A Proactive and Big data-enabled Caching Analysis Perspective

Engin Zeydan
Turk Telekom Labs, Istanbul, Turkey
engin.zeydan@turktelekom.com.tr

Ejder Baştuğ
Nokia Bell Labs, 91620 Nozay, France
ejder.bastug@nokia-bell-labs.com

Mehdi Bennis
Centre for Wireless Communications, University of Oulu, Finland
bennis@ee.oulu.fi

Mérouane Debbah
CentraleSupélec, Gif-sur-Yvette, France
merouane.debbah@centralesupelec.fr

Abstract: Large-scale data analysis is becoming an important source of information for Mobile Network Operators (MNOs). MNOs can now investigate the feasibility of possible new technological advances such as storage/memory utilization, context-awareness and edge/cloud computing using analytic platforms designed for big data processing. Within this context, studying caching from mobile data traffic analytical perspective can offer rich insights on evaluating the potential benefits and gains of proactive caching at base stations. In this chapter, we study how data collected from MNOs can be leveraged using machine learning tools in order to infer insights into the benefits of caching. Through our practical architecture, vast amount of data can be harnessed for content popularity estimations and placing strategic contents at BSs. Our results demonstrate several gains in terms of both content demand satisfaction and backhaul offloading rates while utilizing real-world datasets collected from a major MNO.

I. Introduction

The continuous increase in mobile data traffic demand due to video, social networks and over-the-top (OTT) applications are pushing mobile network operators to search for new ways to handle their growing complex networks. This increase of traffic from diverse domains (e.g., healthcare, machine-to-machine communication, connected cars, user-generated content, smart metering) have different characteristics (e.g., structured/non-structured) and is usually called as *Big Data* [1]. While big data has "big blessings" there are compelling challenges in handling large-scale data sets due to the tremendous volume and dimensionality of the data. A primary challenge of big data analytics is to move through large volumes of data in order to find hidden patterns for decision making. In fact, the time of gathering and storing data in remote standalone servers for offline decision making is outdated. Instead, mobile network operators are looking decentralized and scalable network architectures in which anticipatory resource management play an important role exploiting recent advances in storage/memory, context-awareness and local/edge/cloud computing [2]–[4]. In the world of wireless, big data provides to network planning a plenty of new information sets that can be inter-connected to perform a better understanding of users and networks (e.g., location, user velocity, social geodata). In addition, public data from social networks like Twitter and Facebook gives additional information about the life of the network, which could be further leveraged. The associated advantages are a higher accuracy of user location information or the ability to efficiently identify and estimate user clustering, for example for particular events. Definitely, the enormous potential associated with big data has stimulated a great research interest from industry, government and academics (see [5] for a further reading), and will keep going in the upcoming years.

Concurrently, mobile cellular networks are shifting towards the next generation of 5G (and beyond) wireless communications, in which ultra-dense networks, millimetre wave communications, massive multiple-input multiple-output (massive-MIMO), edge caching, device-to-device communications play a crucial role (see [6] and references therein). Instead of classical base station-centric network paradigm assuming *dumb* terminals and *reactive* network optimization, 5G (and beyond) networks will be surely disruptive in the sense of being user-centric, context-aware and proactive/anticipatory in essence. While continued improvement in spectral efficiency is anticipated, the sophistication of air interfaces of existing systems (LTE-Advanced) mean that no major improvements of spectral efficiency can be expected. Extra measures like the brute force deployment of mobile network infrastructure (number of cells) and the licensing of more spectrum are excessively expensive. Therefore, creative solutions are called upon.

In this book chapter, given the motivations and challenges above, we are aiming to focus on a proactive caching architecture for optimization of 5G (and beyond) wireless networks where we leverage large amount of available data with the use of big data analytics and machine learning tools. Particularly, we examine the gains of proactive caching both in terms of backhaul offloadings and content demand satisfactions, where machine learning tools are utilized to model and estimate the spatio-temporal user behaviour for proactive cache placement. By caching critical contents at the edge of network, specifically at the base stations, mobile

network resources are used more efficiently and users' experience is further enhanced. Notwithstanding, the prediction of content popularity coupled with spatio-temporal behaviour of users is a non-trivial problem given the high dimensional properties of data, data sparsity and lack of measurements. In this respect, we introduce a platform to parallelize the computation and execution of the content estimation algorithms for cache placement at the base stations. As a real-world practical study, a huge amount of data gathered from a Turkish telecom operator, one of the largest mobile operator in Turkey with 16.2 million of active subscribers, is investigated for several caching scenarios. Specifically, the traces of mobile users' activities are gathered from many base stations in hours of time interval and are analyzed inside the network under the privacy constrains and regulations. The analysis is done on a big data platform and cache placement at the base stations has been numerically studied for further improvements of users' experience and backhaul offloadings.

The rest of this book chapter is structured as follows. Requirements, challenges and benefits of big data analytics in 5G networks is described in Section II. Network model for proactive edge caching is given in Section III. A practical case study of content popularity prediction on a big data platform is showcased in Section IV, including a characterization of users' content access behaviour. Afterwards, numerical studies for cache-enabled base stations and technical discussions are given in Section V. We finally conclude in Section VI and give our future directions in the same section.

II. Big Data Analytics for Telcos: Requirements, Challenges and Benefits

Today networking requirements are becoming software-defined with the aim of being more scalable and flexible for big data. Tomorrow's big networks is expected to be even more complex and interconnected. Because of this, mobile operators (MOs)' data centers and network infrastructures shall need to track traffic patterns of tens of millions of users/devices (e.g. location, traffic demand pattern, capability, etc) for a more detailed analysis and better network optimization.

A. Current Challenges and Trends in Big Data Networking

Data traffic patterns inside mobile operators' data centers have indeed changed dramatically. Big data has allowed large traffic exchange between gateway devices at backhaul. Even though wireless technologies have greatly improved from 2G to 4G, backhaul connections of mobile cellular networks have not seen such a rapid jump. Therefore, the mobile backhaul intra-traffic is slowly getting larger than the inter-traffic between mobile backhaul devices and end-users. In fact, in existing carrier networks, in addition to managing mobile users' traffic via mobile backhaul, fetching data from a number of different backend, database and cache servers, as well as the data produced by gateway and backhaul elements also contribute to such a traffic load within the operator's infrastructure. Indeed, interactions of user terminal (UT) requires various interactions with hundreds of servers, routers and switches within the backhaul and core network. For instance, for an original user's HTTP request of 1 KByte, the intranet data traffic might go up to 930x [7]. This is in contrast to the traditional carrier network architecture where it assumes the client and wireless access nodes

as bottlenecks lacking computational overhead rather than the backhaul infrastructure. In addition, as data increase is a grand challenge in today's mobile infrastructures, moving such a big data-driven networks in *cloud environments* is challenging. In this regard, mobile edge computing (sometimes called "Fog" computing) is yet another rising technology where edge devices have *cloud-computing like features* within the radio access network to perform functionalities such as communication, storage and control [3]. Notwithstanding for 5G (and beyond) networks, it could be remarked that installing distributed cloud computing capabilities near to each base stations (BSs) site (particularly at locations where traffic volume is relatively little) may also increase the deployment cost that is comparable to centralised computing solutions due to availability of many sites in a typical MO. Furthermore, for modelling and prediction of users' spatio-temporal content access patterns in 5G (and beyond) mobile networks, traffic handled to a centralized location requires horizontal scale-out across servers and racks that is only possible inside the core-site of a MO for proper analysis rather than distributed places with relatively less traffic.

B. When Big Data Analytics Meets Caching

Thanks to the recent advances in networking, standards as well as personal communications, big data has received great popularity notably inside data centers and mobile network operators. With the increasing challenges of big data in the networking area, it becomes quite clear that the only option to deal with the increasing data traffic is via better data handling and movement of data from cloud in to the edge. Lately, Hadoop has been successful as a big data management application solution providing dramatic cost reduction over conventional tier-one database architectures, processing capabilities of many data formats and parallel computing over multiple nodes. In addition, sophisticated analytic techniques in machine learning together with non-relational databases which can exploit big data (i.e., NoSql databases) have created a better opportunity of understanding big data.

It is evident that moving content's closer to the edge is critical whenever user's connectivity fails while having streaming and/or fetching activities. To alleviate this, moving data closer to users by reducing the distance of content to users and placing the right content and application at the edge gives better user experience. For example, using Hadoop's distributed data processing engine for analyzing users' content access patterns (through the core site of MOs) as well as caching proactively at the edge (i.e., at BSs) can ease the backhaul usage and improve users' quality-of-experience (QoE) in terms of latency reduction. In the following, we detail the related system model, then describe our Hadoop-based big data processing platform and its relation with edge caching, as one way of handling and exploiting big data inside MOs.

III. System Model

Assume a network deployment of M small base stations (SBSs) from the set $\mathcal{M} = \{1, \ldots, M\}$ and N UTs from the set $\mathcal{N} = \{1, \ldots, N\}$. Each SBS m can access to the broadband Internet connection via a wired backhaul link with capacity C_m Mbyte/s, and can provide this broadband service to its users via a wireless link with total capacity of C'_m Mbyte/s. Since the backhaul capacity is usually limited in densely deployed

SBSs scenarios [6], we further assume that $C_m < C'_m$. Moreover, consider that each user $n \in \mathcal{N}$ is associated to single SBS and is served via unicast sessions[1]. Specifically, we suppose that UTs demands contents (i.e., videos, files, news, etc.) from a library $\mathcal{F} = \{1, \ldots, F\}$, where each content f in this library has a lenght of $L(f)$ Mbyte and bitrate requirement of $B(f)$ Mbyte/s, with

$$L_{\min} = \min_{f \in \mathcal{F}} \{L(f)\} > 0 \tag{1}$$

$$L_{\max} = \max_{f \in \mathcal{F}} \{L(f)\} < \infty \tag{2}$$

and

$$B_{\min} = \min_{f \in \mathcal{F}} \{B(f)\} > 0 \tag{3}$$

$$B_{\max} = \max_{f \in \mathcal{F}} \{B(f)\} < \infty. \tag{4}$$

The users' content demands indeed follow a Zipf-like distribution $P_{\mathcal{F}}(f), \forall f \in \mathcal{F}$ such as [10]:

$$P_{\mathcal{F}}(f) = \frac{\Omega}{f^{\alpha}} \tag{5}$$

where

$$\Omega = \Big(\sum_{i=1}^{F} \frac{1}{i^{\alpha}} \Big)^{-1}.$$

The parameter α in (5) models the shape of the distribution. Such power laws (distributions) are used to characterize several real-world phenomena, for example the distribution of files in the web-proxies [10] and the traffic dynamics of mobile cellular devices [11]. Bigger values of α mean a steeper distribution, meaning that a small portion of contents are highly popular than the rest of the catalog. On the other hand, the smaller values means a more uniform behaviour with almost identical popularity of contents. The practical value of parameter α varies depending on users' content access patterns and SBSs deployment strategies (i.e., home, enterprise, urban and rural environments), and its value in our practical setup will be provided in the next sections.

Provided that our global content popularity is decreasingly ordered, the content popularity matrix of the m-th SBS at time t is specifically modelled by $\mathbf{P}^m(t) \in \mathbb{R}^{N \times F}$ where each entry $P^m_{n,f}(t)$ quantizes to the probability that the n-th user demands the f-th content. In other words, the matrix $\mathbf{P}^m(t)$ is the local content popularity distribution experienced at the base station m at time t, whereas the Zipf distribution $P_{\mathcal{F}}(f), \forall f \in \mathcal{F}$ is used to model the global content popularity distribution of all contents in decreasing order.

In our scenario, we suppose that each SBS has a limit storage size of S_m and proactively caches stratecially selected contents from the library $\mathcal{F}$ during peak-off hours. By doing this, the bottlenecks due the limited-backhaul are mitigated during the delivery of users' content demands in peak hours. The amount of satisfied requests and backhaul load are of high importance and are given as follows. Assume that D number of contents are demanded during the duration of T seconds, and are denoted by the set $\mathcal{D} = \{1, ..., D\}$. Consider that the delivery of content is commenced immediately when the content demand $d \in \mathcal{D}$ arrives to

[1]One can also extend the unicast service model to the multicast case. See [8], [9] for works in this direction.

the SBS. Then, the demand d is assumed *satisfied* if the rate of content delivery is equal or higher than the bitrate of the content in the end of delivery, that is:

$$\frac{L(f_d)}{\tau'(f_d) - \tau(f_d)} \geq B(f_d) \tag{6}$$

where f_d models the demanded content, $L(f_d)$ and $B(f_d)$ are the length and bitrate of the content, $\tau(f_d)$ is the arrival time of the content demand and $\tau'(f_d)$ is the end time of delivery.[2] Defining the condition in (6) is due to the fact that, if the delivery rate is not equal nor higher than the bitrate of the requested content, the disruption during the playback (or download) happens therefore users would have lower QoE[3]. For this reason, the scenarios where this condition holds are more preferable for higher QoE. In (6), observe also that the end time of delivery for demand d, given by $\tau'(d)$, highly depends on the load of the network, capacities of the backhaul and wireless links as well as availability of contents at the SBSs. Under these motivations and and given the definition of satisfied demands above, the users' average demand *satisfaction ratio* is therefore defined for the set of all demands, that is:

$$\eta(\mathcal{D}) = \frac{1}{D} \sum_{d \in \mathcal{D}} \mathbb{1} \left\{ \frac{L(f_d)}{\tau'(f_d) - \tau(f_d)} \geq B(f_d) \right\} \tag{7}$$

where $\mathbb{1}\{...\}$ denots the indicator function that takes 1 if the condition holds and 0 otherwise. Moreover, denoting $R_d(t)$ Mbyte/s as the instantaneous rate of backhaul for the demand d at time t, with $R_d(t) \leq C_m$, $\forall m \in \mathcal{M}$, the average *backhaul load* is then expressed as:

$$\rho(\mathcal{D}) = \frac{1}{D} \sum_{d \in \mathcal{D}} \frac{1}{L(f_d)} \sum_{t=\tau(f_d)}^{\tau'(f_d)} R_d(t). \tag{8}$$

Above, the outer sum is over the set of all demands whereas the inner sum gives the total amount of data passed over the backhul for demand d which is at most equal to the size of demanded content $L(f_d)$. The instantaneous rate of backhul for demand d, modelled by $R_d(t)$, excessively depends on the load of the system, capacity of the backhaul link and strategically cached contents at the base stations.

Indeed, by caching the contents at the SBSs, the access delays to the contents are minimized particularly during the peak hours, hence yielding better satisfaction ratio and smaller backhaul load. To detail this, now assume the cache decision matrix of SBSs as $\mathbf{X}(t) \in \{0,1\}^{M \times F}$, where the entry $x_{m,f}(t)$ is 1 if the f-th content is cached at the m-th SBS at time t, and 0 otherwise. Then, the backhaul offloading problem under

[2]One can also leverage future information (i.e., start time of demands, end time of content delivery) in the context of proactive resource allocation (see [12] for example).

[3]In general, a video content has typically a bitrate requirement between 1.5 to 68 Mbit/s [13].

a specific demand satisfaction constraint is formally modelled as follows:

$$\underset{\mathbf{X}(t),\mathbf{P}^m(t)}{\text{minimize}} \quad \rho(\mathcal{D}) \tag{9}$$

$$\text{subject to} \quad L_{\min} \leq L(f_d) \leq L_{\max}, \qquad \forall d \in \mathcal{D}, \tag{9a}$$

$$B_{\min} \leq B(f_d) \leq B_{\max}, \qquad \forall d \in \mathcal{D}, \tag{9b}$$

$$R_d(t) \leq C_m, \qquad \forall t, \forall d \in \mathcal{D}, \forall m \in \mathcal{M}, \tag{9c}$$

$$R'_d(t) \leq C'_m, \qquad \forall t, \forall d \in \mathcal{D}, \forall m \in \mathcal{M}, \tag{9d}$$

$$\sum_{f \in \mathcal{F}} L(f) x_{m,f}(t) \leq S_m, \quad \forall t, \forall m \in \mathcal{M}, \tag{9e}$$

$$\sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} P^m_{n,f}(t) = 1, \quad \forall t, \forall m \in \mathcal{M}, \tag{9f}$$

$$x_{m,f}(t) \in \{0,1\}, \ \ \forall t, \forall f \in \mathcal{F}, \forall m \in \mathcal{M}, \tag{9g}$$

$$\eta_{\min} \leq \eta(\mathcal{D}), \tag{9h}$$

where $R'_d(t)$ Mbyte/s is the instantaneous rate of wireless link for demand d and $\eta_{\min}$ denotes the minimum target satisfaction ratio. Moreover, the constraints (9a) and (9b) are to limit the length and bitrate of contents in the catalog for feasible solution, the constraints (9c) and (9d) describes the backhaul and wireless link capacity constraints, (9e) is the storage capacity for caching, (9f) is to justify the content popularity matrix as a probability measure, (9g) is the binary decision variables of caching, and finally the expression in (9h) models the satisfaction ratio constraint for QoE.

In order to deal with this problem, the cache decision matrix $\mathbf{X}(t)$ and the content popularity matrix estimation $\mathbf{P}^m(t)$ need to be optimized jointly. Unfortunately, solving the problem (9) is not straightforward as:

i) the storage capacity of SBSs, the backhaul and wireless link capacities are limited,
ii) the contents in the catalog and users with unknown ratings[4] are very large in real world,
iii) the optimal uncoded[5] cache placement for a given demand is non-tractable [15]–[17],
iv) the SBSs have to track, learn and predict the sparse content popularity/rating matrix SBSs $\mathbf{P}^m(t)$ while making the cache placement.

In order to deal with these issues, we restrict ourselves to the case that cache placement is done during peak-off hours, therefore $\mathbf{X}(t)$ is static during the content delivery in peak hours and is modelled by $\mathbf{X}$. In addition, the content popularity matrix is stationary during T time slots and identical among the base stations, therefore $\mathbf{P}^m(t)$ is denoted by $\mathbf{P}$.

[4]The term "rating" refers to the empirical value of content popularity/probability and is interchangeable throughout the paper.

[5]In the information-theoretical sense, the caching placement can be divided into "coding" and "uncoded" groups (see [14] for example).

After these assumptions, we now also assume that the problem can be decomposed into two separate parts in which the content popularity matrix $\mathbf{P}$ is first predicted, then is exploited for the caching placement $\mathbf{X}$ accordingly. Indeed, if sufficient amount of users' ratings can be collected at the SBSs, we can build a k-rank approximate popularity matrix $\mathbf{P} \approx \mathbf{N}^T\mathbf{F}$, by jointly learning the factor matrices $\mathbf{N} \in \mathbb{R}^{k\times N}$ and $\mathbf{F} \in \mathbb{R}^{k\times F}$ which minimizes the following cost function:

$$\underset{\mathbf{P}}{\text{minimize}} \sum_{P_{ij}\in\mathcal{P}} \Big(\mathbf{n}_i^T\mathbf{f}_j - P_{ij}\Big)^2 + \mu\Big(||\mathbf{N}||_F^2 + ||\mathbf{F}||_F^2\Big) \tag{10}$$

where the summation is applied over the corresponding user/content rating pairs P_{ij} in the training set $\mathcal{P}$. The vectors $\mathbf{n}_i$ and $\mathbf{f}_j$ here represent the i-th and j-th columns of $\mathbf{N}$ and $\mathbf{F}$ matrices respectively, and $||.||_F^2$ means the Frobenius norm. The parameter μ is to make a balance between the regularization and fitting the training data. Therein, high correspondence between the user factor matrix $\mathbf{N}$ and content factor matrix $\mathbf{F}$ results in a better estimate of $\mathbf{P}$. Indeed, the problem (10) is a regularized least square problem in which the matrix factorization is embedded in the formulation. Among many approaches, the matrix factorization methods are usually applied to solve such problems and has many applications in recommendation systems (i.e., Netflix video recommendation). In our setup detailed in the next sections, we have applied regularized sparse singular value decomposition (SVD) to solve the problem algorithmically, exploiting the least square nature of the problem. The survey of these methods, sometimes named collaborative filtering (CF) tools, can be found in [18], [19]. When the estimation of content popularity matrix $\mathbf{P}$ is attained, the caching placement $\mathbf{X}$ can be done accordingly.

In practical scenarios, the estimation of $\mathbf{P}$ in (10) can be performed by collecting/analysing huge amount of available data on a *big-data platform* of the mobile network operator, and strategic/popular contents from this prediction can be cached at the *cache-enabled base stations* whose cache decisions are modelled by $\mathbf{X}$. With this way of operation, the backhaul offloading problem in (9) is minimized and users' content demands are better satisfied. Our system model including such an practical setup is illustrated in Fig. 1. In the next, as a real world practical study, we describe our big data platform and analyze users' traffic characteristics by gathering large amount of data on this platform. The gathered data will be used to predict the content popularity matrix $\mathbf{P}$ which is then used for the cache placement $\mathbf{X}$ and will be described in the next sections.

IV. Big Data Platform

This section details our big data processing platform for analyzing users' data traffic. The goal of of this platform is to gather users' data traffic and extract meaningful information for proactive cache placement. Assuming that Hadoop is running inside the core location of a MO, requirements of this platform for our analysis can be debated as follows:

i) **Massive Data Volume Processing in Shorter Time:** In order to cache the contents proactively, data processing platform inside the core infrastructure should be able to gather and combine data from various data sources, and provide intelligent insights quickly and reliably. For this matter, after

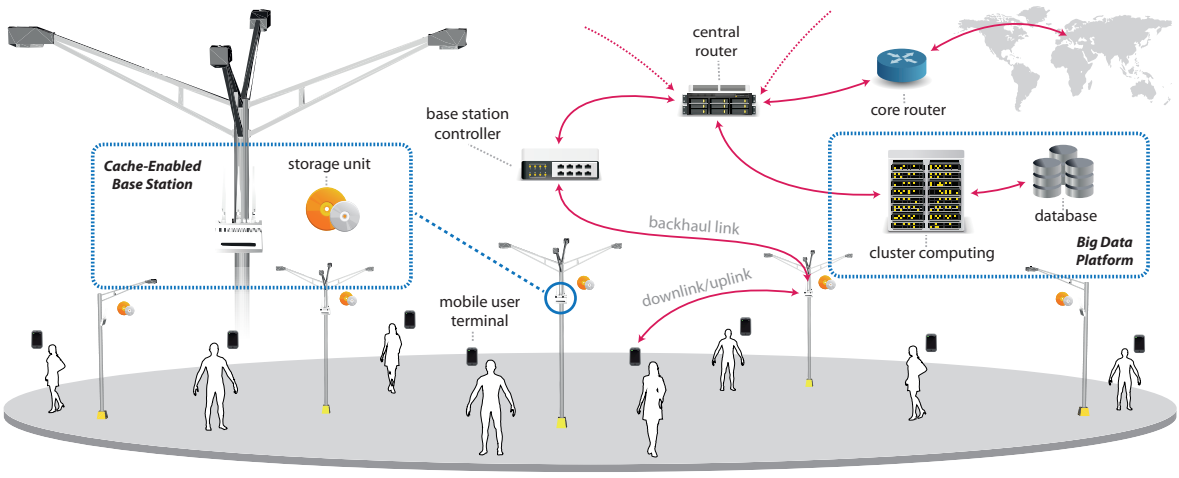

Figure 1. A sketch of the network model. A *big data platform* is tracking/predicting users' demand, whereas *cache-enabled base stations* cache the strategic predicted contents on the big data platform.

mirroring the data streaming interface via network analyzing tools, the gathered/collected data should be exported into a big data storage platform (i.e., Hadoop Distributed File System (HDFS)) via enterprise data integration techniques (i.e., *Spring Integration*) for a comprehensive analysis.

ii) **Cleansing, Parsing and Formatting Data:** Data cleansing is an important procedure in the data analysis process. Indeed, before applying any machine learning and statistical analysis on the data, the data itself has to be cleaned and this generally might take much more time than the machine learning analysis. In fact, there are several steps required in data cleansing process. First, raw data should be cleaned. The raw data by itself might have malfunctioning, inappropriate and inconsistent packets with incorrect character encoding, etc. therefore requires elimination. The next (second) step is then to extract the relevant fields/properties from the raw data. In this step, the required headers from data and control packets that will be used for investigation are extracted based on the data analysis and modelling requirements. Finally, the parsed/extracted data need to be encoded accordingly (i.e., in *Avro* or *Parquet* format) for appropriate storage inside HDFS.

iii) **Data Analysis:** Given the formatted data in HDFS, various data analytics methods can be exploited over header or/and payload information of both control and data planes via high level query languages (i.e., Hive Query language (HiveQL) and Pig Latin). The goal of such a procedure is to detect relationships between data and control packets, e.g. the position or Mobile Subscriber Integrated Services for Digital Network Number (MSISDN) of mobile users (which are available in control packets but not in data packets) to the demanded content (which are available only in data packets) via successive Map-Reduce operations. Note that the concept of Map-Reduce is naturally embedded under HiveQL and Pig Latin.

iv) **Statistical analysis and visualizations:** Once machine learning analysis is applied to predict the

spatio-temporal user content access patterns for proactive cache placement, the results of such analysis can be recorded and recycled. In addition, the results can be re-formatted to be exploited for further analysis using appropriate Extract, Transform and Load (ETL) tools, and can be fed to different kind of processing systems such as Apache Spark's MLlib, etc. Moreover, visualizations such as tables and graphs can be used to depict the data in a visual way for ease of understanding.

With such a platform in hand, machine learning methods that lies at the core of recommendation systems can be applied such that users' strategic contents can be deduced from a massive amount of available data. Subsequently, we perform a simulation study for characterizing the potential gains of caching at BSs.

A. Platform description

The big data platform showed in this work is installed in the operator's core network. The aim of this platform, as explained before, is to store users' traffic and extract useful information that is essentially going to be exploited for content popularity estimation. In overall, the operator's network is made of multiple districts with more than 10 regional core areas scattered around Turkey. The total average throughput over all regional areas is made of approximately over 15 billion packets in uplink direction and over 20 billion packets in the downlink direction per day. This is equivalent to approximately over 80 TByte of total data flowing in uplink and downlink daily in a mobile operator's core network. The data consumption behaviour leads to an exponential increase in data traffic of a mobile network operator. For instance, in 2012, approximately over 7 TByte in both uplink and downlink daily traffic has been observed.

The data traces that will be detailed in the next, are gathered from one of the operator's main network region (which has mobile traffic from several base stations), and are stored on a server with a high speed link of 200 Mbit/sec at peak hours. As part of capturing the Internet traffic by this server on the platform, a procedure is established by mirroring real-world Gn interface data.[6] Once mirroring step of Gn interface is established, network traffic is then forwared to the server on the platform. For our technical analysis, we have gathered mobile traffic of roughly 7 hours starting from 12 pm to 7 pm on Saturday 21'st of March 2015. This mobile traffic is then analyzed on the big data platform that is actually based on Hadoop.

Among the existing platforms, Hadoop appears to be one of the most notable solution as an open source product [20]. It consists of a storage module (called HDFS) and a computation module (called MapReduce). While HDFS has both centralized and distributed implementations, MapReduce naturally consider a distributed structure where the the jobs can be executed in parallel on multiple nodes.

As alluded in the previous section, the precision and accuracy of the was examined in the operator's network. In particular, the big data platform was implemented via using Cloudera's Distribution Including Apache Hadoop (CDH4) [21] version on 4 nodes including 1 cluster name node, with computational powers corresponding to each node with INTEL Xeon CPU E5-2670 running @2.6 GHz, 32 Core CPU, 132 GByte

[6]Gn is an interface Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Network packets transmitted from a user terminal to the packet data network (PDN), i.e. internet, pass via SGSN and GGSN in which GPRS Tunneling Protocol (GTP) is the main protocol in network packets going through Gn interface.

RAM, 20 TByte hard disk. This big data platform is used to exract the meaningful information from raw data which is given as follows.

B. Data extraction process

In the beginning, the raw data is parsed using Wireshark command line utility *tshark* [22] for extraction of the related fields of CELL-ID (or service area code (SAC) in our case), with the aim of uniquely identifying a *service area* within a *location area*[7]), LAC, Hypertext Transfer Protocol (HTTP) request-uniform resource identifier (URI), tunnel endpoint identifier (TEID)[8] and TEID-DATA for data and control plane packets respectively, and FRAME TIME showing arrival time of packets. The HTTP Request-URI is a Uniform Resource Identifier that points the resource upon which to apply the request. The *control* packets has the information fields that contains the information required for future data packets. It has cell identification ID (CELL-ID), LAC and TEID-DATA fields. The *data* packets has HTTP-URI and TEID fields.

In the following step, once those relevant fields from both control and data packets are obtained, the extracted data is then moved into HDFS for further analysis. In HDFS, one can perform several data analytics over the gathered data using HiveQL [23]. For instance, in order to find the HTTP Request-URIs at specific location, the HTTP-URI can be combined with CELL-ID-LAC fields over the same TEID and TEID-DATA fields for data and control packets respectively. In our technical analysis, because of to the limitations on collected number of rows of HTTP-URI fields with a related CELL-ID-LAC fields after mapping, we have continued with HTTP Request-URIs and TEID mappings.

With HDFS, a temporary table called *traces-table-temp* is created using Hive QL. The *traces-table-temp* is made of HTTP Request-URI, FRAME TIME and TEID fields. Once the table is constructed, the length of each HTTP request-URI request is calculated using an additional URI-*size calculator* program that uses HTTPClient API [24] such that the final table named *traces-table* has fields of SIZE, HTTP Request-URIs, FRAME TIME and TEID. This final table contains roughly 420.000 of 4 millions HTTP request-URI's with SIZE field returned as not zero or null due to unavailability of HTTP response for some requests. We remark that, in a given session with a particular TEID, there might be several HTTP request-URIs. Eeach TEID belongs to particular user. Each user might have several TEIDs with muliple HTTP request-URIs. The procedures of data extraction on the big data platform is summarized in Fig. 2. We note that the data extraction procedure is customized to our scenario for proactive caching. Nevertheless, similier line of studies in terms of platform and exploitation of big data analytics for mobile network operators can be found in [25]–[30].

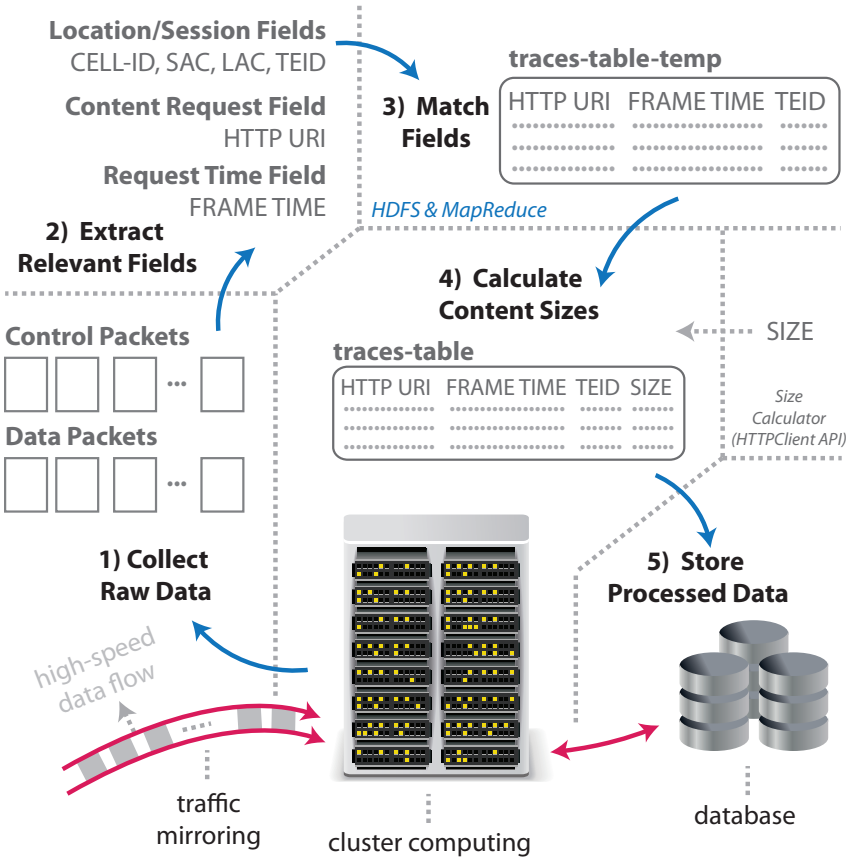

Figure 2. A summary of the data extraction procedures on the big data platform.

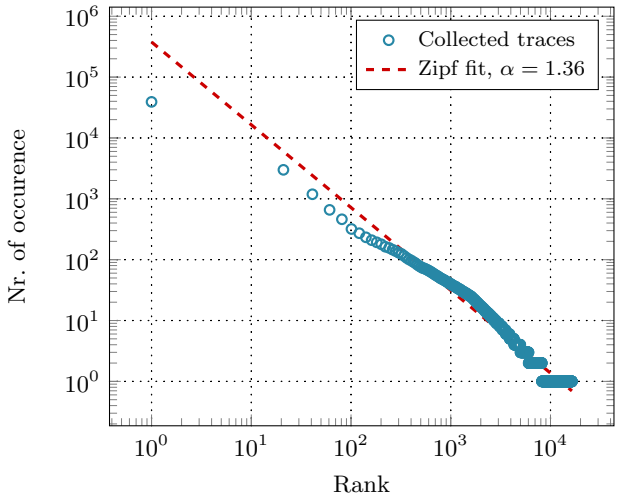

Figure 3. Global content popularity distribution.

C. Traffic Characteristics

Based on stored information in *traces-table*, the global content popularity distribution (namely HTTP-URI popularity distribution) in a decreasing ranked order is shown in Fig. 3. According to the gathered data, we remark that the popularity behaviour of contents can be characterized by a Zipf law with steepness parameter $\alpha = 1.36$.[9]. Therein, the Zipf line is calculated in the least square sense from the gathered traces, and the parameter α is then calculated from the slope of the curve. On the other hand, cumulative size of ranked contents is depicted in Fig. 4. The cumulative content size up to 41-th most-popular contents has 0.1 GByte of size, whereas a sharp increase occurs afterwards. This clearly points out that the most of demanded contents in our traces has low content sizes and the content with bigger sizes are relatively less demanded.

We would like to remark that a more comprehensive characterization of the traffic for cache placement is left for future work. In fact, characterization of the traffic in web proxies which are installed in the intermediate level of network [10], a particular video content catalog in a campus network [33], traffic of mobile users in Mexico [34] are found in the literature. Different than these works, we focus on the characterization of mobile traffic gathered from several base stations in a large regional area, and exploit this traffic for proactive cache placement (i.e., content popularity distribution, cumulative size distribution). Given the available information in *traces-table*, we in the next simulate a scenario of cache-enabled base stations.

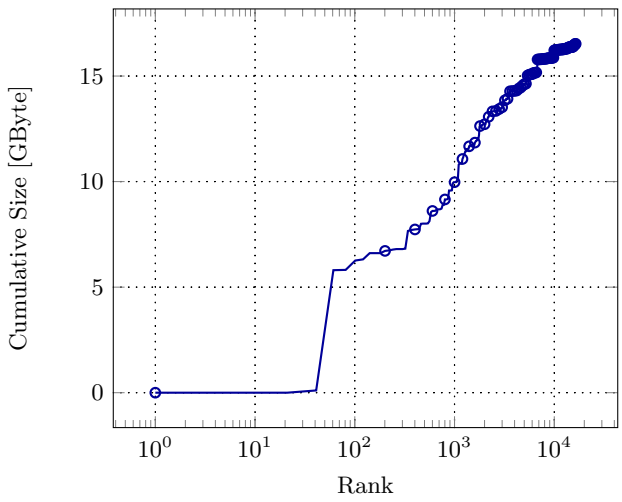

Figure 4. Cumulative size distribution.

[7]The service area identified by SAC is an area of single or multiple base stations, and corresponds to a location area that is uniquely quantified by location area code (LAC). In general, tens or even hundreds of base stations can run in a specific location area.

[8]A TEID uniquely characterizes a tunnel endpoint on the receiving end of the GTP tunnel. A local TEID value is given at the receiving end of a GTP tunnel for sending messages through the tunnel.

[9]The practical values of steepness parameter α might vary depending on the scenario. For example, the steepness parameter of content popularities in YouTube catalog is between 1.5 to 2.5 [31], [32].

V. Numerical Results and Discussions

The summary of parameters used in the numerical setup is given in Table I. For sake of simplicity in analysis, the storage, backhaul, and wireless link capacities of small cells are considered to be identical within each other.

Table I
Summary of simulation parameters.

Parameter	Description	Value
T	Time slots	6 hours 47 minutes
D	Nr. of requests	422529
F	Nr. of contents	16419
M	Nr. of small cells	16
$L_{\min}$	Min. size of a content	1 Byte
$L_{\max}$	Max. size of a content	6.024 GByte
$B(f)$	Bitrate of content f	4 Mbyte/s
$\sum_m C_m$	Total backhaul link capacity	3.8 Mbyte/s
$\sum_m \sum_n C'_m$	Total wireless link capacity	120 Mbyte/s

In the simulation study, the total D number of demands are taken from the processed collected data (namely *traces-table*), spanning over a duration of 6 hours 47 minutes. The arrival times of each demand (FRAME TIME), demanded content (HTTP-URI) and content size (SIZE) information are taken from the same table. Then, these demands are associated to M base stations pseudo-randomly. For the backhaul offloading problem in (9), the content popularity matrix $\mathbf{P}$ and cache placement strategy $\mathbf{X}$ are calculated separately. More precisely, the following two approaches are used for constructing the content popularity matrix $\mathbf{P}$:

- *Ground Truth*: The content popularity matrix $\mathbf{P}$ is build from all existing data in *traces-table* without solving the problem in (10). Observe that the rows of $\mathbf{P}$ quantify base stations and columns represent contents. The rating density of the matrix $\mathbf{P}$ is 6.42%.
- *Collaborative Filtering*: In order to estimate the content popularity matrix $\mathbf{P}$, the problem in (10) is solved by first choosing 10% of ratings in *traces-table* uniformly at random. Afterwards, these selected ratings are used for the training of the algorithm then remaining entries/ratings of $\mathbf{P}$ are predicted. In particular, the regularized SVD from the CF approaches [19], [35] is considered in the algorithmic part.

After building the content popularity matrix $\mathbf{P}$ given the methods above, the cache placement ($\mathbf{X}$) is set by storing the most-popular contents greedily at the SBSs until all storage is fulfilled (see [15] for more details). Once these contents are cached proactively at the SBSs at $t = 0$, the content demands are then served until the delivery of all contents are finalized. The performance metrics, namely demand satisfaction and backhaul load, are computed accordingly.

The variation of users' demand satisfaction with respect to the storage size is depicted in Fig. 5. Therein, the storage size is normalized where 100% of storage size means the sum of all length of contents in the catalog (17.7 GByte). From zero storage (0%) to complete storage (100%), we can observe that the users' demand

satisfaction improves monotonically and reaches up to 100%, both for the ground truth and collaborative filtering methods. Regardless, we observe a performance gap between the ground truth and CF until 87% of storage size, which is because of the estimation errors. For example, given 40% of storage size, the ground truth reachs 92% of demand satisfaction whereas the CF stays at 69%.

The variation of backhaul load/usage with respect to the storage size of base stations is depicted in Fig. 6. As the storage capacity of SBSs increases, we see a higher backhaul load reduction in both approaches. For instance, given 87% of storage capacity for caching, both methods offload 98% of backhaul load. The performance in ground truth is obviously better than the CF as all the existing information from the traces is taken into account for caching. We also observe that there is a sharp decrease of backhaul load in both methods after a specific storage size. Indeed, most of earlier works on caching suppose a content catalog with identical content lenghts. In our study, we are considering real traces where the length of contents varies from content to content, as mentioned earlier (see Fig. 4). According to this setup, on the one hand, storing a very popular content with very small size might not radically reduce the backhaul load. On the other hand, storing a relatively less popular content of very high size can dramatically reduce the backhaul load. Hence, as the CF method here is purely based on content popularity, it fails to consider these content length aspects on the backhaul load, which therefore ends up in more storage requirements to achieve the exact same performance as in the ground truth. This points out the importance of size distribution of popular contents.

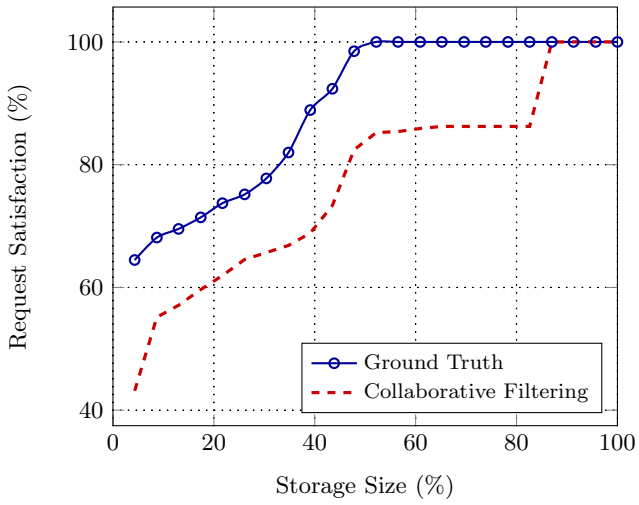

Figure 5. Evolution of satisfaction with respect to the storage size.

We have so far detailed the performance boosts of these approaches for 10% of rating density in CF. Indeed, as the rating density for training is increased, we expect to experience smaller estimation errors, therefore having closer satisfaction gains to the ground truth. To elaborate this, the variation of root-mean-square error (RMSE) as a function of the training density is shown in Fig. 7. For the figure, we define the error as the root-mean-square of difference of users' content satisfaction of the ground truth and CF methods,

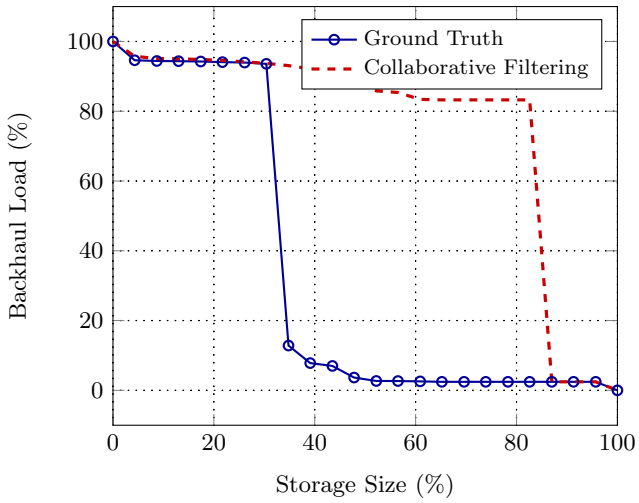

Figure 6. Evolution of backhaul usage with respect to the storage size.

over all possible storage capacities. As evidenced in 7, the performance of CF is improved by increasing the rating density, thus confirming our intuitions.

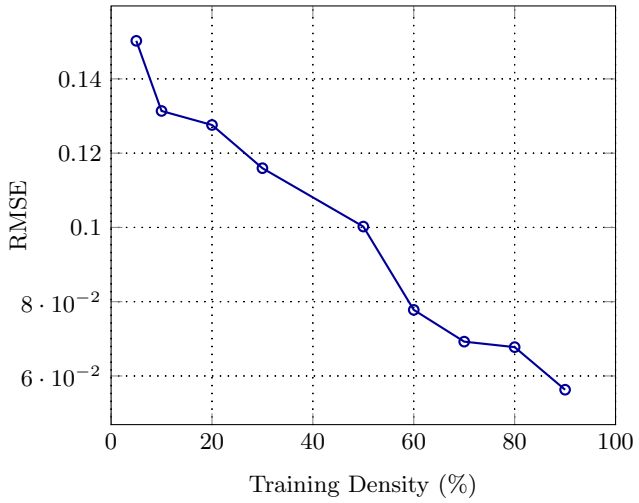

Figure 7. Evolution of RMSE with respect to the training density.

VI. Conclusions

This work studied a proactive caching method for $5G$ (and beyond) wireless networks by leveraging large amount of available data using machine learning techniques. Specifically, an experimental tested for data gathering/extraction procedures has been demonstrated on a big data platform and tools from machine learning (CF in particular) have been used to estimate the content popularity distribution. Depending on the storage capacity and rating density, the simulation results showed that several caching gains in terms of users' demand satisfaction and backhaul offloadings are possible.

An interesting direction of this work would be to asses a more detailed characterization of the mobile data traffic that takes into account spatio-temporal content access patterns. For the estimation of content access patterns for cache placement, the design of novel machine learning mechanisms is yet another line of work. In addition, development of new deterministic/randomized cache placement algorithms are needed and should consider not only content popularity but also content lengths and other factors, therefore higher backhaul offloading and demand satisfaction can be achieved. Finally, another line of work would be to extend the big data analysis framework in a real-time fashion. In this regard, the recent tools of Hadoop ecosystem, like Apache Spark and its built-in libraries Spark Streaming, could be considered as well as MLLib for machine learning analysis.

VII. Acknowledgement

This research has been supported by the ERC Starting Grant 305123 MORE (Advanced Mathematical Tools for Complex Network Engineering), the SHARING project under the Finland grant 128010 and TUBITAK TEYDEB 1509 project grant, numbered 9120067 and the project BESTCOM. Some of results here has appeared in parts in [36]–[38].

References

[1] C. Lynch, "Big data: How do your data grow?" *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.

[2] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the Edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82 – 89, August 2014.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[4] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *[Online] arXiv:1502.01815*, 2015.

[5] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[6] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[7] N. Farrington and A. Andreyev, "Facebook's data center network architecture," in *Proceedings of IEEE Optical Interconnects Conference*, CA, USA, May 2013.

[8] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," in *IEEE Wireless Communications and Networking Conference (WCNC'2014)*. IEEE, 2014, pp. 2300–2305.

[9] B. Zhou, Y. Cui, and M. Tao, "Optimal dynamic multicast scheduling for cache-enabled content-centric wireless networks," *[Online] arXiv:1504.04428*, 2015.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, vol. 1. IEEE, 1999, pp. 126–134.

[11] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 305–316.

[12] J. Tadrous, A. Eryilmaz, and H. E. Gamal, "Proactive data download and user demand shaping for data networks," *submitted to IEEE Transactions on Information Theory [Online] arXiv: 1304.5745*, 2014.

[13] Google, "Recommended upload encoding settings (Advanced)," https://goo.gl/KJXfhh, 2015, [Online; accessed 30-August-2015].

[14] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[15] E. Baştuğ, J.-L. Guénégo, and M. Debbah, "Proactive small cell networks," in *20th International Conference on Telecommunications (ICT'13)*, Casablanca, Morocco, May 2013.

[16] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3665–3677, October 2014.

[17] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, August 2009.

[19] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *[Online] arXiv: 1205.3193*, 2012.

[20] "Apache Hadoop," http://hadoop.apache.org/, 2015, [Online; accessed 02-April-2015].

[21] "Cloudera," http://www.cloudera.com/content/cloudera/en/documentation.html, 2015, [Online; accessed 02-April-2015].

[22] "The Wireshark Network Analyzer 1.12.2," https://www.wireshark.org/docs/man-pages/tshark.html, 2015, [Online; accessed 02-April-2015].

[23] "Apache Hive TM," https://hive.apache.org/, 2015, [Online; accessed 02-April-2015].

[24] Apache, "HttpClient API Tutorial," https://hc.apache.org/httpcomponents-client-ga/tutorial/pdf/httpclient-tutorial.pdf, 2015, [Online; accessed 25-April-2015].

[25] Y. Dong, Q. Ke, Y. Cai, B. Wu, and B. Wang, "Teledata: data mining, social network analysis and statistics analysis system based on cloud computing in telecommunication industry," in *Proceedings of the third international workshop on Cloud data management*. ACM, 2011, pp. 41–48.

[26] H.-D. J. Jeong, W. Hyun, J. Lim, and I. You, "Anomaly teletraffic intrusion detection systems on hadoop-based platforms: A survey of some problems and solutions," in *15th International Conference on Network-Based Information Systems (NBiS)*. IEEE, 2012, pp. 766–770.

[27] J. Magnusson and T. Kvernvik, "Subscriber classification within telecom networks utilizing big data technologies and machine learning," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. ACM, 2012, pp. 77–84.

[28] W. Indyk, T. Kajdanowicz, P. Kazienko, and S. Plamowski, "Mapreduce approach to collective classification for networks," in *Artificial Intelligence and Soft Computing*. Springer, 2012, pp. 656–663.

[29] O. F. Celebi, E. Zeydan, O. F. Kurt, O. Dedeoglu, O. Ileri, B. A. Sungur, A. Akan, and S. Ergut, "On use of big data for enhancing network coverage analysis," in *20th International Conference on Telecommunications (ICT'13)*, Casablanca, Morocco, May 2013.

[30] I. A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *Proceedings of European Wireless 2014*. VDE, 2014, pp. 1–5.

[31] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 1–14.

[32] D. Rossi, G. Rossini *et al.*, "On sizing ccn content stores by exploiting topological information." in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2012, pp. 280–285.

[33] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network–measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[34] E. Mucelli Rezende Oliveira, A. Carneiro Viana, K. P. Naveen, and C. Sarraute, "Measurement-driven mobile data traffic modeling in a large metropolitan area," INRIA, Research Report RR-8613, October 2014.

[35] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, pp. 5–8.

[36] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36–42, 2016.

[37] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data meets telcos: A proactive caching perspective," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 549–557, 2015.

[38] M. A. Kader, E. Bastug, M. Bennis, E. Zeydan, A. Karatepe, A. S. Er, and M. Debbah, "Leveraging big data analytics for cache-enabled wireless networks," in *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE, 2015, pp. 1–6.