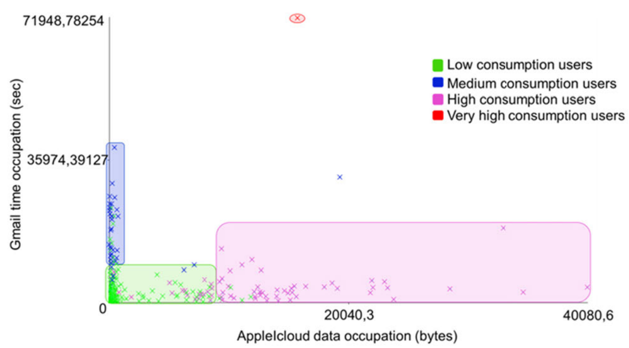


FIGURE 7. Cluster visualization.

A common observation is that medium consumption users spend more time with consuming data while maintaining a low data occupation rate. High consumption users present higher data occupations in shorter periods. Furthermore, there is only one very high consumption user that exceeds all the other clusters. This user exhibits the highest data and time occupations. However, considering the clusters' behavior, this case could be considered as an outlier in the data.

Step 6: data preprocessing. In data preprocessing, we focused on two goals: (i) to remove anomalies and (ii) to improve the class balance. To achieve this, we applied the Inter Quartile Range (IQR) [32] approach to detect and remove the outliers in the dataset. To improve the class balance, we used the SMOTE algorithm [33]. As a result, we obtained a dataset with 1,249 instances (users). The classes were distributed based on their consumption leading to the following statistics:

- 510 low consumption users,
- 333 medium consumption users, and
- 406 high consumption users.

C. EVALUATION OF IL ALGORITHMS

KDN is a commonly used paradigm in network traffic management. One of its disadvantages is poor efficiency sourcing mainly from its utilization of traditional ML.

However, the OTT applications market is highly volatile and unpredictable (e.g., a wide variety of applications, constant new trends, and technologies). These constraints must be addressed through an approach that can adapt to the dynamic aspects of the network traffic and users' behavior changes. That being said, our goal was to compare the IL algorithms' classification performance in the domain of user OTT consumption behavior. To achieve this, we examined several IL algorithms' performance using the dataset described in Section 4.B.

1) OVERVIEW OF THE ALGORITHMS

The most commonly used IL algorithms [34]–[39] include decision trees, rule-based systems, NB, KNN, SVM, and neural networks. Other common alternatives represent ensemble methods such as OB, ARF, LB, and Learn++. We evaluated the Python scikit-multiflow [40] implementation of these algorithms. The main characteristics of the algorithms are shown in Table 4 as per [41]. Furthermore, their brief description is as follows.

The Hoeffding Tree (HT) or Very Fast Decision Tree (VFDT) [42] is an incremental decision tree induction algorithm. It exploits the fact that a small sample of data can often be enough to choose an optimal splitting attribute for the construction of the decision tree. This idea is supported mathematically by the Hoeffding bound, which gives a level of confidence that allows choosing the best attribute to split the tree [43].

The NB algorithm's incremental adaptation is a classifier known for its simplicity and low computational cost. This algorithm performs the classification process while applying the Bayes theorem, where a strong (naive) assumption is made (all the input data features are independent of each other).

The KNN algorithm, in an incremental setting, works by keeping track of a fixed number of training samples of the last window of observed samples.Then, whenever new input data is received, the algorithm searches within these stored samples and finds the closest neighbors using a selected distance metric (e.g., Euclidean distance). While maintaining low search times, scikit-multiflow uses a structure called a K Dimensional Tree to store the samples.

OB [44] is an incremental ensemble learning method that improves the traditional Bagging from the batch setting. It simulates the training phase by taking each arriving sample to train the base estimator over k times. This sample is drawn by a binomial distribution – a method that can be considered as "a good drawing with replacement" substitution from the one implemented in the traditional batch learning.

LB [45] is based on OB. It tries to obtain better results by modifying the Poisson distribution parameters obtained from the binomial distribution when assuming an infinite input data stream. This Poisson distribution is used to perform the drawing with replacement to reduce zero values in the distribution's mass probability function. This is achieved by increasing the value of λ from 1 to 6.

ARF [46] is an adaptation of the traditional random forest algorithm applied to the incremental learning scope. ARF generates multiple decision trees and decides how to classify the input data through a weighted voting system. Within the voting system, the individual tree with the best performance (in terms of accuracy or the Kappa statistic) has a more substantial weight in the votes, i.e., higher priority in the decision.

Learn++ [47] is an ensemble method inspired by the Adaptive Boosting (AdaBoost) algorithm. It was originally developed to improve the classification performance of weak classifiers. In essence, Learn++ generates an ensemble of weak classifiers, each trained using a different distribution of training samples. The outputs of these classifiers are then combined using a majority-voting scheme to obtain the final classification.

The MLP algorithm [48], [49], is a neural network with three or more layers. It is used to classify data that cannot be separated linearly. It is a type of artificial neural network that is fully connected. Every node in a layer is connected to each node in the following layer. It uses a nonlinear activation function (mainly hyperbolic tangent or logistic function).

Several conclusions can be drawn from Table 4. All algorithms have advantages and disadvantages that must be taken into consideration during selection. Neural Networks, for example, can handle noise but need a considerable amount of data for training to offer good performance. Besides, its implementation lacks a systematic approach making the approach mostly empirical. Lazy methods like KNN require an appropriate number of neighbors to be determined. However, once this is obtained, it usually offers good performance on both regression and classification tasks. Decision trees are easy to interpret but are quite sensitive when there is a large number of attributes to handle. SVMs have strong mathematical foundations, but its training and hyperplane calculations can take a considerable amount of time when there is a large

TABLE 4. General overview of common IL algorithms [41].

Feature/algorithm	*Decision trees (VFDT, ARF)*	*SVM*	*Neural Networks (MLP)*	*Lazy methods (KNN)*	*Ensemble methods (OB, LB, Learn++)*	*Naïve Bayes*
Type of input parameters	Numeric					
Parameters considered for the classification	Selection of the most discriminant attribute	The zone of influence in the hyperplane border	The rule combination of classifiers	The number of nearest neighbors	Classification is determined by a weighted voting system	Estimated probabilities for each target class
Type of classification	Strict					
Advantages	Simplicity in comprehension and interpretation	Strong mathematical foundations	Resistance to noise in the data	Versatile algorithm for classification and regression	Allows the composition of classifiers improving their individual performance	Fast and simple classification based on a statistical approach
Disadvantages	Depending on the number of attributes its representation can be difficult	Large calculation time of hyperplanes	The absence of a systematic method for its implementation	Prediction might be slow if the number of neighbors is large	Depending on the composition the prediction time might be large.	Naïve assumption of correlation independence between input features

amount of data. Naïve Bayes is the fastest approach. Though, it can present bad results due to its assumption of correlation independence among the features. Finally, ensemble methods offer an advanced approach by improving the performance of individual algorithms through their combination. However, it is crucial to choose a composition that is not too complex. Otherwise, its training can take a large amount of time and require several computational resources.

In conclusion, when selecting the best algorithm for a specific goal, several algorithms should be taken into consideration, tested and evaluated (compared).

2) EVALUATION OF CLASSIFICATION ACCURACY

Considering the available evaluation approaches that can be considered for IL [50], we first focused on analyzing the algorithms' accuracy and learning rate when performing classification. Each algorithm's performance in terms of averaged success or error rate can be analyzed through accuracy measurements. These measurements include precision, recall, and the Kappa statistic [51], [52]. As the dataset does not hold many instances, the storage and computational requirements are not considered an evaluation metric for the current scenarios. As shown in Figure 8, a set of prequential evaluations was carried out for each algorithm changing the warm-up data sample size between 10% (125 instances), 15% (187 instances), 20% (250 instances), and 50% (625 instances) of the original dataset.

The grace period (i.e., the number of instances a leaf should wait before doing a split) for the decision trees (HT and ARF) was set to 100 instances. The split criterion was based on the information gain. Furthermore, ten estimators were selected for the ensemble methods (OB, LB, Learn++, and ARF) [53]. Finally, a series of tests were carried out to obtain the best results for MLP.

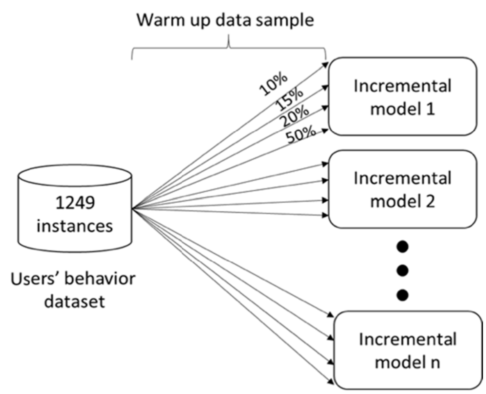


FIGURE 8. Configuration for the precision, recall and Kappa statistics evaluation.

Figure 9 depicts the obtained results for the algorithms tested in terms of precision, recall, and the Kappa statistic with a warm-up of 10%. In comparison, Figure 10 shows the achieved results with a warm-up of 15%. Similarly, Figures 11 and 12 show the obtained results for warm-ups with samples of 20% and 50% of the original dataset.

From the results, several conclusions can be derived. In general, in all tests, we observed similar performance behaviors. We noticed that the NB, HT, KNN algorithm and the ensemble methods used as base estimators in the composition achieved the worst performance.

For NB, the obtained performance could be due to its (naive) assumption of statistical independence between the input data attributes. Since the time and data consumption of the same OTT applications are related, assuming all the attributes are independent could increase the error rate throughout the classification process. For the classification process, the KNN algorithm maintains a small window of previously seen instances (previous perception data). It looks

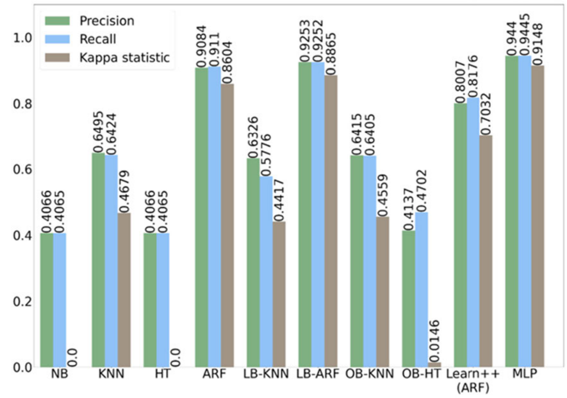


FIGURE 9. Results – Warm-up with 10%.

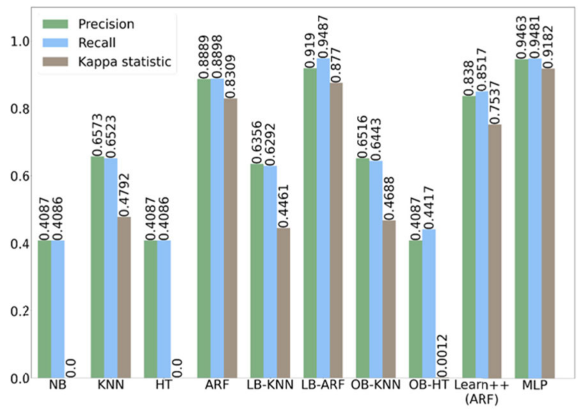


FIGURE 10. Results – Warm-up with 15%.

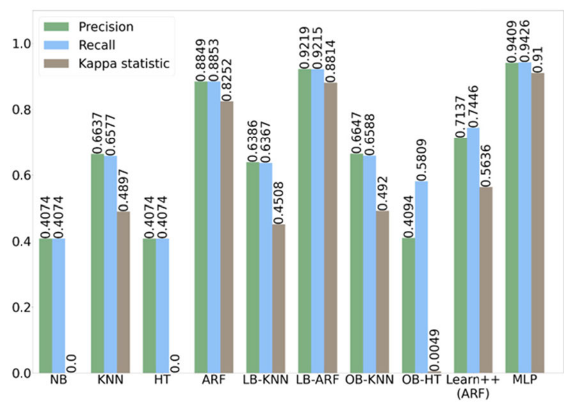


FIGURE 11. Results – Warm-up with 20%.

among them for the nearest neighbors to the current input. Consequently, the data within the stored window likely does not provide a correct classification for the input. It can also be the case for the ensemble methods that use KNN as the base estimator (LB and OB). The ensemble methods are composed of 10 KNN estimators, while each estimator stores a data window. Therefore, the ensemble method's performance becomes compromised if such a window does not

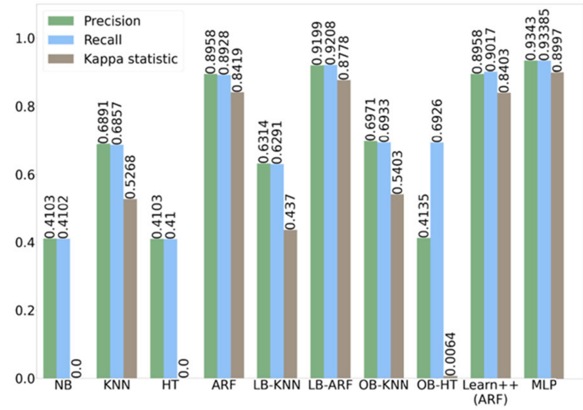


FIGURE 12. Results – Warm-up with 50%.

provide a correct estimator classification. As a result, a wrong classification for the input into the voting process is selected.

Furthermore, the HT algorithm and the OB with the HT as a base estimator exhibited poor performance. This algorithm builds a decision tree while exploiting the following assumptions. (*i*) A small sample can often be enough to choose an optimal splitting attribute. (*ii*) The distribution that generates the data does not change over time. These assumptions make the algorithm efficient in terms of computational resources. However, in some instances, they can be a disadvantage as once a node is created in the tree, it cannot be changed anymore [54]. Therefore, once the users present different consumption behaviors, the performance might not achieve the best results since the tree had a warm-up with a small sample, and the nodes built will be unable to classify the subsequent data inputs correctly.

On the other hand, we observed that four algorithms (ARF, LB-ARF, Learn++ with ARF, and MLP) achieved an excellent overall performance. ARF's excellent performance is likely due to the multiple decision trees that this approach generates and the weighted voting system used for classification. Furthermore, this algorithm also has a warning- and drift-detection method. The warning-detection method tries to detect when possible concept drift occurs in the target class. For the other method, if drift is detected, the algorithm starts training another decision tree besides building the main ensemble method composition. Once the drift-detection method confirms a concept drift in one of the individual decision trees, it is replaced with the tree trained in the background to keep the classification accuracy. The ARF algorithm was used as a base estimator for Learn++ and LB in order to observe if its excellent individual performance could be further improved through ensemble methods. Such improvement was achieved with LB, however, not with Learn++.

Finally, MLP also achieved an excellent performance, especially after performing several experimental setups to help find the optimal network structure and an activation function. We altered the number of hidden layers, the number of nodes, and the activation function as per [55]. The best results were obtained with a 5-layer network (including the input and output layers). It means we could ensure a fast warm-up stage and low computational resource use. The number of nodes in each layer was set between 1 and 113. The best results were obtained with 113 nodes for the first hidden layer and 40 nodes for the two subsequent layers (113 × 40 × 40). As an activation function, we considered ReLU, hyperbolic tangent, sigmoid, and identity functions. The best results were achieved using the hyperbolic tangent function. Finally, we selected Adam as the optimization algorithm. It was shown to be an efficient extension of stochastic gradient descent [56].

In conclusion, the best results were achieved using the ARF, LB-ARF, and MLP algorithms. This is also evident in Figures 9–12. These algorithms appeared to be suitable for user OTT application consumption behavior classification. However, the transfer of these approaches into real-world applications depends on how well they maintain their accuracy whenever a user consumption change occurs. Therefore, we examined the classification performance maintenance of these algorithms, which will be discussed in the following section.

3) EVALUATION OF CLASSIFICATION PERFORMANCE MAINTENANCE

We assessed how well the algorithms determined in the previous section maintain their classification performance. We also assessed how their learning rates are affected when a change occurs in users' OTT consumption behavior. First, the algorithms were pre-trained using the entire dataset. Then, we investigated their performance maintenance capabilities through a prequential evaluation by systematically modifying the behaviors of 60 users. This was to simulate a real-world implementation use case where user behavior changes are likely to happen over time. The introduced changes are as follows:

- 20 Low Consumption users had their behavior changed (10 users to Medium Consumption and 10 to High Consumption),
- 20 Medium consumption behaviors were changed (10 users to Low Consumption and 10 to High Consumption), and
- 20 High Consumption users had their behaviors changed (10 users to Low Consumption and 10 users to Medium Consumption).

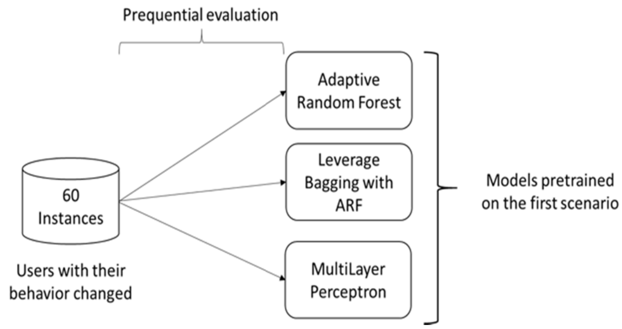


FIGURE 13. Classification performance maintenance setting.

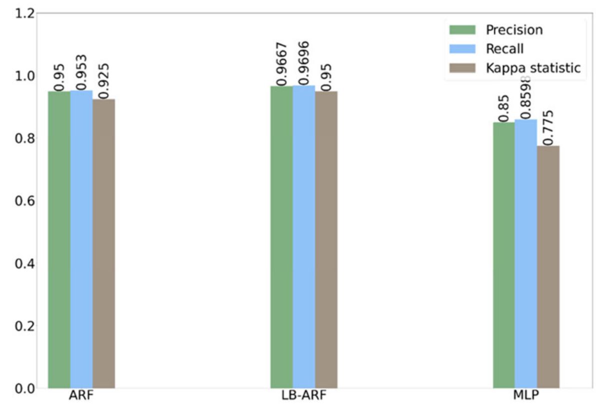


FIGURE 14. Achieved results the precision, recall, and Kappa statistics with changing samples.

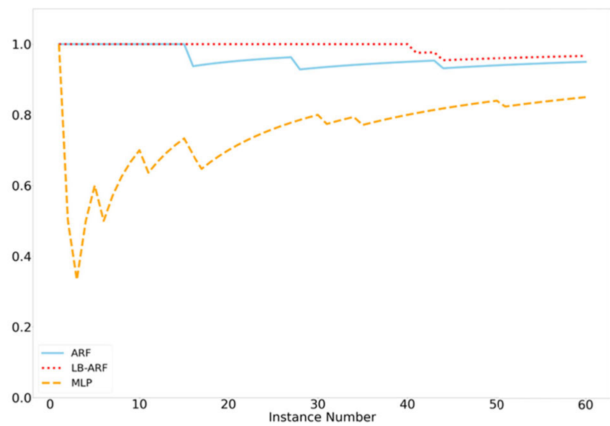


FIGURE 15. Precision evolution.

Figure 13 depicts the configuration for this evaluation. The obtained results are shown in Figure 14, while each algorithm's precision evolution is depicted in Figure 15. From the figures, we can observe that all the three algorithms presented an excellent overall performance. However, as Figure 16 shows, MLP is affected considerably by the changes introduced in the users' behavior at the beginning of the data stream. While the results eventually improve, the adaptation to the changes is slow. On the other hand, both ARF and LB with ARF can maintain their performance throughout the experiment, as shown in Figure 16.

Considering the above mentioned, we can conclude that both ARF and the composition between LB and ARF are suitable for performing user OTT application consumption behavior classification. More importantly, these algorithms can provide useful information, even in the event of dynamic user behavior change. Network administrators could make fair use of such an approach. IL can make

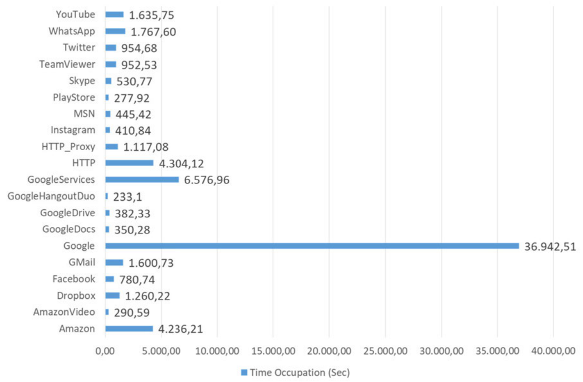


FIGURE 16. Time occupation (sec) – Low consumption users.

network traffic-related managerial tasks faster, more robust, and resource-efficient.

D. DECISION MAKING

Our case study's final step was the applicability evaluation of our reference model in a specific scenario. For this purpose, we have selected decision-making as a process focused on the recommendation of a set of personalized service degradation policies. The policies were developed based on the users' OTT consumption behavior classified using IL. Since service degradation is majorly implemented in mobile networks, the recommendation was developed in the Policy and Charging Control (PCC) architecture scope. Specifically, the concept of a 5G network PCC rule [14] was taken as the basis for service degradation.

The elements considered in the structure of the PCC rule are presented in Table 5. PCC rules can be of two types: predefined or dynamic. Considering the nature of service degradation, we proposed a dynamic PCC rule since it is activated by a specified event (e.g., a user exceeds the consumption limit).

In the 3GPP technical specification [14], the ARP is defined by the 5QI parameter. It logically follows that if a QoS flow has a GBR (5QI 1 to 4) or a non-GBR (5QI 5 to 9), there are only two possibilities for defining a service degradation policy (see Table 5). Either (i) degrade the service by affecting the Maximum Bit Rate (MBR) associated with the SDF or (ii) block the service by modifying the Gate status parameter.

Considering that 56 OTT applications were identified in the dataset, we focused on identifying which applications were most commonly consumed in each user group. With this in mind, the 20 most used applications were identified in time and data occupation. Figures 16-18 illustrate the 20 applications most commonly used in terms of time occupation for low, medium, and high consumption users, respectively. Similarly, Figures 19-21 illustrate the 20 most used applications in data occupation for the three groups.

Essentially, a larger time occupation does not imply higher data occupation. Therefore, our recommendation considers that data occupation is the main indicator for determining how much the users' consumption demands network resources.

Figures 16-21 led to the following observations. Generally, the applications that demanded the most data occupation from highest to lowest were HTTP (browsing), GoogleDocs, Youtube, Gmail, Google, GoogleDrive, and AmazonVideo (7 applications). The Google application was used most of the time by the low consumption users, with 10.26 hours (see Figure 16). For the medium consumption users, Google remained the application where users spent the most time with 77.19 hours (see Figure 17). High consumption users exhibited a large amount of time spent on Google with 18.52 hours, as shown in Figure 18. Low consumption users exhibited their largest data occupation on HTTP and Google

TABLE 5. General overview of common IL algorithms.

Element	Definition
Policy rule name	This is the name given to the PCC rule in order to be easily identified.
SDF template	This is the packet filter pre-configured by network operators in accordance with their policy, and each of them typically consists of a 5-tuple (Source IP address, Destination IP address, Source port number, Destination port number, and Protocol ID). The Protocol ID is checked with DPI.
5G QoS Identifier (5QI)	It's an integer that indicates different QoS performance characteristics of each IP packet. QCI values are standardized to reference specific QoS characteristics, and each QCI contains standardized performance characteristics (values), such as resource type (GBR or non-GBR), priority, packet delay budget (allowed packet delay shown in values ranging from 50 ms to 300 ms), Packet Error Loss Rate (allowed packet loss shown in values from 10^{-2} to 10^{-6}).
GBR & Non-GBR (Guaranteed Bit Rate)	For a QoS flow, having a GBR resource type means the bandwidth of the bearer is guaranteed, i.e., a GBR type QoS flow has a "guaranteed bit rate" associated as one of its QoS parameters. The 5QI of a GBR type QoS flow can range from 1 to 4. On the other hand, having a non-GBR resource type means that the QoS flow is a best effort type flow and its bandwidth is not guaranteed. The 5QI of a non-GBR type QoS flow can range from 5 to 9.
MBR (Maximum Bit Rate)	This parameter indicates the maximum bit rate allowed in the 5G network. Any packets arriving at the QoS flow exceeding the specified MBR will be discarded.
ARP (Allocation and Retention Priority)	An integer ranging from 1 to 15, with 1 being the highest level of priority. When a new QoS flow is needed in a 5G network with insufficient resources, the SMF decides, based on the ARP, whether to: remove the existing QoS flow and create a new one (e.g. removing an QoS flow with low priority ARP to create one with high priority ARP) or refuse to create a new one.
Gate status	The gate status indicates whether the SDF, detected by the SDF template, may pass (gate is open) or shall be discarded (gate is closed).

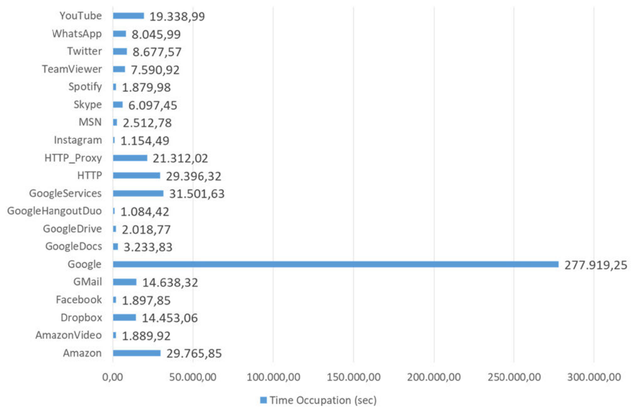


FIGURE 17. Time occupation (sec) – Medium consumption users.

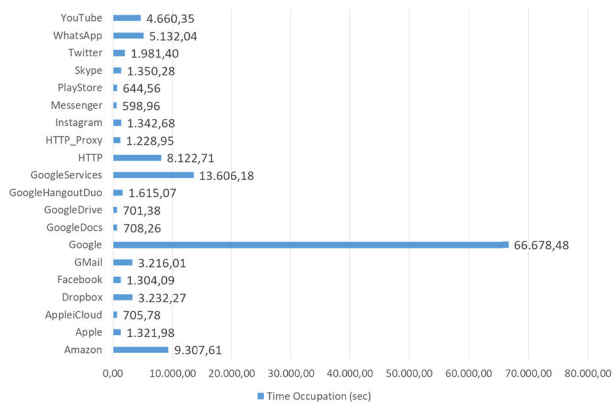


FIGURE 18. Time occupation (sec) – High consumption users.

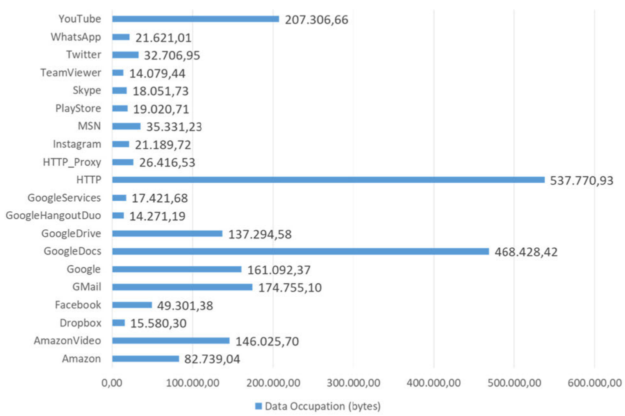


FIGURE 19. Data occupation (bytes) – Low consumption users.

docs (see Figure 19). Figure 20 shows that applications that required the most network resources were Youtube, GoogleDocs, Gmail, GoogleDrive, GoogleHangoutDuo, Spotify, Twitter, and AmazonVideo (8 applications). In terms of data occupation (see Figure 21), these users consumed the most resources using Youtube, GoogleDrive, GoogleHangoutDuo,

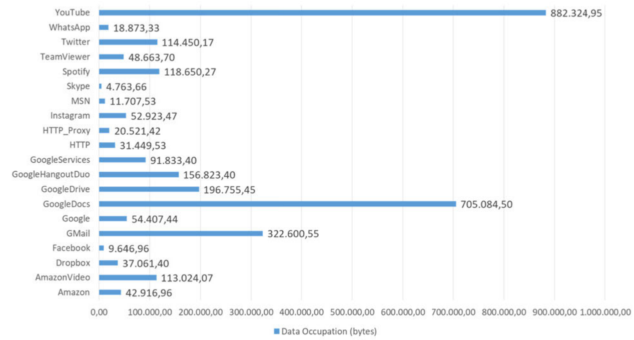


FIGURE 20. Data occupation (bytes) – Medium consumption users.

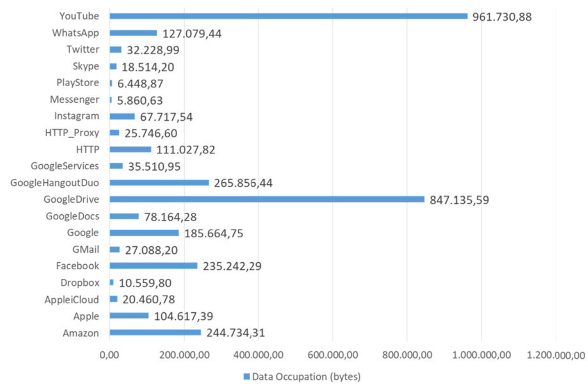


FIGURE 21. Data occupation (bytes) – High consumption users.

Amazon, Facebook, Google, WhatsApp, and HTTP (8 applications).

Considering the above observations, when a user exceeds their allowed consumption limit, we recommend that all user groups degrade the most commonly used applications. In other terms, when a service degradation status is identified, the MBR parameter is modified for the most commonly used applications, while the rest are blocked by closing the Gate Status. This way, the user can keep using the applications that are consumed most frequently. Simultaneously, the network operator can save network resources by blocking the rest of the applications not accessed by the user so frequently. To incorporate fairness in our proposal, we also take into consideration the consumption profiles of the users that exceed the limits. In consequence, the personalized service degradation policies per each group are defined as follows:

- Low consumption group – HTTP, GoogleDocs, Youtube, Gmail, Google, GoogleDrive, and AmazonVideo will be degraded while the rest will be blocked.
- Medium consumption group – Youtube, GoogleDocs, Gmail, GoogleDrive, GoogleHangoutDuo, Spotify, Twitter, and AmazonVideo will be degraded while the rest will be blocked.
- High consumption group – Youtube, GoogleDrive, GoogleHangoutDuo, Amazon, Facebook, Google,

WhatsApp, and HTTP will be degraded while the rest will be blocked.

Table 6 shows a summary of our recommendations for the set of personalized service degradation policies separated per each group of users.

TABLE 6. Policies recommendation – DS (degrade service) - BS (block service).

Applications/Clusters	Low consumption		Medium consumption		High consumption	
	DS	BS	DS	BS	DS	BS
Amazon		X		X	X	
Amazon Video	X		X			X
Apple		X		X		X
Apple Icloud		X		X		X
Dropbox		X		X		X
Facebook		X		X	X	
Gmail	X		X			X
Google	X			X	X	
Google Docs	X		X			X
Google Drive	X		X		X	
Google Hangout Duo		X	X		X	
Google Services		X		X		X
HTTP	X			X	X	
HTTP_Proxy		X		X		X
Instagram		X		X		X
Messenger		X		X		X
Skype		X		X		X
Spotify		X	X			X
Teamviewer		X		X		X
Twitter		X	X			X
WhatsApp		X		X	X	
Youtube	X		X		X	

E. SUMMARY

This case study was aimed at demonstrating the applicability of our reference model when assisting network operators in critical decision-making processes. The network administrators execute decisions based on the model that guides them through several tasks and processes. Following the model, they eventually achieve the ultimate goal – maintaining user experience and reliable network performance. The defined representation model helped to comprehend users' OTT consumption behavior and ensure that data preprocessing requirements are met. The decision-making capabilities of the architecture were shown to be efficient. The capabilities and applicability of our approach are far beyond this case study. Several components, tasks, and processes can be customized and modified to suit the network, services, and user needs. As a result, our model highly flexible, adaptable, and replicable.

V. CONCLUSION

Service degradation is a standard method to address unfair consumption behaviors. Unfortunately, they lack configuration on a finer granularity. Consequently, degradation is applied globally instead of on a user-by-user basis. As a result, consumption patterns and habits are rarely incorporated into the decision-making process. With this work, we aim to address this limitation.

We proposed a reference model and showed that it can help to achieve smart service degradation. It enables the incorporation of user consumption behavior when defining and applying service degradation policies. The proposed model provides a guideline for users' OTT consumption behavior classification based on IL. We defined its main components, actors, and workflow. To the best of our knowledge, this work is the first effort in this matter. It can be regarded as a guideline for understanding OTT applications and their demands on network technologies to propose more efficient solutions. As part of our case study evaluation, we created two datasets. We publish them both via [57], [58] to support the reproducibility of research in the domain.

In future work, we plan to automatize the definition process of personalized service degradation policies. We also plan to implement ARF and LB/ARF in a real-world network monitoring architecture. Furthermore, our plans also include tests focused on incremental Support Vector Machines.

ACKNOWLEDGMENT

The authors would like to thank Universidad del Cauca for supporting this research.

REFERENCES

[1] J. K. MacKie-Mason and H. R. Varian, "Some FAQs about usage-based pricing," *Comput. Netw. ISDN Syst.*, vol. 28, no. 1, pp. 257–265, Dec. 1995, doi: 10.1016/0169-7552(95)00096-1.

[2] J. Krämer and L. Wiewiorra, "Data caps and two-sided pricing: Evaluating managed service business models," presented at the Eur. Conf. Inf. Syst. (ECIS), Tel Aviv, Israel, 2014. [Online]. Available: https://aisel.aisnet.org/ecis2014/proceedings/track10/10/.

[3] J. Wortham, "As mobile networks speed up, data gets capped," New York Times, Aug. 14, 2011.

[4] M. Lasar. (Apr. 4, 2011). It could be worse: Data caps around the world. Ars Technica. [Online]. Available: https://arstechnica.com/tech-policy/news/2011/04/how-internet-users-are-disciplined-around-the-world.ars.

[5] M. Chetty, R. Banks, A. J. Brush, J. Donner, and R. Grinter, "You'Re capped: Understanding the effects of bandwidth caps on broadband use in the home," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2012, pp. 3021–3030, doi: 10.1145/2207676.2208714.

[6] M. Chetty, H. Kim, S. Sundaresan, S. Burnett, N. Feamster, and W. K. Edwards, "uCap: An Internet data management tool for the home," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 3093–3102, doi: 10.1145/2702123.2702218.

[7] Ixia. (2013). *Quality of Service (QoS) and Policy Management in Mobile Data Networks*. [Online]. Available: https://www.celemetrix.co.nz/assets/images/Ixia%20QoS.pdf

[8] T. Sudtasan and H. Mitomo, "Effects of OTT services on consumer's willingness to pay for optical fiber broadband connection in Thailand," presented at the 27th Eur. Regional Conf. Int. Telecommun. Soc. (ITS), Evol. North-South Telecommun. Divide, Role Europe, Cambridge, U.K., 2016. [Online]. Available: https://www.econstor.eu/handle/10419/148709

[9] S. Ponmaniraj, R. Rashmi, and M. V. Anand, "IDS based network security architecture with TCP/IP parameters using machine learning," in *Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2018, pp. 111–114, doi: 10.1109/GUCON.2018.8674974.

[10] L. Zhang, X. Meng, and H. Zhou, "Network fault diagnosis using hierarchical SVMs based on kernel method," in *Proc. 2nd Int. Workshop Knowl. Discovery Data Mining*, Jan. 2009, pp. 753–756, doi: 10.1109/WKDD.2009.79.

[11] S. Troia, D. E. Martinez, I. Martin, L. M. M. Zorello, G. Maier, J. A. Hernandez, O. G. de Dios, M. Garrich, J. L. Romero-Gazquez, F.-J. Moreno-Muro, P. P. Marino, and R. Casellas, "Machine learning-assisted planning and provisioning for SDN/NFV-enabled metropolitan networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 438–442, doi: 10.1109/EuCNC.2019.8801956.

[12] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 2451–2455, doi: 10.1109/CompComm.2016.7925139.

[13] J. S. Rojas, Á. R. Gallón, and J. C. Corrales, "Personalized service degradation policies on OTT applications based on the consumption behavior of users," in *Computational Science and Its Applications—ICCSA* (Lecture Notes in Computer Science), vol. 10962. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-95168-3_37.

[14] *Policy and Charging Control Framework for the 5G System (5GS)*, document 123.503, 3GPP Technical Specification (TS), Sep. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123503/16.05.01_60/ts_123503v160501p.pdf

[15] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, Swinton, U.K., 2008, pp. 68–77. [Online]. Available: http://dl.acm.org/citation.cfm?id=2227115.2227123

[16] M. V. Vinupaul, R. Bhattacharjee, R. Rajesh, and G. S. Kumar, "User characterization through network flow analysis," in *Proc. Int. Conf. Data Sci. Eng. (ICDSE)*, Aug. 2016, pp. 1–6, doi: 10.1109/ICDSE.2016.7823965.

[17] A. Hess, I. Marsh, and D. Gillblad, "Exploring communication and mobility behavior of 3G network users and its temporal consistency," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5916–5921, doi: 10.1109/ICC.2015.7249265.

[18] D. Rajashekar, A. N. Zincir-Heywood, and M. I. Heywood, "Smart phone user behaviour characterization based on autoencoders and self organizing maps," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 319–326, doi: 10.1109/ICDMW.2016.0052.

[19] A. Bakhshandeh and Z. Eskandari, "An efficient user identification approach based on netflow analysis," in *Proc. 15th Int. ISC, Iranian Soc. Cryptol., Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2018, pp. 1–5, doi: 10.1109/ISCISC.2018.8546856.

[20] S. Zhao, Y. Xu, X. Ma, Z. Jiang, Z. Luo, S. Li, L. T. Yang, A. Dey, and G. Pan, "Gender profiling from a single snapshot of apps installed on a smartphone: An empirical study," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1330–1342, Feb. 2020, doi: 10.1109/TII.2019.2938248.

[21] I. Paul, A. Khattar, P. Kumaraguru, M. Gupta, and S. Chopra, "Elites tweet? Characterizing the Twitter verified user network," Mar. 2019, *arXiv:1812.09710*. Accessed: Jul. 16, 2020. [Online]. Available: http://arxiv.org/abs/1812.09710

[22] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2037–2064, 4th Quart., 2014, doi: 10.1109/COMST.2014.2321898.

[23] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Comput. Netw.*, vol. 76, pp. 75–89, Jan. 2015, doi: 10.1016/j.comnet.2014.11.001.

[24] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Karlsruhe, Germany, Aug. 2003, pp. 3–10, doi: 10.1145/863955.863957.

[25] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett, G. Estrada, K. Ma'ruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Sep. 2017, doi: 10.1145/3138808.3138810.

[26] J. S. Rojas. (2020). Jsrojas/FlowLabeler. GitHub. [Online]. Available: https://github.com/jsrojas/FlowLabeler.

[27] A. Pekar. (2020). Drnpkr/FlowRecorder. GitHub. [Online]. Available: https://github.com/drnpkr/flowRecorder

[28] Z. Aouini. (2020). Nfstream/Nfstream. GitHub. [Online]. Available: https://github.com/nfstream/nfstream

[29] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, "NbClust: AnRPackage for determining the relevant number of clusters in a data set," *J. Stat. Softw.*, vol. 61, no. 6, 2014, Art. no. 1, doi: 10.18637/jss.v061.i06.

[30] K. Djouzi and K. Beghdad-Bey, "A review of clustering algorithms for big data," in *Proc. Int. Conf. Netw. Adv. Syst. (ICNAS)*, Jun. 2019, pp. 1–6, doi: 10.1109/ICNAS.2019.8807822.

[31] M. Hassani, "Overview of efficient clustering methods for high-dimensional big data streams," in *Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications*, O. Nasraoui and C.-E. B. N'Cir, Eds. Cham, Switzerland: Springer, 2019, pp. 25–42.

[32] G. Upton and I. Cook, *Understanding Statistics, Illustrated*. Oxford, U.K.: OUP Oxford, 1996.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[34] P. Joshi, "Incremental learning: Areas and methods—A survey," *Int. J. Data Mining Knowl. Manage. Process*, vol. 2, no. 5, pp. 43–51, Sep. 2012, doi: 10.5121/ijdkp.2012.2504.

[35] R. R. Ade and P. R. Deshmukh, "Methods for incremental learning: A survey," *Int. J. Data Mining Knowl. Manage. Process*, vol. 3, no. 4, pp. 119–125, Jul. 2013, doi: 10.5121/ijdkp.2013.3408.

[36] V. Lemaire, C. Salperwyck, and A. Bondu, "A survey on supervised classification on data streams," in *Proc. 4th Eur. Bus. Intell. Summer School (eBISS)*, in Tutorial Lectures, Berlin, Germany, E. Zimányi and R.-D. Kutsche, Eds. Cham, Switzerland: Springer, Jul. 2014, pp. 88–125.

[37] J. Zhong, Z. Liu, Y. Zeng, L. Cui, and Z. Ji, "A survey on incremental learning," presented at the 5th Int. Conf. Comput., Automat. Power Electron., Nov. 2017. [Online]. Available: https://webofproceedings.org/proceedings_series/article/artId/777.html

[38] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 399–404, doi: 10.1109/CCDC.2019.8832774.

[39] S. Madhavan and N. Kumar, "Incremental methods in face recognition: A survey," *Artif. Intell. Rev.*, Aug. 2019, doi: 10.1007/s10462-019-09734-3.

[40] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *J. Mach. Learn. Res.*, vol. 19, no. 72, pp. 1–5, 2018.

[41] A. Chefrour, "Incremental supervised learning: Algorithms and applications in pattern recognition," *Evol. Intell.*, vol. 12, no. 2, pp. 97–112, Jun. 2019, doi: 10.1007/s12065-019-00203-y.

[42] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2001, pp. 97–106, doi: 10.1145/502512.502529.

[43] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *J. Mach. Learn. Res.*, vol. 11, no. 52, pp. 1601–1604, 2010.

[44] N. C. Oza, "Online bagging and boosting," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 3, Oct. 2005, pp. 2340–2345, doi: 10.1109/ICSMC.2005.1571498.

[45] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases. ECML PKDD* (Lecture Notes in Computer Science), vol. 6321, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-15880-3_15.

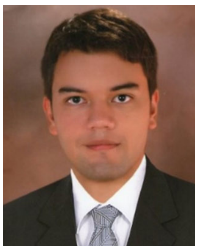
[46] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1469–1495, Oct. 2017, doi: 10.1007/s10994-017-5642-8.

[47] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001, doi: 10.1109/5326.983933.

[48] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987, doi: 10.1109/MASSP.1987.1165576.

[49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[50] A. Cervantes, C. Gagné, P. Isasi, and M. Parizeau, "Evaluating and characterizing incremental learning from non-stationary data," Jun. 2018, *arXiv:1806.06610*. [Online]. Available: http://arxiv.org/abs/1806.06610
[51] D. M. W. Powers. (Dec. 2007). *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*. [Online]. Available: https://www.researchgate.net/publication/228529307_Evaluation_From_Precision_Recall_and_F-Factor_to_ROC_Informedness_Markedness_Correlation
[52] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha, "Beyond kappa: A review of interrater agreement measures," *Can. J. Statist.*, vol. 27, no. 1, pp. 3–23, Mar. 1999, doi: 10.2307/3315487.
[53] (Oct. 23, 2020). *Scikit-Multiflow's Documentation*. Accessed: Sep. 23, 2020. [Online]. Available: https://scikit-multiflow.github.io/scikit-multiflow/
[54] P. K. Srimani and M. M. Patil, "Performance analysis of Hoeffding trees in data streams by using massive online analysis framework," *Int. J. Data Mining Model. Manage.*, vol. 7, no. 4, pp. 293–313, 2015.
[55] D. Stathakis, "How many hidden layers and nodes?" *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009, doi: 10.1080/01431160802549278.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[57] (Oct. 29, 2020). *Labeled Network Traffic Flows—141 Applications*. Accessed: Sep. 29, 2020. [Online]. Available: https://kaggle.com/jsrojas/labeled-network-traffic-flows-114-applications
[58] (Oct. 29, 2020). *User OTT Consumption Profile—2019*. Accessed: Sep. 29, 2020. [Online]. Available: https://kaggle.com/jsrojas/user-ott-consumption-profile-2019



JUAN SEBASTIÁN ROJAS received the bachelor's degree in electronics and telecommunications engineering and the M.Sc. degree in telematics engineering from the Universidad del Cauca, Colombia, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in telematics engineering. He is also an Active Researcher with the Telematics Engineering Group, Universidad del Cauca. Since 2016, he has been granted a Ph.D. Scholarship by MinCiencias and the Leader entity of the science, technology and innovation system in Colombia. Furthermore, he has been an Assistant Professor in several courses related to converged services, data mining, and machine learning. His research topic involves the study of users' Over The Top consumption behavior aimed at the personalization of service degradation through incremental learning supported by the Knowledge Defined Networking paradigm.

ADRIAN PEKAR received the Ph.D. degree in computer science from the Technical University of Košice, Slovakia, in 2014. He is currently an Assistant Professor with the Department of Networked Systems and Services, Budapest University of Technology and Economics, Hungary. Prior to this, he held research, teaching, and engineering positions in New Zealand and Slovakia. His research interests include network traffic classification and management, traffic measurement data reduction and visualization, and stream processing.

ÁLVARO RENDÓN (Senior Member, IEEE) received the B.S. degree in electronics engineering and the M.S. degree in telematics engineering from the Universidad del Cauca, Colombia, in 1979 and 1989, respectively, and the Ph.D. degree in telecommunications engineering from the Technical University of Madrid, in 1997. He is currently a Full Professor with the Department of Telematics, University of Cauca, where he is also the Director of the Doctoral Program in Telematics Engineering. His research interests include telecommunications services, e-health, and S&T management. He was a recipient of the Outstanding Paper Award in the category of Tools Track at the First IEEE International Conference on Engineering of Complex Computer Systems in 1995.

JUAN CARLOS CORRALES received the bachelor's and master's degrees in telematics engineering from the University of Cauca, Colombia, in 1999 and 2004, respectively, and the Ph.D. degree in sciences, specialty computer science from the University of Versailles Saint-Quentin-en-Yve-lines, France, in 2008. He is currently a full-time Professor and leads the Telematics Engineering Group, University of Cauca. His research interests include service composition and data analysis.