AN INVESTIGATION OF MOBILE NETWORK TRAFFIC DATA AND APACHE HADOOP PERFORMANCE

Man Si, Chung-Horng Lung, Samuel Ajila
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada, K1S 5B6
mansi@email.carleton.ca, {chlung, ajila}@sce.carleton.ca

Wayne Ding
RAN System, Business Unit Radio
Ericsson, Canada
Ottawa, Ontario, Canada, K2K 2V6
wayne.ding@ericsson.com

Abstract

Since the emergence of mobile networks, the number of mobile subscriptions has continued to increase year after year. To efficiently assign mobile network resources such as spectrum (which is expensive), the network operator needs to critically process and analyze information and develop statistics about each base station and the traffic that passes through it. This paper presents an application of data analytics by focusing on processing and analyzing two datasets from a commercial trial mobile network. A detailed description that uses Apache Hadoop and the Mahout Machine learning library to process and analyze the datasets is presented. The analysis provides insights about the resource usage of network devices. This information is of great importance to network operators for efficient and effective management of resources and for supporting high-quality of user experience. Furthermore, an investigation has been conducted that evaluates the impact of executing the Mahout clustering algorithms with various system and workload parameters on a Hadoop cluster. The results demonstrate the value of performance data analysis. Specifically, the execution time can be significantly reduced using data pre-processing, some machine learning techniques, and Hadoop. The investigation provides useful information for the network operators for future real-time data analytics.

Keywords: Mobile network traffic; clustering; Principal Component Analysis (PCA); real-time data analytics; Hadoop; Mahout

1. INTRODUCTION

With the emergence of mobile networks, more and more users are connected to the network and have access to the Internet through their mobile devices (Ericsson, 2015). Each connected user occupies the limited radio resources of a mobile network (e.g. spectrum) for a certain period of time. Since the radio resources are very expensive, mobile network operators spend a great deal of time and money to determine how to efficiently allocate these resources. More specifically, the mobile network operators aim to maximize profits and ensure that the users are satisfied with the quality of service (QoS). To support resource allocation, the mobile network operators collect the statistics of the traffic (e.g. throughput and average number of connections/cell) from base stations. This information can be analyzed to determine how the resources are used and to find traffic patterns from the data. However, the base stations collect a wide variety of information for a long period, and hence the amount of data collected becomes vast. Analyzing such a large amount of data becomes a challenge and this problem can be solved efficiently by using Big Data Analytics approach and techniques.
Hadoop (Apache, 2015) and its machine learning library called Mahout (Hortonworks, 2015) (discussed more in Section 2.1) are well-known techniques used for analyzing big data. This research focuses on exploring and comparing the popular clustering algorithms of Mahout to analyze real-life base station data from a commercial trial mobile network. In addition, Hadoop parameters, such as the number of Map tasks, the number of Reduce tasks, need to be tuned to achieve the optimal performance when running machine learning techniques, e.g., K-means clustering, on a Hadoop cluster. The experiments are performed on a Hadoop cluster composed of Virtual Machines within a server.

There are two main motivating factors for this research work:

- Growth of mobile subscriptions brings a big challenge for mobile network operators to provide a robust and scalable wireless network. However, a mobile network operator has limited wireless resources. Thus, they need to devise efficient and effective techniques to allocate these limited wireless resources.

- System data collected by base stations consists of valuable information that can reveal the current state and quality of the system. The amount of data that needs to be analyzed to find meaningful information and useful patterns can be substantial. However, the traditional data analysis techniques that use a single machine cannot efficiently process the large volume of data.

To address these two problems, this paper presents an approach to processing and analyzing mobile network data for more than 5000 base stations. The approach consists of multiple steps, including identification of the relationship between features, using Principal Component Analysis (PCA) (Fodor, 2002) for feature reduction, and application of clustering algorithms for categorizing data records.

Therefore, this paper presents an empirical investigation on applying data analytics techniques. The main contributions of this paper include the following. First, the paper applies Mahout K-means and fuzzy K-means algorithms on the experimental datasets to group together data records that have similar attributes. Second, experiments have been performed using PCA to reduce execution time as a result of feature reduction, while still maintaining an accurate clustering solution. Third, a detailed investigation of a critical feature to mobile network operators: the Average RRC (Radio Resource Control) Connections/Cell that specifies the average number of user connected to the base station. The analysis reveals how the Average RRC Connections/Cell changes throughout a 24-hour period and how the base station cells are affected. The study provides useful information for the network operator and for future research on data analytics, including real-time data analytics.

The rest of the paper is organized as follows. Section 2 describes the background information and related work. Section 3 discusses the proposed data analysis procedure. Section 4 presents and discusses the experimental results. Lastly, Section 5 concludes the paper and outlines directions for future work.

2. BACKGROUND AND RELATED WORK

2.1 HADOOP AND MAHOUT

Apache Hadoop (Apache, 2015) is a software platform used for analyzing and processing large datasets. It is an open-source software framework implemented using Java and allows for the distributed processing of large datasets using a cluster of computers. Hadoop consists of two core components: the Hadoop Distributed File System (HDFS) and the MapReduce Software Framework. HDFS is a distributed file system that runs on a cluster of commodity hardware and is designed to store very large files (White, 2010). On the other hand, MapReduce (Dean & Ghemawat, 2004) is a programming model and technique for distributing a job across multiple nodes. It works by separating the processing into two phases, the Map phase and Reduce phase. The Map, with processing function Mapper, takes original input data and produces intermediate data as the input for the Reduce phase. In the Reduce phase, the function Reducer accepts intermediate data and merges together those intermediate values which have the same key.

Mahout (Hortonworks, 2015) is a machine learning library built on top of the Hadoop platform and uses the MapReduce paradigm. Mahout contains a collection of algorithms to solve recommendation, classification and clustering problems. Mahout contains the implementation of various clustering algorithms, including K-means, fuzzy K-means and canopy clustering.
2.2 CLUSTERING ANALYSIS AND K-MEANS

Clustering analysis (Tan et al., 2006) refers to the process of organizing items into groups based on their similarity (Owen et al., 2012). The generated clusters consist of a set of items that are similar to one another in the same group but dissimilar from the items belonging to other groups. Clustering analysis is regarded as a form of classification since both techniques are used to divide items into groups. However, clustering analysis categorizes unlabeled items based only on the data and is thus referred to as unsupervised classification. On the contrary, classification techniques which assign new class labels to unlabeled items based on a labeled model are regarded as supervised classification.

One of the advantages of clustering analysis is that it can be used to help provide meaningful and useful insights into the data. Some applications use clustering analysis to achieve a better understanding of their dataset. For example, clustering is used in software evolution to help reduce legacy properties in code; in image segmentation to divide digital image into distinct regions; in recommender systems to recommend new items based on user’s tastes; and in business to summarize dataset.

One of the most well-known and widely used clustering algorithms is called K-means (Tan et al., 2006). In K-means algorithm, first, the user chooses K initial centroids, where K is the number of clusters. The K-means algorithm centroid is defined as the mean of a group of points. Step 2 is to assign each point to the cluster which has the nearest centroid to this point. Closest is quantified by a proximity measure such as Euclidean distance, Manhattan distance and cosine similarity (Owen et al., 2012). Thirdly, the centroids are then updated using the mean of the points assigned to a cluster. Next, steps 2 and 3 are repeated until the result converges which means that the centroid values do not change in subsequent iterations.

2.3 RELATED WORK

This section presents related work on analyzing mobile network traffic data collected from a trial network. With the popularity of mobile devices, more and more users are accessing the Internet using mobile devices (e.g. phones and tablets) constantly. To improve the performance of wireless networks and the development of the 5G network, network operators need to monitor and analyze the network traffic.

Yang et al. (2014) analyzed the mobile Internet traffic data collected from a capital city in southern China and developed a Hadoop-based system, Traffic Analysis System (TAS). Their analysis shows the basic make up of current mobile devices and users’ interests via the available network providers. Tang and Baker (2000) examined a twelve-week trace of a building-wide local-area wireless network. They analyzed the overall user behavior such as how many users are active at a time and how much users move between access points. The analysis can help determine how wireless hardware and software should be optimized to handle traffic generated for this specific local-area network.

Esteves and Rong (2011) compared K-means clustering and fuzzy K-means clustering algorithms in Mahout to perform clustering of Wikipedia’s latest articles. Based on their research, the authors commented that Mahout is a promising tool for performing clustering but the preprocessing tools need to be further developed. Esteves et al. (2011) studied the performance of Mahout by using a large dataset comprising of tcpdump (a common packet analyzer) data collected from a US Air Force local area network (LAN). They tested the scalability of Mahout with regards to the data size and also compared the performance of processing the dataset on a single node versus multiple nodes.

Comparison with Related Work. The data used for this research was statistical traffic data from a commercial mobile network. Unlike the data analyzed in (Yang et al., 2014) and (Tang & Baker 2000), which do not have base station information, experiment datasets in this paper include records from more than 5000 base stations. Analyze this large volume of data with multiple dimensions from different base stations is a challenge.

Esteves and Rong (2011) and Esteves et al. (2011) tested the Mahout clustering methods on Amazon EC2. In this research, we performed additional experiments using a network traffic dataset to compare the performance of Mahout’s K-means and fuzzy K-means clustering algorithms (see Section 4.3). Furthermore, this research also varies the number of Map and Reduce tasks in a Hadoop job, and it also varies the number of cores assigned to slave nodes, which were not investigated by (Esteves & Rong, 2011) or (Esteves et al., 2011)
3. METHODOLOGY AND APPROACH

Section 3.1 describes the base station datasets that are analyzed in this research. Following that, in Section 3.2, the proposed data processing and analysis procedure is presented.

3.1 DESCRIPTION OF DATASETS

Two datasets have been collected from a commercial trial mobile network.

Description of Dataset 1 (2015 dataset). The first dataset, referred to as the Dataset 1 (236 MB), is anonymized mobile network traffic data collected over a one-week period from January 25, 2015 to January 31, 2015. The data was collected from 5840 unique cells in a region. Each row is a record which contains the unique Cell ID and the values of the 24 features (e.g., downlink/uplink throughput and average number of connected users) collected every hour for one week. Therefore, each unique cell should have 168 records (24 hours per day × 7 days per week). However, some cells have missing records (i.e., data was not collected for certain time slots). To fill in these missing values so that further analysis (e.g., clustering) can be performed, the average value of the particular column (time slot) of where the missing value is located is used in place of the missing value.

An example of UE is a mobile phone. PDCP is one of the user plane protocols used in LTE mobile networks. This protocol sends and receives packets from User Equipment and eNodeB (abbreviation for Evolved Node B), which is the hardware used to communicate directly with UEs in a mobile phone network. There are two kinds of PDCP bearers, which are data carriers in LTE systems, the SRB and DRB. SRB is used for carrying signals and DRB is used to carry User Plane content on the Air Interface.

Table 1 presents the 24 features used in the 2015 dataset including the unit used for each feature. The following list outlines the abbreviations used in Table 1:

- UE: User Equipment
- DL: Downlink
- UL: Uplink
- PDCP: Packet Data Convergence Protocol
- RRC: Radio Resource Control
- DRB: Data Radio Bearer
- SRB: Signal Radio Bearer

The Average RRC Connections is calculated by dividing the total number of RRC Connections with the number of samples.

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ActiveUeDlSum</td>
<td>Number of Active UE at DL</td>
</tr>
<tr>
<td>2</td>
<td>ActiveUeUlSum</td>
<td>Number of Active UE at UL</td>
</tr>
<tr>
<td>3</td>
<td>PdcpBitrateDlDrb Max</td>
<td>Max PDCP DRB Bit Rate in Mbps at DL</td>
</tr>
<tr>
<td>4</td>
<td>PdcpBitrateDlDrb Min</td>
<td>Min PDCP DRB Bit Rate in Mbps at DL</td>
</tr>
<tr>
<td>5</td>
<td>PdcpBitrateUlDrb Max</td>
<td>Max PDCP DRB Bit Rate in Mbps at UL</td>
</tr>
<tr>
<td>6</td>
<td>PdcpBitrateUlDrb Min</td>
<td>Min PDCP DRB Bit Rate in Mbps at UL</td>
</tr>
<tr>
<td>7</td>
<td>PdcpLatPktTrans Dl</td>
<td>PDCP Latency for Packet Transmission at DL</td>
</tr>
<tr>
<td>8</td>
<td>PdcpLatTimeDl</td>
<td>PDCP Latency Time</td>
</tr>
<tr>
<td>9</td>
<td>PdcpPktReceived Dl</td>
<td>Number of PDCP Packets Received at DL</td>
</tr>
<tr>
<td>10</td>
<td>PdcpPktReceived Ul</td>
<td>Number of PDCP Packets Received at UL</td>
</tr>
<tr>
<td>11</td>
<td>PdcpVolDlDrb</td>
<td>PDCP DRB Volume at DL</td>
</tr>
<tr>
<td>12</td>
<td>PdcpVolDlDrbLas rTTI</td>
<td>PDCP DRB Volume at DL for the Last TTI</td>
</tr>
<tr>
<td>13</td>
<td>PdcpVolDlSrb</td>
<td>PDCP SRB Volume at DL</td>
</tr>
<tr>
<td>14</td>
<td>PdcpVolUlDrb</td>
<td>PDCP DRB Volume at UL</td>
</tr>
<tr>
<td>15</td>
<td>PdcpVolUlSrb</td>
<td>PDCP SRB Volume at UL</td>
</tr>
<tr>
<td>16</td>
<td>AvgRrcConn/Cell</td>
<td>Number of Average RRC Connections/Cell</td>
</tr>
<tr>
<td>17</td>
<td>RccConnMax</td>
<td>Number of Peak Connected Users/Cell</td>
</tr>
<tr>
<td>18</td>
<td>SchedActivityCell Dl</td>
<td>Scheduling Time in Seconds per Cell at DL</td>
</tr>
<tr>
<td>19</td>
<td>SchedActivityCell Ul</td>
<td>Scheduling Time in Seconds per Cell at UL</td>
</tr>
<tr>
<td>20</td>
<td>SchedActivityUe Dl</td>
<td>Scheduling Time in Seconds per UE at DL</td>
</tr>
<tr>
<td>21</td>
<td>SchedActivityUe Ul</td>
<td>Scheduling Time in Seconds per UE at UL</td>
</tr>
<tr>
<td>22</td>
<td>UeThpTimeDl</td>
<td>UE Throughput Time at DL</td>
</tr>
<tr>
<td>23</td>
<td>UeThpTimeUl</td>
<td>UE Throughput Time at UL</td>
</tr>
<tr>
<td>24</td>
<td>UeThpVolU</td>
<td>UE Throughput Volume at UL</td>
</tr>
</tbody>
</table>

Table 1. Feature description for Dataset 1

Description of Dataset 2 (2013 dataset). The second dataset, which is referred to as the Dataset 2 (24 MB), is also anonymized mobile network traffic data. The dataset has a similar format as the Dataset 1. The records were collected from 7592 unique cells in a region. The data for these unique cells were collected every 15 minutes in the morning from 6:00 to 7:00 and in the afternoon from 14:00 to 15:00 on November 18, 2013. At each time slot, data for 17 features, including the average number of connected users, are recorded. The description of the 17 features in the 2013 dataset is presented in Table 2.

The following list outlines the abbreviations used in Table 2:
- UE: User Equipment
- TTI: Transmission Time Interval
- DL: Downlink
- UL: Uplink
- RRC: Radio Resource Control
- DRB: Data Radio Bearer
- SRB: Signal Radio Bearer

TTI is a parameter used in digital communication networks that refers to the duration of a transmission on the radio link. Occupancy denotes how much percentage the TTI has been occupied for data transmission. Note that the other terms, which are also used in the 2015 dataset (see Table 1) are explained in the section named “Description of Dataset 1”.

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ActiveUeDlMax/TTI/Cell</td>
<td>Number of Maximum Active UE at DL/TTI/Cell</td>
</tr>
<tr>
<td>2</td>
<td>ActiveUeUlMax/TTI/Cell</td>
<td>Number of Maximum Active UE at UL/TTI/Cell</td>
</tr>
<tr>
<td>3</td>
<td>RrcConnMax/Cell</td>
<td>Number of Maximum RRC Connection/Cell</td>
</tr>
<tr>
<td>4</td>
<td>RrcConnAvg/Cell</td>
<td>Number of Average RRC Connection/Cell</td>
</tr>
<tr>
<td>5</td>
<td>ActUeDl/TTI (incl. Idle time)</td>
<td>Number of Active UE at DL/TTI (include idle time)</td>
</tr>
<tr>
<td>6</td>
<td>ActUeUl/TTI (incl. Idle time)</td>
<td>Number of Active UE at UL/TTI (include idle time)</td>
</tr>
<tr>
<td>7</td>
<td>ActUeDl/TTI (SE/TTI)</td>
<td>Number of Active UE at DL/TTI (Schedule Entity/TTI)</td>
</tr>
<tr>
<td>8</td>
<td>ActUeUl/TTI (SE/TTI)</td>
<td>Number of Active UE at UL/TTI (Schedule Entity/TTI)</td>
</tr>
<tr>
<td>9</td>
<td>DlDrbCellTput</td>
<td>DL DRB Cell Throughput in Mbps</td>
</tr>
<tr>
<td>10</td>
<td>UlDrbCellTput</td>
<td>UL DRB Cell Throughput in Mbps</td>
</tr>
<tr>
<td>11</td>
<td>DlDrbUeTput</td>
<td>DL DRB UE Throughput in Mbps</td>
</tr>
<tr>
<td>12</td>
<td>UlDrbUeTput</td>
<td>UL DRB UE Throughput in Mbps</td>
</tr>
<tr>
<td>13</td>
<td>DlTtiOccupancy</td>
<td>DL TTI Occupancy</td>
</tr>
<tr>
<td>14</td>
<td>UlTtiOccupancy</td>
<td>UL TTI Occupancy</td>
</tr>
<tr>
<td>15</td>
<td>TotalAvgTput (Vol/ROP)</td>
<td>Total Average Throughput Volume/Recording Output</td>
</tr>
<tr>
<td>16</td>
<td>AvgServiceLengt h(s)/CU</td>
<td>Average Service Length/Connected User</td>
</tr>
<tr>
<td>17</td>
<td>AvgSessionLengt h(s)/CU</td>
<td>Average Session Length/Connected User</td>
</tr>
</tbody>
</table>

Table 2. Feature description of Dataset 2

3.2 DATA ANALYSIS PROEDURE

This paper applies a sequence of steps to analyze the mobile network traffic datasets. The procedure shown in Algorithm 1 outlines the key steps.

**Data Preprocessing.** The features in the experimental dataset have different units. Before applying any analysis method (e.g. checking the correlation of different features or clustering the dataset), we first applied a standardization method (Kuhn, 2015) to eliminate the effect of different parameter units. Without the same scale, different parameter units will cause an unfair comparison when data mining techniques is applied, for example, clustering algorithm. A standardized score is calculated using the following formula in R, a data analysis tool (R Core Team, 2015):

\[
Z = \frac{x - \mu}{\sigma}
\]

where Z is the standardized score, \( \mu \) is the value of an element, \( \mu \) is the mean value of all elements, and \( \sigma \) is the standard deviation. After standardization, we get a dataset with standardized data where each feature has the same weight.

1: Preprocess data (Standardization)
2: Find feature correlation (using R package)
3: If analyzing with all features then
4: Apply K-means and fuzzy K-means clustering method
5: Else if analyzing with reduced feature dimension then
6: Apply PCA
7: Apply K-means clustering method
8: Else if individual feature analysis then
9: Select one feature at a time
10: Convert the data format (include handling missing time slots)
11: Apply K-means clustering method
12: End if
13: Evaluate and analyze results

Algorithm 1. Data analysis procedure

**Identification of Feature Correlation.** In order to have a better understanding of the relationship between different features, the next step is to calculate the correlation between each pair of features by using the cor() function from the ‘stats’ package of R (R Core Team, 2015).

**Applying Clustering Methods.** Clustering is used to organize items into groups from a large set of items based on their similarity. The steps of applying Mahout K-means and fuzzy K-means clustering to the dataset are shown in Algorithm 2.
Retrieval of Clustering Results. The final results of a clustering job are stored in files located in HDFS. To retrieve the clustering results, two commands are used: seqdumper and clusterdump.

1: Prepare data file and convert the data format to Mahout Vector
2: Convert Mahout Vector to Hadoop SequenceFile
3: Randomly select K centroids
4: Copy the SequenceFile to HDFS
5: Set Hadoop configuration parameters: NameNode IP
6: Set K-means clustering parameters: threshold
7: Run Mahout K-means clustering
8: Use ClusterDumper/SequenceFileDumper to get the clustering result

Algorithm 2. Steps of applying Mahout K-means Clustering

Evaluation of Clustering Quality. Two common metrics provided by the Mahout Library to evaluate clustering quality are: (1) scaled average inter-cluster distance (can be regarded as separation) and (2) scaled average intra-cluster distance (cohesion). A large scaled average inter-cluster distance represents good clustering quality, since good clusters usually do not have centroids that are close to each other. Scaled average intra-cluster distance computes the distance between members within a cluster. A small intra-cluster distance value shows that the points within the cluster are close to one another (i.e. a cohesive cluster), which is desired. Some research has used one or both of the two metrics separately to evaluate the clustering quality (e.g., Esteves & Rong 2011). In this paper, we propose a new metric, called the validity metric, that combines the two evaluation metrics together, as shown in Eq. (2):

$$\text{validity} = \frac{\text{scaled average inter-cluster distance}}{\text{scaled average intra-cluster distance}}$$  \hspace{1cm} (2)

A large validity value is desirable, since a large inter-distance and a small intra-distance indicate a good quality clustering result.

Dimension Reduction Using Principal Component Analysis (PCA) - Dimension reduction is the process of reducing the number of features (or dimensions) in a dataset. With dimension reduction, the original data is mapped from a high-dimensional space to a lower dimensional space to simplify the analysis of the dataset. This can remove redundant information and noise in the original dataset that can impact the analysis results. PCA is a well-known linear dimension reduction technique (Fodor, 2002). It reduces the dimension of a dataset by finding a few principal components with large variances. Principal components (PCs) are orthogonal and generated by a series of linear combinations of the variables.

Feature-by-feature analysis - allows us to see the distribution of the base stations from just one feature. This can be useful because mobile network service providers sometimes are interested in one specific feature and want to see the behavior of base stations connected to that feature. Furthermore, feature-by-feature analysis can provide a more detailed result compared to using values of all features in the same experiment.

To perform feature-by-feature analysis, we develop an algorithm to convert the data into an appropriate format. The goal is to transpose the start time of the records. That is, the start time for each traffic trace now becomes the feature of the record. Thus, by analyzing the converted dataset, we can analyze how one feature changes at different points in time.

4. EXPERIMENTATION AND ANALYSIS

4.1 ANALYSIS OF THE CORRELATION OF FEATURES

To analyze the correlation of features, a correlation matrix is generated using the R data analysis tool (R Core Team, 2015).

Correlation Matrix for Dataset 1. The correlation matrix for Dataset 1 is displayed in Figure 1. The color blue indicates positive correlation, and the color red indicates negative correlation. The darker the color (or the larger the circle size and the value) the stronger the correlation. The result shows that many of the features in Dataset 1 are strongly correlated (e.g. X9, X10 and X11).

Correlation Matrix for Dataset 2. The correlation matrix for Dataset 2 is illustrated in Figure 2. The results show that features X3 (Maximum RRC Connection/Cell) and X4 (Average RRC Connection/Cell), and features X1 (Active UE at DL/TTI/Cell) and X5 (Active UE UL/TTI) are observed to have a strong positive correlation. Conversely, it is observed that some feature pairs (e.g. X1 and X 11, and X11 and X3) are negatively correlated. X11 represents the downlink UE throughput and the negative relationship means that as the number of active UE (or the RRC Connections)
increases, downlink UE throughput decreases. There are also some feature pairs that are loosely correlated or have almost no correlation (e.g. X7 and X8), and thus have a small absolute correlation value. X7 and X8 represent the number of active users at downlink and uplink, respectively. The features X7 and X8 are not strongly correlated, which indicates that downlink and uplink channels are independent in terms of user numbers.

As outlined in Table 3, five virtual machines (VMs) are deployed on the server. The specification of each VM is outlined in Table 3. All the VMs have the same virtual disk size equal to 100.73 GB, and each VM also runs Ubuntu 14.04 (64-bit) operating system.

<table>
<thead>
<tr>
<th>VM ID</th>
<th>Type</th>
<th>Number of Virtual CPUs</th>
<th>RAM (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Master Node (NameNode +JobTracker)</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>Slave Node A (DataNode +TaskTracker)</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Slave Node B (DataNode +TaskTracker)</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Slave Node C (DataNode +TaskTracker)</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Slave Node D (DataNode +TaskTracker)</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 3. VM specifications inside host server

The Hadoop daemons are executed on these VMs, and thus each VM is considered a node in the Hadoop cluster. VM1 operates as the Master Node of the Hadoop cluster and therefore executes the NameNode and JobTracker. The other VMs, VM2 to VM5, are considered the Slave Nodes and run the DataNode and TaskTracker Hadoop daemons. Note that the Hadoop cluster used in the experiments comprises of one Master Node and one Slave Node (which can be VM2, VM3, VM4, or VM5). Each of the Slave Node VMs is configured to have a \( \frac{n}{2} \) map slots and \( \frac{n}{2} \) reduce slots, which determine the number of map tasks and reduce tasks the Slave Node can execute in parallel at a point in time. The value of \( n \) equals the total number of cores the VM has. For example, VM2 is configured with 4 map and 4 reduce slots because it has 8 cores.

4.3 Choosing Appropriate K and Fuzzy Values

This section presents and analyzes the clustering results of Dataset 1 and Dataset 2. K-means (Tan et al., 2006) and fuzzy k-means (Tan et al., 2006) techniques are used for clustering the experimental datasets in this paper. Note that the inputs to the clustering techniques are the preprocessed versions of Dataset 1 and Dataset 2. The standardization technique that is used to preprocess the datasets is detailed in Section 3.2. We vary the K value for the
k-means technique to determine the K value that achieves the best validity value (refer to Eq. (3.2)). Furthermore, additional experiments (which use the best K value identified in the previous experiment) are conducted to find the value of the fuzziness parameter for the fuzzy k-means technique that generates the best validity value.

The Hadoop cluster used in this section is composed of the Master Node and the Slave Node D (with 8 CPUs as shown in Table 3). The convergence threshold of the k-means clustering algorithm is set to 0.01, which means that the difference between the last two iterations has to be less than 1% for the algorithm to stop. However, the maximum number of iterations is set to 150, hence even if the algorithm does not converge based on the threshold after 150 iterations the algorithm with still stop. Note that the number of iterations is set to 150 so that the experiments can complete in a reasonable amount of time if the algorithm does not converge. These configuration values are the same as those used in (Esteves et al., 2011).

In the experiments conducted, the K value varies from 3 to 10 (see Table 4). For each experiment the following metrics are collected or calculated: scaled average inter-cluster distance (denoted as inter-cluster density), scaled average intra-cluster distance (indicated as avg. intra-cluster density), the validity value (calculated using Eq. (3.2), the number of iterations, and the execution time of each iteration). Since the initial centroids are selected randomly, each experiment is run three times and the average value of each metric is calculated. Running the experiments three times was adequate, because less than 5% variation was observed in the final results and all three runs converge to a similar solution.

K-means Result of Dataset 1. This section applies k-means clustering method to Dataset 1. As shown in Table 4, as the K value (No. of clusters) increases, it is observed that in general the value of Inter-cluster density tends to decrease. The largest inter-cluster density, which means that the clusters have good separation, is achieved when K=4. The value of intra-cluster density decreases as K increases, except when K increases from 8 to 9. A smaller value for intra-cluster density indicates higher cohesion. The largest validity value (defined in 3.2) is reached when K=4. Thus K=4 is the most appropriate K value for Dataset 1. As shown in Table 4, it requires 35 iterations for the k-means algorithm to converge when K=4. Lastly, it is also observed that the execution time for each iteration increases with an increase in K. This is expected because k-means requires calculating the distance between each data point and each cluster center. Therefore, a higher value of K (clusters) means that more calculations need to be performed causing a higher execution time.

Fuzzy K-means Results of Dataset 1. The experiments in this section apply fuzzy k-means to Dataset 1 to find the fuzziness parameter that obtains the highest validity value. It can be observed from Table 4 that K=4 achieves the highest validity value when k-means is used to cluster Dataset 1. Therefore, K=4 is chosen for the fuzzy k-means algorithm used in this section which aims to find the value of the fuzziness parameter that achieves the highest validity value. The fuzziness parameter varies from 1.0 to 1.9. The experimental results of applying fuzzy k-means clustering to Dataset 1 are displayed in Table 5. The largest validity value is generated when fuzziness=1.0. Note that using fuzzy k-means with fuzziness=1.0 is identical to using normal k-means clustering 0. This result indicates that k-means clustering with K=4 mostly fits Dataset 1, meaning that the data points in Dataset 1 can be divided into four well-defined clusters. By comparing the execution time for fuzzy k-means in Table 5 and k-means in Table 4, it is observed that the execution time per iteration for fuzzy k-means is longer compared to k-means. This is reasonable, because the complexity of the fuzzy k-means algorithm is more than the k-means algorithm.

### Table 4. K-means results of Dataset 1

<table>
<thead>
<tr>
<th>K</th>
<th>Inter-CD</th>
<th>Intra-CD</th>
<th>Validity</th>
<th>N</th>
<th>Execution Time(s) / Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.501</td>
<td>0.673</td>
<td>0.744</td>
<td>35</td>
<td>67.05</td>
</tr>
<tr>
<td>1.1</td>
<td>0.485</td>
<td>0.671</td>
<td>0.723</td>
<td>83</td>
<td>66.70</td>
</tr>
<tr>
<td>1.2</td>
<td>0.469</td>
<td>0.668</td>
<td>0.702</td>
<td>69</td>
<td>66.51</td>
</tr>
<tr>
<td>1.3</td>
<td>0.448</td>
<td>0.668</td>
<td>0.671</td>
<td>45</td>
<td>66.69</td>
</tr>
<tr>
<td>1.4</td>
<td>0.431</td>
<td>0.660</td>
<td>0.653</td>
<td>34</td>
<td>66.95</td>
</tr>
<tr>
<td>1.5</td>
<td>0.414</td>
<td>0.653</td>
<td>0.634</td>
<td>27</td>
<td>66.71</td>
</tr>
<tr>
<td>1.6</td>
<td>0.394</td>
<td>0.656</td>
<td>0.601</td>
<td>21</td>
<td>66.45</td>
</tr>
</tbody>
</table>
Table 5. Fuzzy K-means results of Dataset 1

<table>
<thead>
<tr>
<th>F</th>
<th>CD</th>
<th>N</th>
<th>Cluster ID</th>
<th>Fuzziness</th>
<th>CDensity</th>
<th>Nof Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7</td>
<td>0.380</td>
<td>0.658</td>
<td>0.578</td>
<td>17</td>
<td>66.49</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>0.382</td>
<td>0.684</td>
<td>0.558</td>
<td>14</td>
<td>66.69</td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td>0.394</td>
<td>0.614</td>
<td>0.642</td>
<td>45</td>
<td>66.79</td>
<td></td>
</tr>
</tbody>
</table>

F = fuzziness, CD = cluster density, N= no. of iterations

4.4 APPLYING PRINCIPAL COMPONENT ANALYSIS (PCA)

This section discusses the performance and accuracy of applying the PCA (Fodor, 2002) technique before clustering. Dataset 1 has 24 features and most of these features are strongly correlated. This phenomenon implies that redundant features exist in the dataset and thus processing the dataset with all the features, which requires a substantial amount of time for a very large data, is not required.

After applying PCA, the dataset will lose some information based on the number of Principal Components (PCs) that have been chosen. The cumulative proportion of variance (see Eq. (3)) is a way to measure this lost.

Cumulative proportion of variance: \( V = \sum_{i=1}^{n} \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i} \)

where \( \lambda \) is the variance and \( n \) indicates the total number of PCs. To determine the number of PCs to be used, we set the threshold of \( V \) to be 0.85 (Kerdprasop, 2005) and PCs to be 1 to 4 are retained as meaningful variables for further analysis.

To evaluate the performance of PCA as applied to our experimental datasets, an accuracy function (see Eq. (4)) is proposed. We have evaluated 8 K values (from 3 to 10, inclusive, cf. Table 4) for K-means clustering and selected K = 4 as it generates the highest validity value, see Eq. (2), compared to other values. Hence, K-means clustering with K = 4 is performed on Dataset 1 using PCA before clustering and without using PCA. In addition, our experimental results for K-means are better than that for fuzzy K-means, since the validity value for K-means when K=4 is higher than any of the results for fuzzy K-means (excepts when fuzzy number is equal to 1 which is identical to using normal K-means clustering). Hence, the rest of the paper focuses only on K-means.

Accuracy = \( \frac{\text{Total no. of correctly assigned records}}{\text{Total no. of input records}} \) (4)

Table 6 displays the results of two clustering experiments: A and B. Experiment A represents the clustering result of using all 24 features of the original Dataset 1 and experiment B denotes the clustering result of Dataset 1 using top 4 principal components (PCs) after using PCA. Since K=4 is chosen for both experiments, both experiments A and B generate 4 clusters. The percentage of overlapping data records in A and B is over 99.45% which is confirmation of the selection of K=4. The numbers displayed in the second, third, and fourth columns of Table 6 are the number of data records in each cluster. For example, there are a total of 978108 records in experiment A, 978108 records in B and 972738 records in records A∩B (intersection of A and B). A∩B shows the number of data records that are grouped in the same cluster in both experiment A and experiment B. In cluster #1, there are 4644 records in experiment A, 4662 records in B and 4527 records in A∩B which means that 4527 data records from cluster 1 in experiment A also appear in cluster 1 in experiment B.

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>A</th>
<th>B</th>
<th>A∩B</th>
<th>A∩B/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4644</td>
<td>4662</td>
<td>4527</td>
<td>97.48%</td>
</tr>
<tr>
<td>2</td>
<td>84566</td>
<td>85371</td>
<td>84052</td>
<td>99.39%</td>
</tr>
<tr>
<td>3</td>
<td>301373</td>
<td>302629</td>
<td>299442</td>
<td>99.36%</td>
</tr>
<tr>
<td>4</td>
<td>587525</td>
<td>585446</td>
<td>584717</td>
<td>99.52%</td>
</tr>
<tr>
<td>Total</td>
<td>978108</td>
<td>978108</td>
<td>972738</td>
<td>99.45%</td>
</tr>
</tbody>
</table>

Table 6. Comparison of clustering results without using PCA (A) and with using PCA (B)

As shown in Table 6, the number of correctly assigned records after applying PCA is 972738. What is “correct” is evaluated by comparing the results of B with the results of A, as experiment A is based on all 24 features (i.e. A∩B/A). Thus, the accuracy of applying PCA for feature reduction is 972738/978108 = 99.45%.

By Using PCA, the number of features in Dataset 1 was reduced from 24 to 4, which results in the input data size reduction from 236MB to 38MB. This in turn reduced the total execution time of a K-means clustering job from 1637.30s to 1046.08s, or 36.11% reduction. Extrapolating these figures (i.e., data size and execution time) over a bigger dataset (GB or TB) indicates a substantial gain in memory space and time. The savings in computation time can also be critical for real-time data analytics that could provide useful information rapidly to the network operator for potential better resource management.
4.5 Analysis of Impact of Avg. RRC Connections per Day

This section discusses the results of analyzing a single feature for a whole day in order to have a better understanding of how the feature value changes during the day and how the base station cells are affected by the feature. The feature that is chosen for analysis is “Feature 16” of Dataset 1 which is the average number of radio resource control (RRC) connections per cell. The reasons for choosing Feature 16 are as follows:

- Many features in Dataset 1 have a strong correlation with Feature 16 based on correlation analysis. Therefore, analyzing Feature 16 is representative of analyzing the other features of Dataset 1.
- Average RRC Connections/Cell is a crucial parameter for mobile network operators, since it can influence user experience and resource utilization.

The first step of this experiment is to select Feature 16 from Dataset 1 and convert it to generate the dataset of which each row has one unique base station ID and the values of Feature 16 collected every hour from the hour (0:00) to the end of the day (23:00). K-means clustering is then applied to the modified dataset. The K-means clustering result of Feature 16 with K = 4 is displayed in Table 7. K = 4 is selected due to its high validity value in comparison to other K values (cf. section 4.2 Table 4). The number of unique cells assigned to each cluster is shown in Table 7. Each cluster has a center vector that is calculated by averaging all of the points’ values in a cluster. By plotting and analyzing the 4 center vectors, with each having 24 dimensions (one for each hour of the day), we can have a better understanding of how the base station cells in different groups are affected by Feature 16 i.e. the Average RRC Connections/Cell, a dominant feature for resource usage.

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cells</td>
<td>71</td>
<td>604</td>
<td>1935</td>
<td>3230</td>
</tr>
</tbody>
</table>

Table 7. K-means clustering result of Feature 16 from Dataset 1 for K = 4.

Figure 3 shows the feature values of the 4 cluster centers at different times of the day. As seen in Figure 3, a large number of cells have a small number of average connected users (RRC Connections) (e.g., Cluster 4), and only 71 of 5840 cells have a large number of average connected users (see Cluster 1). As time changes from morning to night, we can see the number of average connected users fluctuates. For Cluster 1, the value varies a lot throughout the day. It reaches its minimum value of 45.93 at 5:00 in the morning and achieves its peak value of 194 at 15:00. This is reasonable since mobile network users are more active in the early afternoon compared to that in early morning. A valuable result obtained from the experiment is that those cells in this cluster may need to be monitored closely as the number of users can be high, which may have impact on resource management, system performance, and user experience. Identifying those cells becomes useful for the operator in network planning and management.

The values for Cluster 2 and Cluster 3, however, do not vary as much compared to that of Cluster 1, but they still follow a similar trend. Cluster 4 has the least amount of connected users both in the morning and in the afternoon. This indicates that the cells in Cluster 4 either have only a small number of UEs or the UEs are not highly utilized.

In addition, mobile network operators and users are concerned with QoS which can be evaluated using throughput. Since Dataset 1 does not have features related to throughput, the following experiment uses Dataset 2 which has several metrics related to throughput. To understand the impact of RRC Connections/Cell on throughput, we select two throughput related features from Dataset 2 and
analyze the relationship between these features. These two features are *Downlink UE Throughput* (Feature 11) and *Uplink UE Throughput* (Feature 12). To analyze the relationship of these features with respect to the *RRC Connections/Cell* (Feature 4 in Dataset 2) we proceed as follow. There are 60129 records in total for different base stations at different times. We group every 500 records and calculate the average value for each of these three features and then plot the result as depicted in Figure 4 (for *RRC Connections/Cell* and *UE throughput* results).

To analyze the relationship between these features, we proceed as follows. There are 60,129 records in total for different base stations at different times. We group every 500 records and calculate the average value for each of these three features and then plot the result as depicted in Figure 4 (for *RRC Connections/Cell* and *UE throughput* results).

Figure 4 has two vertical axes, the left axis is for *RRC Connections/Cell* and the right axis is for *Downlink and Uplink Throughputs*. Figure 4 shows that as the number of *RRC Connections* increases, both the downlink and uplink data radio bearers (DRB) UE throughputs decrease. The uplink DRB UE throughput drops quickly when the number of average connected users starts to increase and the downlink DRB UE throughput decreases more evenly and slowly. This figure demonstrates that a large value of the number of average connected users can lead to lower resource utilization (both downlink and uplink) and thus QoS can be affected. For example, Netflix, the popular video streaming service recommends at least 5 Mbps for downlink throughput for streaming HD (High Definition) quality video (Netflix, 2015). From this analysis, we observe that average connected users/cell has a strong negative correlation with UE throughput. Therefore, a large number of average connected users/cell may degrade QoS because of the lower UE throughput. Conversely, a small number of average connected users/cell may not be desirable because it can lead to lower resource utilization if the service provider has a larger number of base stations.

The results of the experiment can help the network operator effectively allocate and manage resources and improve QoS for users.

**Analysis of Impact of RRC Connections for a weekday.** To compare the distribution of base station cells between a weekend (Sunday, Jan. 25, 2015) and a weekday, an analysis of a weekday (Monday, Jan. 26, 2015) is performed. The K-Means clustering technique is applied to this data and the number of clusters (K value) is chosen to be 4 since it best represents this dataset as described in Section 4.3. Four clusters are generated and the number of base station cells assigned to each cluster is shown in Table 8. For example, the first cluster includes 176 base station cells.

Table 8. The number of cells for each cluster

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cells</td>
<td>176</td>
<td>853</td>
<td>2027</td>
<td>2802</td>
</tr>
</tbody>
</table>

Figure 5 shows the value of average RRC Connections at different time slots of a weekday (Monday, Jan. 26, 2015). By comparing the results of Figure 3 and Figure 5, it is observed that the overall trend is similar. However, cluster 1 of the weekday result (cf. Figure 5) has a slightly lower average value compared with cluster 1 of the weekend result (cf. Figure 3). In Figure 5, cluster 1 reaches the minimum of 32 at 5:00 and achieves the maximum of 150 at 14:00 compared to Figure 3 where cluster 1 achieves a minimum value of 45.93 at 5:00 and its peak value of 194 at 15:00. This is because cluster 1
of the weekday result has more cells and some of these cells have some lower values of number of Avg. RRC Connections/Cell compared with cluster 1 of the weekend result. Data of other weekdays are analyzed and the results follow the same trend as that of Monday as shown in Figure 5. Note that those cells assigned to the first cluster in results of Monday (Jan 26, 2015) and Tuesday (Jan 27, 2015) have 93 in common but only 3 cells are common on Sunday (Jan 25, 2015) and Monday (Jan 26, 2015).

4.6 PERFORMANCE EVALUATION OF HADOOP

This section evaluates the performance of executing the K-means clustering technique on a Hadoop cluster with a larger amount of data. Note that the experiments are not meant to achieve the optimal performance. Rather, this is a set of preliminary experiments to demonstrate the effect of various parameters and the potential benefits of using Hadoop for performance analysis and potential improvement.

The default values for the Hadoop cluster parameters used for all experiments in this section are as follows:

- Clustering method: K-means;
- K = 4;
- ConvergeThreshold = 0.01;
- DistanceMeasurement=EuclideanDistance; and
- MaximumIterations = 150.

The experimental results using Hadoop provide useful performance guidelines for future data analysis. The information becomes crucial if real-time traffic data analysis needs to be performed, as the execution speed is important for real-time data analysis. The following highlights some of our experimental results.

Evaluation of Parallel Execution. Experiments in this section are executed on the Hadoop cluster composed of the Master Node (VM1) and the Slave Node D (VM5) with five experiment data sets. The first experimental data is Dataset 1, which comprises data for one week and is 236 MB. The other four data sets contain two to five weeks of data with sizes ranging from 472 MB to 1180 MB (with 236 MB increment for each week). Besides changing the size of the experiment dataset, the numbers of Map and Reduce tasks running in parallel are also varied to investigate how the Hadoop cluster performance is affected.

Figure 6 illustrates how changing the number of map tasks affects the execution time of the Map Stage. Overall, the performance improves when the number of Map tasks increases from 1 up to 8, but deteriorates at larger values (i.e., 16 and 32 Map tasks). This means that when the data size is small (236 MB), using a large number of Map tasks (e.g., 8, 16 or 32) for parallel processing does not generate a better performance due to the overhead associated with creating Map tasks and managing the splits. It is safe to conclude that for the example datasets the optimum number of parallel processing is 8 (i.e. 8 Map tasks). Using the 8 Map tasks as a baseline, as the size of the input dataset increases, there is improvement in the performance because of increased parallelism. The reduction can be significant for a larger data size, e.g., the execution time is reduced from 120+ seconds to ~30 seconds for data size 5 of 1180MB, as shown in Figure 6.

![Figure 6. Effect on the execution time of the Map Stage when varying the number of Map tasks.](image)

Figure 7 shows the impact of the number of Reduce tasks on system performance when using Dataset 5: 1180MB. The number of Reduce tasks is changed from 1 to 32 and the number of Map tasks is fixed at 32, since the number of Map tasks should not be less than the number of Reduce tasks (Leskovec, J., 2014). Note that the results of the experiments using the smaller datasets show a similar trend in performance. As shown in Figure 7, increasing the...
number of Reduce tasks decreases system performance. This is because the time taken for the Reduce stage keeps increasing, which in turn also increases the turnaround time. The Reduce stage is composed of shuffling, sorting and Reduce processing. The execution time of the shuffling and sorting operations are observed to increase with the number of reduce tasks. Thus, for the dataset experimented with, it is observed that using one Reduce task achieves the best performance and increasing the number of reduce tasks is not effective.

Figure 7: Effect on system performance when varying the number of reduce tasks.

Comparison of Performance for Different Number of CPUs. This section compares the performance of a Hadoop job executing on a slave node with different numbers of cores. The Hadoop cluster used is composed of the Master Node (VM1) and Slave Nodes A, B, C, and D with 1, 2, 4, and 8 CPUs, respectively. The size of dataset used for this experiment is 1180 MB. The number of Map tasks is chosen to be 8 and the number of Reduce tasks is set to 1 since they are the optimal values based on the result as shown in Figure 6 and Figure 7.

The execution time of the Map stage, the Reduce stage, and the turnaround time are depicted in Figure 8. Increasing the number of CPUs decreases the execution time for the Map stage and slightly for the Reduce stage due to the increased parallel execution using more cores. Note that the turnaround time is typically more than the sum of the average execution time required for the Map stage and Reduce stage because of the execution time required by the other phases of the MapReduce jobs, including the shuffle and sort phases. In the one case where the opposite is true (i.e. when the number of CPUs is 8), this can be attributed to Hadoop executing some of the Reduce tasks before all the map tasks finish executing. The total execution time can be reduced from ~188 seconds to ~37 seconds from 1 CPU to 8 CPUs for our data set with 1180MB.

Figure 8. Effect of varying the number of CPUs.

5. CONCLUSIONS AND FUTURE WORK

This paper focused on processing and analyzing mobile network traffic data. One of the main objectives was to identify important information from traffic data patterns that may provide useful insight for the network operator. We presented a detailed approach to processing and analyzing the datasets collected from real life mobile networks. The approach used the clustering algorithms (K-means and Fuzzy K-means) and other data mining algorithms of the Mahout Machine learning library to analyze the datasets. Furthermore, this research also investigated how the performance of the Hadoop cluster is affected when executing the Mahout clustering algorithms using different number of Map and Reduce tasks, and workload parameters. A summary of our experiences and key findings from analyzing the datasets are outlined below.

- The analysis of the Average RRC (radio resource control) Connections/Cell (a critical feature for mobile networks) over a 24h period, showed that a majority of base stations had an acceptable number of connected users. However, 71 base stations have a high value of Average
RRC Connections/Cell. The utilization and performance of those 71 base stations may be affected by the high demands, and hence they need to be monitored closely for better resource management and future network planning.

- The results showed that applying PCA before clustering could reduce the execution time (e.g., for Dataset 1 the execution time decreased from 1637s to 1046s) while still maintaining the accuracy of the clustering solution (99.45%). The study is useful for large datasets and for real-time traffic data analysis where execution time becomes more important.

The advantage of using PCA is obvious; however, in some cases it is not practical to apply dimension reduction because the information of the original features needs to be analyzed. For example, when we analyzed how the Average RRC Connections/Cell changes in a 24-hour period, the original information is needed as opposed to transposed data generated by PCA transformation.

With regards to the performance evaluation of the Hadoop cluster when executing K-means clustering, it was observed that increasing the number of Map tasks could noticeably reduce the execution (i.e., 8 Map tasks for our datasets). However, generating many Map tasks for some data size can lead to poor performance due to the overhead. Furthermore, as expected, increasing the number of processing cores improves performance, which shows that Hadoop and Mahout are scalable. Based on our preliminary experimentation, the performance can even be significantly improved for a reasonably large data size, i.e., 1180MB, using multiple Maps and/or CPUs. A great deal of research using Hadoop for performance has been reported in the literature. The approach, together with a reduced number of features using PCA, can considerably shorten the total execution time. As a result, it can facilitate real-time big data analytics, which can provide useful information in a timely manner for the network operator to effectively manage resources.

Using other machine learning algorithms, such as classification algorithms in the Mahout library to analyze the datasets, is an interesting direction for future research. In addition, developing a tool that can automate the data analysis approach presented in this paper can be investigated. The tool can automate the process of collecting, processing, analyzing, and displaying the analysis results.

6. ACKNOWLEDGMENT

The authors of this paper would like to thank Ericsson, Canada for providing various contributions and technical support. In addition, this research is partly sponsored by NSERC Engage and NSERC Discovery grants.

7. REFERENCES


Authors

Man Si is currently a software developer at CENX Inc. located in Ottawa, ON, Canada. In 2015, she received her M.A.Sc. in Electrical and Computer Engineering from Carleton University (Ottawa, ON, Canada), and in 2013, she graduated from Dalian Maritime University with a B.Eng. degree in Electronic and Information Engineering. Her research interests include data analysis and distributed systems.

Chung-Horng Lung received the B.S. degree in Computer Science and Engineering from Chung-Yuan Christian University, Taiwan and the M.S. and Ph.D. degrees in Computer Science and Engineering from Arizona State University. He was with Nortel Networks from 1995 to 2001. In September 2001, he joined the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, where he is now a Professor. His research interests include: Software Engineering, Cloud Computing, and Communication Networks.

Samuel Ajila is currently an associate professor of engineering at the Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada. He received B.Sc. (Hons.) degree in Computer Science from University of Ibadan, Ibadan, Nigeria and Ph.D. degree in Computer Engineering specializing in Software Engineering and Knowledge-based Systems from LORIA, Université Henri Poincaré – Nancy I, Nancy, France. His research interests are in the fields of Software Engineering, Cloud Computing, and Big Data Analytics and Technology Management.

Wayne Ding is a senior specialist in Ericsson, Canada. He is involved in 4G and 5G wireless system R&D. His work focuses on mobile big data and wireless network KPIs. He received the BS and MS degrees Dalhousie University, Canada.