Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

MSC THESIS

Analysis and Prediction of Mobile Application Usage Based on location In case of ethiotelecom

Bancheamlak G/Tsadik

Advisor
Dr. Surafel Lemma

February 21, 2020

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Thesis proposal on

Analysis and Prediction of Mobile Application Usage Based on location In case of ethiotelecom

By: **Bancheamlak G/Tsadik**

Signed by :

Adviser	Dr. Surafel Lemma	Signature	_____	Date	_____
Evaluator	_____	Signature	_____	Date	_____
Evaluator	_____	Signature	_____	Date	_____

Abstract

The explosive growth of smart devices, network access points, and new mobile application development drives users to use more and more mobile applications and, this has lead to the explosive growth of mobile data traffic. It has a high impact on mobile service providers to manage network data traffic because application usage is different from one location to other with time. Understanding the application-level traffic patterns from a completely different location angle is effective for operators and content providers to create technical and business plans.

In this paper, we have established several typical traffic patterns and predict application category traffic demand per clustered location in a mobile cellular network. We explore mobile traffic patterns by clustering each application category into five clusters based on traffic volume and location. Then, we implement a random forest model to predict the traffic demand of three of the most highly utilized applications per cluster location.This outcome could be useful in relevant future applications, with the prospect to achieve average 96% predictive accuracy per application category per cluster.

Understanding popular application at the clustered locations and predicting the traffic demand of a popular application could significantly improve user experience, average latency, energy consumption, spectral efficiency, back-haul traffic, and network capacity. Those outcomes are possible via designing and implementing a cache server or planning and optimizing the network resources based on predicted traffic demand.

Keywords

Keywords: Mobile application usage, Traffic demand, Popular application.

Acknowledgment

Above all, thanks to the almighty God for his help in giving me the courage to cope up with complicated situations I have faced for pursuing the study and for his help during my whole study time.

I would like to express my gratitude to my advisor, Dr. Surafel Lemma, whose continual supervision and direction made this paper possible. Thank you for pushing me to engage critically and thoughtfully during this iterative exercise.

I would also like to thank my special friend Trufre Abera and Fikru Feleke for all the assists all are provided throughout this work. Thanks, Friends!

Contents

Acronyms **vii**

Abstract **viii**

1 Introduction **1**
1.1 Statement of the Problem 2
1.2 Objective 3
1.3 Scope of the Work 3
1.4 Methodology 3
1.5 Contribution of the Thesis 4
1.6 Thesis Organizations 4

2 Background **6**
2.1 UMTS Network 6
2.1.1 UMTS Network Architecture 7
2.1.2 User Equipment (UE) 8
2.1.3 UMTS Radio Access Network (UTRAN) 8
2.1.4 UMTS Core Network (CN) 8
2.2 UMTS Functions 10
2.3 Machine Learning Algorithms 11
2.3.1 ML for Time Series Forecasting 12
2.3.2 Multi Variable Regression(MVR) 13
2.3.3 Artificial Neural Networks (ANNs) 14
2.3.4 Random Forests (RF) 16

3 Related Work **20**
3.1 Source of Available Mobile APP Data 20
3.2 Mobile APP Usage Characteristics 20

3.3 Mobile APP Traffic Usage Prediction 21

4 Experimental Design **24**
4.1 Data Collection and Analysis 25
4.1.1 Data Collection 25
4.1.2 Analysis 26
4.2 Location Clustering 27
4.2.1 APPs Usage Distributions Per Cluster 28
4.3 Popular APP Usage Traffic Prediction 28
4.3.1 Model Selection 30
4.3.2 Finding Best Parameters 30
4.3.3 Validation Curve With k-fold Cross-Validation 31
4.3.4 K-Fold Cross-Validation With Grid Search 33
4.4 Building RF Model 34
4.5 Evaluation Metric 34

5 Results and Discussion **36**
5.1 Analysis Results 36
5.2 Cluster Analysis Result 38
5.3 Parameter Tuning 40
5.4 Popular APP Prediction Result 44

6 Conclusion and Recommendation **49**
6.1 Conclusion 49
6.2 Recommendation 50

List of Figures

1.1 Overall experimental process 4

2.1 UMTS network architecture 7
2.2 The three-layer feed-forward ANN architecture 15
2.3 Random forest Regression Tree[1] 17

4.1 Experimental process Block diagram 24
4.2 Mobile network architecture[2] 25
4.3 APP Distribution per each cluster 29
4.4 Basic diagram of 10-fold cross validation 32

5.1 APP traffic distribution in one week 36
5.2 Streaming, File Access and Web_Browsing traffic distribution in one week 37
5.3 Selected sites Total Traffic Distribution 38
5.4 Cells cluster based on RNC 39
5.5 Traffic Distribution pattern per cluster 40
5.6 n_estimator validation curve 41
5.7 Error Vs n_estimators Curve 41
5.8 max_depth validation curve 42
5.9 min_samples_leaf validation curve 42
5.10 min_samples_split validation curv 43
5.11 APPs category Traffic demand per cluster 45
5.12 Prediction out put of APPs for the first category 46
5.13 Prediction out put of Web browsing by MVR Model 46
5.14 Prediction out put of Web browsing by RF Model 47

List of Tables

2.1 UMTS interface 10

4.1 APPs categories List 26
4.2 Selected six APPs categories description 27

5.1 Traffic Distribution per cluster per APP category 40
5.2 Parameter Tuning result 43
5.3 Percentage Traffic distribution per cluster per category 44
5.4 Prediction Evaluation Result 47

Acronyms

2G second-generation.

3GPP 3rd generation Partnership Project.

ANNs Artificial neural networks.

AUC Authentication Center.

CDMA Code Division Multiple Access.

CN Core Network.

CS circuit switch.

EDGE Enhanced Data rates for GSM Evolution.

EIR Equipment Identity Register.

GGSN Gateway GPRS Support Node.

GPRS General Packet Radio Service.

GSM Global System for Mobile.

HLR Home location Register.

IMSI International Mobile Subscriber Identity number.

IMT-2000 International Mobile Telecommunications - 2000.

IT Information technologies.

ITU International Telecommunication Union.

ML Machine Learning.

MR Multi variable Regression.

MSC Mobile switching center.

MSISDN Mobile Station International ISDN Number.

MVR Multi variable Regression.

PS Packet-switched.

RF Random forests.

RMSE Root Mean Square Error.

SGSN Serving GPRS Support Node.

UMTS Universal Mobile Telecommunications System.

USIM Universal Subscriber Identity Module.

UTRAN UMTS Radio Access Network.

WCDMA Wideband Code Division Multiple Access.

XDR Traffic Data Record.

Chapter 1

Introduction

Mobile internet grows rapidly because of the increment of new mobile Application(APP) development, smart phone adaption and mobile internet access points. With the increase in the number of mobile APPs and a remarkable increase in the network capacity of a mobile phone, people use more and more mobile APPs to receive news and updates through mobile internet nearly anywhere at any time. Much of this network activity is done either through web services or a mobile APP downloaded on the mobile device. In the past few specified years, the gradual increase of smart devices such as modern smart phones lead to an explosive growth of mobile data traffic.

According to Cisco forecasting , global data traffic demand grows 71% over a year in 2018[3]. Most mobile APPs have streaming video contents that consume a lot of valuable resources. For example, users using web-based APPs like You Tube and Face book, requires a broadband link to access multimedia-rich content. With limited available cellular resources, the increasing traffic demand is another burden for a network provider. To handle this problem it needs a proper understanding of APP usage, efficient resource allocation and, properly planning and optimizing of the mobile network resource.

Characterizing APP usage from a traffic demand perspective is necessary for both mobile network operators and content providers in order to design, manage and optimize cellular networks[4][5]. If traffic demand for APP usage is identified and predicted, operators get more information about the user's traffic demand. The information helps to provide good quality of service (QoS) for customers.Content providers can target the potential users and make good marketing strategies. In addition to that, it helps the network provider to reduce the call dropping probability and congestion and improve re-

source utilization by planning the network resources based on the predict traffic demand.

In this paper, we propose a random forest regression technique to predict the mobile application usage at a given clustered location by using the Traffic Data Record (XDR) of APP.It has a good accuracy, strength and, ease of use. By analyzing a large scale application usage dataset over 6,968 BSs, which is collected from the mobile network of Addis Ababa over a period of one week, we investigate the challenges and opportunities of location with APPs usage and estimate the APP usage in each clustered area of the city.

1.1 Statement of the Problem

With the increasing popularity of mobile internet access, an exhaustive understanding of user behavior becomes important for Internet service providers to perform reasonable network management, network capacity planning, and resource allocation. If operators do not reasonably understand their customer APP usage behavior, their network planning especially capacity planning will not satisfy the required demand. In some proper places, mobile users have experienced difficulties in getting the desired QoS of the mobile network. Most of the time, the traffic to the sites is time and location-dependent [6]. The traffic is nonuniform all over the covered areas because of the high APP usage and movement of the users. Due to this reason, there is a problem in using the network resource efficiently. As a result, studying user behavior in terms of their APP usage behavior is necessary to resolve the problem.

The mobile network operators generate and hold big data which is a valuable resource in getting users and network characteristics. However, there is a gap in proper mining and analysis of data to get deeper insights into customer APP usage behavior and proper utilization of the resource. In ethio telecom, so far there is no attempt made to study mobile APPs usage behavior for planning and resource allocation in the mobile network. Ethio telecom resource allocation is flat which makes resource utilization inefficient. This thesis will predict mobile APP service usage traffic demand based on location and time period for the case of Addis Ababa mobile Internet users.

1.2 Objective

The objective of the thesis is to predict APP usage based on location in Addis Ababa mobile internet users.

The following lists what we plan to do in order to achieve the objective.

- To categorize the APP by function and traffic similarity type.
- To analyze Existing Cluster BSs based on RNC and traffic demand.
- To analyze APP usage demand based on location, time and day.
- To predict the top APP categories service in each cluster BSs based on the traffic demand.
- Identify the traffic distribution of different APPs in different location.

1.3 Scope of the Work

We will predict the APP service usage of Addis Ababa's 3G mobile internet data service based on location and time for proper resource utilization, allocation and, planning. In addition to that, recommend cache severer implementation to over came capacity problem.

1.4 Methodology

The research work has five phases. In the initial part, APPs data is collected from 6968 cells placed in Addis Ababa from ethiotelecom smart care network information system. The data is preprocessed to fit the experiment design.

In the second part, we analyze the traffic data from the perspective of time of the day, day of the week and location to have sense of the data property.

In the third part of the experiment, we cluster the traffic into different locations with BSs having similar properties grouped to the same location.

Predicting APP usage per every cluster and choosing the popular APP usage category based on the traffic demand is within the fourth part of the experiment.

Finally, the performance of the prediction techniques is evaluated by using r^2 and Root Mean Square Error (RMSE) metric. For analysis and prediction we use python data science libraries such as a k-means algorithm for clustering and Random Forest algorithm for prediction. The methodology is illustrated in block diagram shown in Figure1.1.

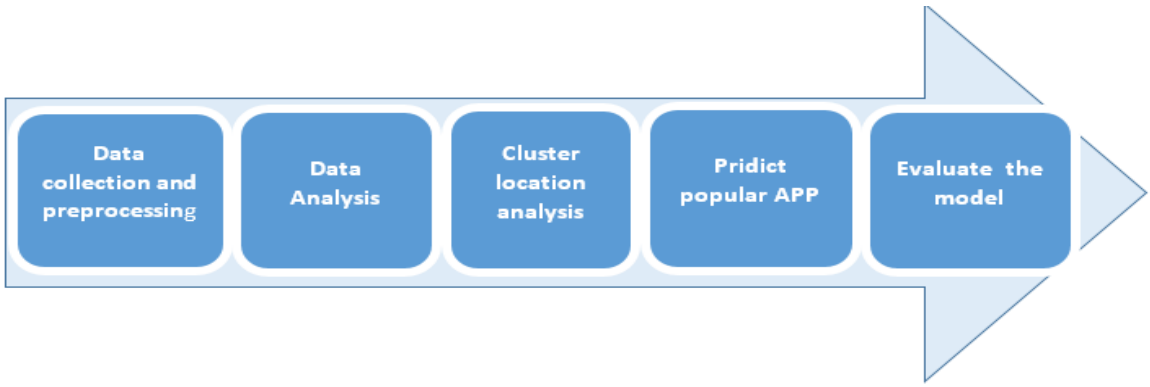

Figure 1.1: Overall experimental process

1.5 Contribution of the Thesis

This work can commit to predict mobile APP usage traffic and identify popular APPs categories per clustered location. The prediction result helps network providers to proactively allocate resources based on the user traffic demand. Moreover it controls congestion and call drop that may occur by the APP usage traffic consumption.The prediction output result gives information to operators for planning, optimizing their network and utilizing the resource properly .

Understanding the popular APP distribution information is important to design a cache server and implementation at base station location [7].

1.6 Thesis Organizations

The rest of the paper is organized as follows, in chapter 2, the subject matter of the research, the UMTS network and machine learning techniques have been discussed. Chap-

ter 3, types of research on APP usage characteristics and prediction is reviewed. Chapter 4 details the experimental analysis used in this research task will be discussed under each sub-module of the system model. In Chapter 5, the results and discussion on the experiment simulation result with selected experimental areas will be discussed. Finally, we conclude the finding of the paper in Chapter 6.

Chapter 2

Background

In this chapter, we disclose the required information on APP usage prediction is introduced, which has the basic cellular network structure and Machine Learning Algorithms. The information has been necessary for mobile APP traffic demand prediction, network management, and resource allocation. In the first section, we are defining the UMTS network and architectures that have presented. Within the second section, Machine Learning algorithmic rules, definitions, and it is working principles are described.

2.1 UMTS Network

Universal Mobile Telecommunications System (UMTS) network is the convergence of mobile communications Information technologies (IT) and multimedia technologies[8]. UMTS creates new opportunities for network operators service providers and content providers to generate revenue and hold market share. It is a suite of radio and network technologies that provide better spectrum efficiency, high data transmission rates (up to 2 Mbit/s), the capability to support new multimedia APPs and interoperability with both fixed and mobile telecommunications networks [9].

UMTS is the natural evolution from Global System for Mobile (GSM) and other second-generation (2G) mobile systems. It provides interconnection with 2G networks as well as other global and satellite-based networks.UMTS presents a unique opportunity to cater to the needs of individuals in the Information Society[8]. As a multi-national, multi-sector system that supports numerous protocols and transport technologies, UMTS eliminates complications that one posed problems for communications and enables the creation and delivery of fully adapted communication services to both network providers and users.

UMTS is a International Mobile Telecommunications - 2000 (IMT-2000) 3G system and it is the 3rd generation Partnership Project (3GPP) developing technical specifications or IMT-2000 and the International Telecommunication Union (ITU) framework for 3G standards. The other main IMT–2000 system proposed by the ITU has Code Division Multiple Access (CDMA). Operators with existing Interim Standard 95 networks will migrate to CDMA 2000. CDMA 2000 will be deployed in North America and Asia. DMA 2000 has been a narrow band system whereas, UMTS usesWideband Code Division Multiple Access (WCDMA) technology. The first available release of CDMA 2000 does not provide transmission speeds recommended by the IMT-2000. However, the first UMTS release (3GPP Release 99) is on time, and guarantees recommended speeds. CDMA 2000 will eventually deliver full IMT-2000 requirements[9].

2.1.1 UMTS Network Architecture

The UMTS 3G architecture has been required to provide a greater level of performance to that of the novel GSM network. However, as many networks had transferred through the use of General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE), they already could carry data. Therefore, many of the elements required for the WCDMA / UMTS network architecture were seen as the transfer. With one of the major aims of UMTS presence to be able to carry data. The UMTS network architecture has been designed to enable a considerable improvement in data performance over that provided one [2].

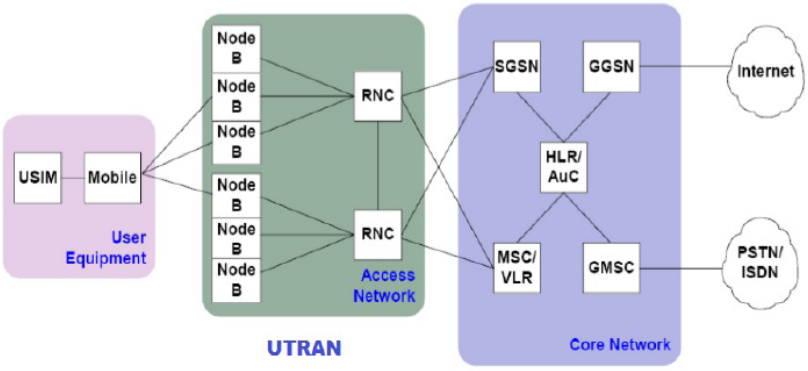

Figure 2.1: UMTS network architecture

2.1.2 User Equipment (UE)

UE represents a critical element of the overall 3G UMTS network architecture. It forms the final interface with the active user. Several elements within the UE are UE RF integrated circuit, Baseband processing, Battery, and Universal Subscriber Identity Module (USIM). UE RF area that handles all elements of the desired signal, both for the receiver and the local transmitter. One of the significant challenges for the UE RF power amplifier was to sufficiently reduce power consumption, to properly maintain excellent battery life, and measures were properly introduced into many of the modern designs to ensure optimum efficiency. Baseband signal processing contains primarily of digital circuitry. This is significantly more complex than that used in phones for previous generations. Again this has been optimized to reduce the current consumption as far as possible.

USIM is the UE also containing a SIM card although, in the case of UMTS, it is termed a USIM. This has a more version of the SIM card used in GSM, and other systems but embodies the identical types of information it typically contains the considerable International Mobile Subscriber Identity number (IMSI) as well as the Mobile Station International ISDN Number (MSISDN).

2.1.3 UTRAN

UTRAN or RNS comprises two main components RNC and Node B. The RNC undertakes the radio resource management and some of the mobility management functions, although not all. It is also the point at which the data encryption or decryption is performed to protect the user data from eavesdropping. Node B is the term used within UMTS to denote the base station transceiver. This part of the UTRAN contains the transmitter and receiver to communicate with the UEs within the cell. It participates with the RNC in resource management. Node B is the 3GPP term for the base station, and often the terms are used interchangeably.

2.1.4 UMTS CN

The 3G UMTS core network architecture is a drive that used for GSM with further elements covered to enable the additional functionality required by UMTS. Given the different ways in which data may be carried, the UMTS core network, may be split into two

different areas circuit switch (CS) fundamentals and, Packet-switched (PS) elements. CS elements are primarily based on the GSM network units, and carry data in a CS method, that is a permanent channel for the duration of the call [2]. To facilitate effective handover between Node Bs under the control of different RNCs. The RNC not only communicates with the Core Network but also, with neighboring RNC.

The CS elements of the UMTS core network architecture include the following network entities Mobile switching center (MSC) and Gateway MSC. MSC is essentially the same as that within GSM and, it manages the CS calls underway and gateway MSC this is effectively the interface to the external networks.

PS is network entity that id designed to carry packet data. This enables much higher network usage as the capacity can be shared and data is carried as packets that are routed according to their destination. The PS elements of the 3G UMTS core network architecture include the following network entities Serving GPRS Support Node (SGSN), Equipment Identity Register (EIR), Gateway GPRS Support Node (GGSN), Home location Register (HLR),EIR andAuthentication Center (AUC).

The SGSN provides a number of functions within the UMTS network architecture Mobility management, session management, interaction with other areas of the network and Billing. Like the SGSN, this entity was also first introduced into the GPRS network. GGSN is the central element within the UMTS PS network. It handles inter-working between the UMTS PS network and external PS networks. HLR database contains all the administrative information about each subscriber along with their last known location. The EIR is the entity that decides whether a given UE equipment may be allowed onto the network. The AuC is a protected database that contains the secret key also contained in the users USIM card.

Four interfaces are connecting the UTRAN internally or externally to other functional entities: Iu, Uu, Iub and, Iur. The Iu interface is an external interface that connects the RNC to the CN. The Uu is also external, connecting Node B with the UE. The Iub is an internal interface connecting the RNC with Node B. And at last, there is the Iur interface which is internal most of the time but can, exceptionally be an external interface too for some network architectures. The Iur connects two RNCs as a summary all interface between UMTS components are shown in Table 2.1

Table 2.1: UMTS interface

Interface	Description
Uu	interface between UE and Node B
Iub	interface between Node B and RNC
Iur	interface between RNC and RNC
Iu-CS	interface between RNC and MSC
Iu-PS	interface between RNC and SGSN

2.2 UMTS Functions

Network architecture shows what functions a network must provide, e.g, data forwarding, mobility support, and how these functions are grouped in the network. It creates functional groups that should have connected through protocols that to designed later[8]. The following are some of the functions in the mobile network which can be grouped in UMTS architecture.

- Transport – is a network functionality which enables movement of information from one network element to the other.
- Routing – this function deals with selecting path for the transportation of traffic between different elements in the cellular network.
- Security – solves safety issues from both the network and users perspective. From the network side, it checks whether a user has the right to access or not and from the user side it filters trustworthy connections. Additionally security functions keep protected the users' privacy, such as identity and location.
- Session control or call control – are responsible for monitoring the communication between two users or devices after the call is set up or session is established. Most of the time word session is used in packet-switched communications whereas call is in circuit switched.
- Quality of Service – These functions ensure delivery of a service in the range of predefined agreement levels or network standards.
- Radio resource control and provisioning – mainly acts in the communication between the user equipment and Node B. It is for setting up, modifying and releasing resources in different layer of the radio interface protocol stack.
- Mobility – this function enables the system to maintain a session or call while the user is moving across the network by making new connections.

- Charging – this is a billing function which controls the users' payment for the service they got from the network.

2.3 Machine Learning Algorithms

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being openly programmed. In essence, the goal of Machine Learning (ML) is to identify and exploit hidden patterns in training data. The patterns are determine and used to analyze unknown data, such that it can be grouped or mapped to the known groups[10]. This brings about a shift in the traditional programming model, where programs are written to automate tasks and ML creates the program (model) that adjusts the data. Currently, ML was enjoying renewed interest. Early ML techniques were rigid and unable to tolerate any variations from the training data.

Recent advances in ML have made these techniques flexible and strong in their applicability to various real-world scenarios, ranging from unusual to ordinary. For instance, ML in telecom companies can instantaneously analyze through millions of Traffic Data Record (XDR) in real-time, identify patterns, create scalable data visualizations, and predict future problems. Ordinarily, we frequently employ technological tools founded upon ML. For example, search engines extensively use ML for non-trivial tasks, such as query suggestions, web indexing, and page ranking. As we look forward to automating more aspects of our lives, ranging from home automation to autonomous vehicles.ML techniques will become a gradually important facet in various systems that aid in decision making, analysis, and automation.

Most importantly, the success of ML techniques depends unquestionably on data [11]. A massive amount of data in today's' networks are bound to grow further with emerging to data traffic. Networks, such as the Internet of Things (IoT) and it is billions of connected devices. This inspires the APP of ML that not only identifies hidden and unexpected patterns but can equally be applied to determine and accept the processes that generate the data. Recent advances in computing offer storage and processing capabilities for training and testing ML models for considerable data.

There are several machine learning algorithms mainly classified as supervised and

unsupervised algorithms. Supervised machine learning algorithms need a set of labeled examples as a training dataset to construct a classification model. Unlike the supervised ones, unsupervised algorithms don't need labeled examples and usually used in clustering tasks instead of classification.ML models based on time series forecasting are much difficult to implement compared to the supervised and unsupervised learning models because of the temporal difference in the data.

2.3.1 ML for Time Series Forecasting

Time series molding has an active research area that is attracted to the attention of researchers over the last decades. The main aim of time series modeling is to carefully collect and thoroughly study the past comments of a time series to develop an appropriate model that describes the structure of the data. This model is used to generate future values for the series, which is to make forecasts. Time series forecasting thus can be termed as the act of predicting the future by understanding the past data pattern[12]. Due to the essential importance of time series forecasting in numerous practical fields such as business, economics, finance, science, and engineering, etc[13]. Proper care should be taken to fit a passable model to the underlying time series. A successful time series forecasting depends on an appropriate model fitting. A lot of efforts have been done by researchers over many years for the development of efficient models to improve forecasting accuracy.

In time series forecasting, past observations are collected and analyzed to develop a suitable mathematical model that captures underlying data generating a process for the series [7, 8]. Future events have been predicted using this model. This approach has been particularly useful when there is not much knowledge about the numerical pattern and when an acceptable helpful model is absent. Time series forecasting has important applications in various fields. Often valued strategic decisions and preventive measures are taken based on the forecast results and available data characteristics. Thus making a good forecast, that is fitting a passable model to a time series choosing. Over the past, many efforts have been made by researchers for the development and improvement of appropriate time series forecasting models. Various important time series forecasting models have been evolved in collected works but the most popular and frequently used time series models are Multi variable Regression (MR)),Artificial neural networks (ANNs) and Random forests (RF).

2.3.2 Multi Variable Regression(MVR)

A Multi variable Regression (MVR) model has often thought of as a model during which multiple variables square measure found on the proper aspect of the model equation. This sort of applied mathematics model has often familiarized requires assessing the connection between the variety of variables. One will measure independent relationships where as adjusting for potential confounded.

Multiple linear regression model Mathematical form is:

$$Y = b_0 + b_1X_1 + b_2X_2 + ...b_PX_P$$

Where from the equation indicates the predicted or expected value of the dependent variable, X_1 through X_p are Y distinct independent or predictor variables, b_0 is the value of Y when all of the independent variables (X_1 through X_P) are equal to zero, and b1 through b_p is the estimated regression coefficients. Each regression coefficient represents the change in Y relative to a one-unit change in the respective independent variable. In the multiple regression situation, b_1, for example, has been the change in Y relative to a one-unit change in X_1, holding all other independent variables constant (i.e., when the remaining independent variables are held at the same value or are fixed). Again, statistical tests can be performed to assess whether each regression coefficient is significantly different from zero.

Multiple regression analysis can be used to assess whether trying exists, and, since it allows us to estimate the association between a given independent variable and the outcome holding all other variables constant, multiple linear regression has also provided a way of adjusting for (or accounting for) potentially confounding variables that have been included in the model.

Suppose we have a risk factor or an exposure variable, which we denote X_1, and an outcome or dependent variable which we denote Y. We can estimate a simple linear regression equation relating the risk factor (the independent variable) to the dependent variable as follows:

$$Y = b_0 + b_1X_1$$

where b1 is the estimated regression coefficient that quantifies the association between

the risk factor and the outcome.

If we now want to assess whether a third variable is a confounder, we can denote the potential confounder X_2, and then estimate a multiple linear regression equation as follows:

$$Y = b_0 + b_1X_1 + b_2X_2$$

In the multiple linear regression equation, b_1 is the estimated regression coefficient that quantifies the association between the risk factor X_1 and the outcome, adjusted for X_2 (b_2 is the estimated regression coefficient that quantifies the association between the potential confounder and the outcome). As noted earlier, some investigators assess confounding by assessing how much the regression coefficient associated with the risk factor changes after adjusting for the potential confounder. In this case, we compare b_1 from the simple linear regression model to b_1 from the multiple linear regression model. As a rule of thumb, if the regression coefficient from the simple linear regression model changes by more than 10 percent, then X_2 is said to be a confounder.

Once a variable is identified as a confounder, we can then use multiple linear regression analysis to estimate the association between the risk factor and the outcome adjusting for that confounder. The test of significance of the regression coefficient associated with the risk factor can be used to assess whether the association between the risk factor is statistically significant after accounting for one or more confounding variables.

2.3.3 Artificial Neural Networks (ANNs)

ANNs, the approach has been suggested as an alternative technique to time series forecasting and it gained immense popularity in the last few years. The basic objective of ANNs was to construct a model for mimicking the intelligence of the human brain into a machine [14]. Similar to the work of a human brain, ANNs try to recognize regularities and patterns in the input data, learn from experience and then provide generalized results based on their known previous knowledge. Although the development of ANNs was mainly biologically motivated, afterward they have been applied in many different areas, especially for forecasting purposes[10]. Below we shall mention the salient features of ANNs, which make them quite favorite for time series analysis and forecasting.

First, ANNs are data-driven and self-adaptive in nature[10] . There is no need to

specify. A particular model form or to make any a priori assumption about the statistical distribution of the data, the desired model is adaptively formed based on the features presented from the data. This approach is quite useful for many practical situations, where no theoretical guidance is available for an appropriate data generation process. Second, ANNs are inherently non-linear, which makes them more practical and accurate in modeling complex data patterns, as opposed to various traditional linear approaches. There are many instances, which suggest that ANNs made quite better analysis and forecasting than various linear models. Finally, as discussed in [15], ANNs use parallel processing of the information from the data to approximate a large class of functions with a high degree of accuracy.

The most widely used ANNs in forecasting problems are multi-layer perception's, Which use a single hidden layer feed-forward network. The model is characterized by a network of three layers, viz. input, hidden and output layer, connected by acyclic links. There may be more than one hidden layer. The nodes in various layers are also known as processing elements. The three-layer feed-forward architecture of ANN models can be diagrammatically shown as below.

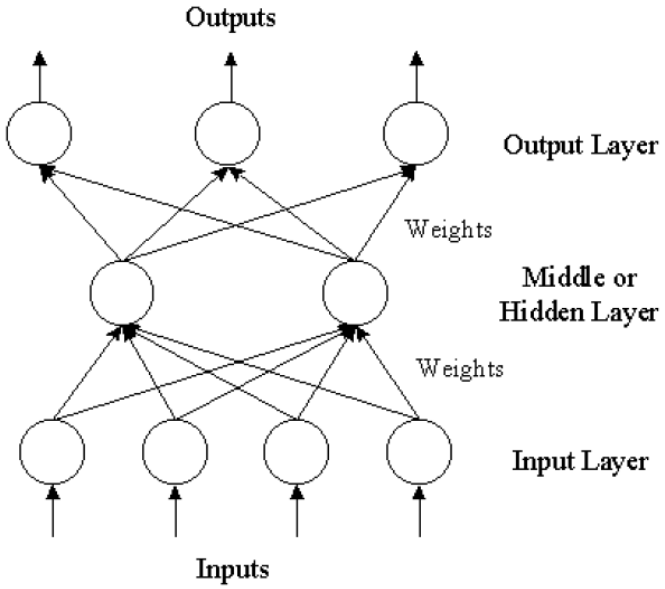

Figure 2.2: The three-layer feed-forward ANN architecture

The output of the model is computed using the following mathematical expression.

$$Y_t = a_j + \sum_{j=1}^{q} a_g(B_0 j + \sum_{j=1}^{a} B_i j Y_t - 1) + E$$

The feed-forward ANN model in fact performs a non-linear functional mapping from the past observations of the time series to the future value, i.e.

$$Y_t = (Y_{t-1}, Y_{t-2}, ...Y_{t_p}) + E$$

Where w is a vector of all parameters and f is a function determined by the network structure and connection weights. To estimate the connection weights, non-linear least-square procedures are used, which are based on the minimization of the error function [13]:

$$F(w) = \sum e_t^2 = \sum (y_t - y_t)$$

Here W is the space of all connection weights.

ANNs have some disadvantages: it has complex structure, it is not so powerful with linear data, and it takes more time when training the data[15]. The display mechanism to be determined here will directly influence the performance of the network. The network is reduced to a certain value of the error on the sample means that the training has been completed.

2.3.4 Random Forests (RF)

A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the tree response is an estimate of the dependent variable given the predictors[1]. The Random Forest algorithm was developed by Breiman [11].

A Random Forest consists of an arbitrary number of simple trees, which are used to determine the outcome. For classification problems, the ensemble of simple trees votes for the most popular class. In the regression problem, their responses are averaged to ob-

tain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The arrangement of trees as shown in Figure 2.3

The response of each tree depends on a set of predictor values chosen independently (with replacement) and with the same distribution for all trees in the forest, which is a subset of the predictor values of the original data set[16]. The optimal size of the subset of predictor variables is given by log2 M + 1 where M is the number of inputs.

For regression problems, Random Forests are formed by growing simple trees, each capable of producing a numerical response value. Here, too, the predictor set is randomly selected from the same distribution and for all trees. Given the above, the mean-square error for a Random Forest is given by mean error = (observed - tree response)

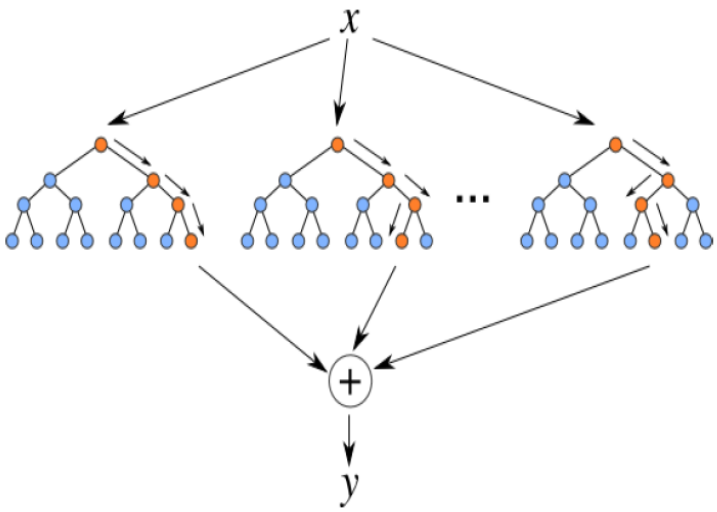

Figure 2.3: Random forest Regression Tree[1]

The predictions of the Random Forest is to be the average of the predictions of the trees. The random forest predictive value S is given as

$$S = \frac{1}{K} \sum_{k=1}^{K} K^{th}$$

Typically, Random Forests can flexibly incorporate missing data in the predictor variables [17]. When missing data is encountered for a particular observation (case) during model building, the prediction made for that case is based on the last preceding (non-terminal) node in the respective tree. So, for example, if at a particular point in the sequence of trees a predictor variable has been selected at the root (or other non-terminal) node for which some cases have no valid data, then the prediction for those cases is simply based on the overall mean at the root (or other non-terminal) node. Hence, there is no need to eliminate cases from the analysis if they have missing data for some of the predictors, nor is it necessary to compute surrogate split statistics.

The RF algorithm works as follows: for each tree in the forest, we select a bootstrap sample from S where $S^{(i)}$denotes the ith bootstrap. We then learn a decision-tree using a modified decision-tree learning algorithm. The algorithm is modified as follows: at each node of the tree, instead of examining all possible feature-splits, we randomly select some subset of the features $f \subseteq F$. where F is the set of features. The node then splits on the best feature in f rather than F. In practice f is much, much smaller than F. Deciding on which feature to split is oftentimes the most computationally expensive aspect of decision tree learning. By narrowing the set of features, we drastically speed up the learning of the tree[1].

Algorithm 1: Random Forest

A training set $S := (x_1,y_1),...,(x_n,y_n)$, features F, and number of trees in forest B

function Random Forest(S, F)

$H \leftarrow 0$ **foreach** $i \in 1,..., B$ **do**

$S^{(i)} \leftarrow$ A bootstrap sample from S

$h_i \leftarrow$ Randomized Tree Learn($S^{(i)}$,F)

H $\leftarrow H \cup \{h_i\}$

end

return H

end function

function Randomized Tree Learn(S, F)

at each node:

$f \leftarrow$ very small subset of F

Split on best feature tn f

return The learned tree

end function

Furthermore, by restricting the features that we consider at each node, we can learn each tree much faster, and therefore, can learn more decision trees in a given amount of time. Thus, not only can we build many more trees using the randomized tree learning algorithm, but these trees will also be less correlated. For these reasons, random forests tend to have excellent performance.

In this chapter, three different types of algorithm models are introduced, including the random forest. Some particular models in each type are also presented, such as MVR and ANNs, in the nonlinear regression.RF models are fast to train and tune. In our research, the time needed for RF model building was much less than for ANN (few minutes compared to hours) due to the structure of the RF algorithm characterized by few parameters to set and a limited number of variables to be permeated at each tree node. ANN needed more time for computer architecture design and learning as it performs a large number of trials moving the number of neurons and the type of activation function in the hidden layer. Due to the above benefit, we are using a random forest regression model for this work.

Chapter 3

Related Work

In this chester previous work done on APPs usage prediction area will be discussed. The discussion is organized in to three sub sections: the type of available mobile APP data, Mobile APP usage characteristics, and the Mobile APP traffic usage prediction in the mobile network.

3.1 Source of Available Mobile APP Data

In existing works, mobile APP data mainly have two Source according to the data collection method, data collected from mobile devices of individual[19],[20] and data collected from network operators [13],[6],[21]. A widely used dataset in the latter category is XDR.It is used in many usage pattern and social behavior study [22] . Data containing mobile traffic or APP usage provided by cellular network operators are also used in many studies to identify a pattern from the perspective of the user or operator [23]. In this study APP usage data is collected from the ethiothelecome dataset(XDR) of APP traffic usage of the mobile network in cell label.

3.2 Mobile APP Usage Characteristics

The usage of mobile phones and tablets is rapidly increasing and there is a constant battle for market share between companies. Mobile devices become even more attractive with the addition of personalized features such as mobile APP.

Using the mobile traffic data collected at core 2G and 3G networks over a week and using the normalized entropy distribution, Yang.Y et.al study user behavior from three aspects: data usage, mobility pattern and application usage [5] . They found that the big consumers of resource are the main driving factor behind the variation of data usage and mobility pattern, and they tend to consume massive data and radio resources at the same time. In addition, both users' data usage and mobility pattern heavily impact their application access behavior.

Understanding of mobile user behavior from the view of APP interests and location has been studied by many researchers [24],[25], [26]. Yan et.al [24] tried to analyze the behavior of the customer and calculate the feature by using improved Apriori algorithm then predict the app usage using Hidden Markov Models (HMM). Q.Chen et.el [25] use three data mining techniques: Recency, Frequency, Monetary (RFM) analysis, and association rule learning methods. The results show that big data analysis enables the mobile app developer to learn customer preferences, patterns of function usage and app users. Jiang et.el in [26] analyzed the user behavior from two perspectives, mobility and APP usage. The analyses are based on three variables: the occurrence patterns (using k mean), the mobility patterns (using Silhouette coefficient) and dominant locations (using cumulative distribution function). The researchers found that a user's mobility pattern is highly correlated with APP usage.

3.3 Mobile APP Traffic Usage Prediction

APP usage prediction refers to the task of predicting the next APP that will be used for a given user and at a given time. In this section, we review several essential aspects of current app usage prediction literature, including what types of features and what type of predictive modeling methodologies used to predict APP usage.

A number of studies have attempted to predict functions or APP use on smart phones by applying machine learning algorithms and contextual information. One study proposed a context-aware interface for cellular phone operations, which involves predicting APPs through a comprehensive analysis of the context related to mobile app use and build prediction models that calculate the probability of an APP in the current context. Based on these models, they developed a dynamic home screen application that presents

icons for the most probable apps on the main screen of the phone and highlights the most probable one[27]. This collects sensory data including APP use from smart phones and found that the dynamic home screen improved accessibility to APPs on the phone, compared to the conventional static home screen in terms of accuracy.

Some researchers attempted to predict functions or APP use on smart phones by point of interest [6],[28].This article aims to design an edge caching strategy for APP services based on the observed characteristics of BSs in terms of points of interest (POIs), logs, and traffic generated by various categories of APPs.The authors first analyze the temporal characteristics of different categories of apps, and then further investigate the logs and traffic generated by APP types under different BSs clustered by POIs[6]. With this the authors get to understand the spatial-temporal application usage behaviors and predicted the top N popular applications over BSs based on a data-driven approach for cache implementation. Other researchers used a deep packet inspection at the network operator level and obtained a geo-tagged dataset with one week. They develop a technique that leverages transfer learning to predict which applications are most popular and estimate the whole usage distribution based on the Point of Interest (POI) information of that particular location[28]. Their findings pave the way for predicting which apps are relevant to a user given their current location, and which applications are popular where.

Liao et.al[29], uses temporal-based APPs Predictor to dynamically predict the APPs which are most likely to be used.The author extracted three APPs usage features, global usage feature, temporal usage feature, and periodical usage feature from the APPs usage trace. Then, based on those explored features, They dynamically derive an Apps usage probability selection algorithms(MaxProb and MinEntropy)model to estimate the current usage probability of each APP in each feature. Using the usage probability in each feature ,the author select k APPs with the highest usage probability from the probability model.

Several factors could impact people's APP usage behavior on smart phones [18]. Existing research verified that current time, location, travel pattern and historic APP usage behavior could have have an influence on the current APP usage behavior [19]. Meng et al. [20] found that user's personality, surrounding environment and location may dramatically impact APP usage behavior by analyzing internet traffic generated by cell phone from a campus. For example, students prefer to use the Android browser in the classroom, but watch sports and listen to streaming music in the dorm. Zhang [1] explained the advantages of using mobile traffic data to analyze user mobility and also

built an improved Markov chain model including time, location, and historic app usage behavior to predict app usage behavior with 90 % prediction accuracy for 80% of users.

Some authors [30] [26][31] studied the user mobility and APP usage whereas [31] study mobility based on location only. Qiao et.al [30] used a chi-squared algorithm to select the most significant mobility features that may influence APP usage behavior.The study particularly focused on three aspects of human mobility in urban areas which are individual mobility characteristics, location and travel behavior from both the crowd and individual points of view. Support Vector Machine (SVM) is used to forecast a limited number of selected mobility and time features in terms of APP usage behavior of crowds and individuals for high accuracy. Furthermore, Sun et.al[31] studied user mobility and location behavior by using Hadoop based mobile big data processing platform together with a systematic mobility analysis framework. The study focused on three aspects: the occurrence patterns, the mobility patterns, and locations by using k means, radius gyration and the user spends the top percentage of all network consumption (traffic and duration) respectively.

Previous usage behavior predictions are based on APP usage interest, time and location provided by smart phone sensors and, mobility characteristics, whereas in this paper, we propose a method based on machine learning to predict user APP usage behavior by using location, time of the day and day of the week features.[6] papers cluster location based on point of interest but clustering Addis Ababa city by point of interest is impossible because our population settlements is mixed. In this paper we cluster the location based on traffic demand to understand APP usage demand per cluster location . We used users favorite APP usage behavior as a categories label.We will analyze each cluster location, data usage per cluster and predict three popular APP usage per each cluster by using random forest regression algorithm.

Chapter 4

Experimental Design

In this Chapter, the overall experimental process followed in conducting this research is discussed. The experimental process is illustrated in block diagram shown in Figure 4.1,The process is composed of three components: the first part is data collection and processing while the second part is location clustering, and analysis. The last one prediction of the popular APP per each cluster. The model has been developed to understand the APP service usage distribution based on location in a different time to handle the problem of performance degradation of the network and managing the network resources. Details of tasks done under those modules have been described as follows.

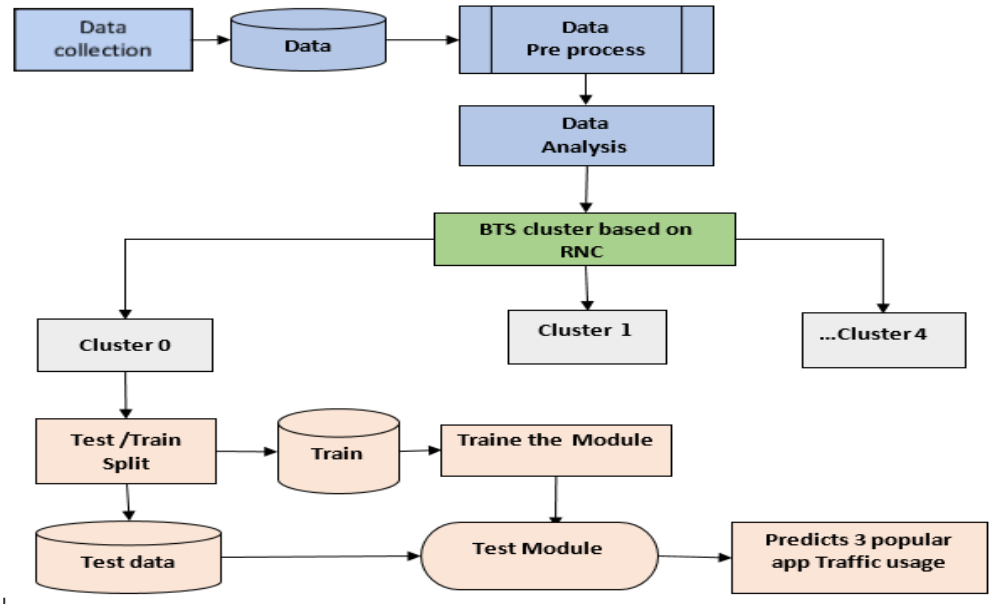

Figure 4.1: Experimental process Block diagram

4.1 Data Collection and Analysis

4.1.1 Data Collection

For this research, the data has been collected from the ethio telecom 3G mobile network in Addis Ababa city. The high-level view of a mobile network has been shown in Figure 4.2. It has three major components in one mobile network as stated in Chapter Two, including mobile devices, radio access control, and core network. A mobile device has a terminal to connect the mobile network to 3G(UMTS) , request data has been collected by Node Bs, and sent to RNC. Core Network contains SGSN and GGSN network components. Gn interface has been placed between SGSN and GGSN. GGSN has an interface that transmits the data traffic to the Internet through the Gi interface.

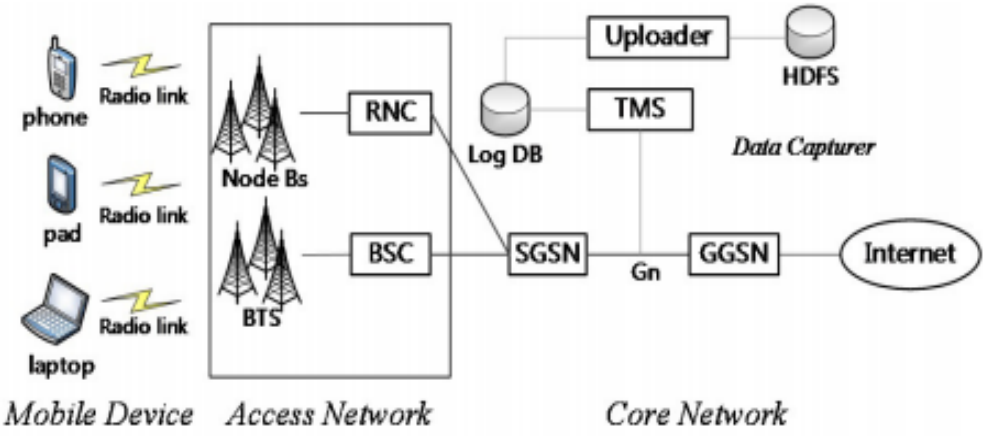

Figure 4.2: Mobile network architecture[2]

We have collected mobile APP service usage traffic information from TMS through Gi interface. The collected information contains, traffic volume (MB) generated by 6968 mobile network cells for each hour of a day and, the coordinate data (in degrees) of 734 BSs located in Addis Ababa city. Overall, the collected data contains over 10180 different APPs, and we have collected seven days of data, from June 01 to June 07, 2019.

Before all analysis stages, data preprocessing was performed. In several cases, data might contain outliers or noise which will chunk a clear understanding of the important information. During this work techniques like standardization and, generalization has

been used to eliminate possible noise from the information. In addition , as the volume of APPs within the dataset is massive, we tend to reduce the data by categorizing APPs by the similarity of their function[28]. The 10180 APPs thus are classified into twelve categories manually. Some of the categories are email, streaming, and so on. The categorized APPs with their description have been shown Table 4.1.

Table 4.1: APPs categories List

Category	**Application category description**
Email	Mail service
File_Access	file transfer app
Finance	Bank service app
Game	Online game
IM	Viber ,Facebooks messenger
Navigation	Google Map ,Apple Map
P2P	point to point app
SNS	simple Notification Service
Stock	
Streaming	Social network
VoIP	Voice over Ip services
Web_Browsing	Browsing

Using the collected data, we have got twelve APP categories. However, we chose six APP categories that have the highest traffic usage for our analysis. The selected categories are Web Browsing, VoIP, Streaming, SNS, IM, and File Access and The APPs in these six categories consume 83% of the total bandwidth used by all APPs. The selected application with their description and some examples category have been demonstrated in Table 4.2 .

4.1.2 Analysis

Before experimenting, we analyze the traffic data from three perspectives, time of the day, day of the week, and location. Those analysis help to understand the collected data distribution per location, and APP category usage. For APP distribution per location, we have to consider sample location to prove traffic demand rate distinction one location to others with a similar time and capacity of the BSs.

Table 4.2: Selected six APPs categories description

App Category	Description	App Example
File Access	File transfer app	Google Play_File Access ,HTTP _Ext_File Access, App Store_File Access
IM	Instant message	Tango_IM,
SNS	Simple Notification service	Twitter_SNS,Google+ _SNS,Whats Up _SNS
Streaming	Social network	You Tube_Streaming ,Face book _Multi Media
VoIP	Voice over Ip services	vibe,skip
Web_Browsing	Browsing	HTTP_Browsing, Google Search_Browsing,

To understand the temporal characteristics of the traffic consumption pattern with the dataset, we analyze APP category usage patterns within a cluster BSs in the time domain[5]. We tend to show the temporal characteristics of traffic consumed by different categories of APPs with time, day of the week and location. At the same time, we try to observed essential APP usage distribution.

The data traffic generated in every base station is very dynamic at different times of the day and on different days of the week. Additionally, different locations have different traffic demand as a result of APP usage .Therefore, we clustered the BSs based on location and traffic usage demand[4].

4.2 Location Clustering

A helpful method that used to understand mobile APP usage behavior is a location cluster based on their traffic demand. In our case, we analyze the APP usage pattern in different clusters so that the network supervisor will apply various resource allocation methods on different BSs and suggest different categories of APPs to mobile APPs users within the coverage of various BSs. Additionally, we wanted to determine what clusters contain similar points of interest-based on the similarity of usage behavior[4]. The final step of this cluster methodology is to show the correlation between the location and the app usage behavior since this type of knowledge will enable the network provider to allocate and planning resources based on location and APP traffic demand.

The idea behind the clustering approach defined here is to divide the larger problem into a variety of smaller issues, which might be solved individually using an easy formulation[18]. At the same time, the cluster method applies in this paper to understand the different Addis Ababa 3G network APP usage characteristics, and APP popularity in a cluster BSs based on location and traffic usage.

In this paper, we are using existing clustered ethio telecom Addis Ababa BSs based on RNC label by using K-mean clustering techniques, regrouping the cluster and analyze the traffic pattern and APP distribution per cluster. This helps to understand and predict the traffic usage at different locations.

4.2.1 APPs Usage Distributions Per Cluster

The APP usage under different clusters location have been explored in this subsection. The details description are shown in Figures 4.3. It presents the number of logs generated by different categories of APPs and its traffic consumed by different APPs categories under each cluster of location by there BSs. Web Browsing generates the highest volume of traffic and is the most popular app type in Addis Ababa city. The traffic distribution of various APP types in BS clusters “1” and “4” are similar in traffic usage. Note that streaming APPs generate the second most logs in all clusters, and clusters ”1” and ”4” have the highest traffic usage compared to other clusters.

The proportions of different categories of APPs used are also different under different clusters location within a given time duration. We observe that the traffic pattern of a base station is highly dynamic under different clusters and times. Based on the observations, we propose to predict popular APP traffic demand per cluster to characterize the traffic pattern of each cluster and understand the APP usage traffic demand per cluster .

4.3 Popular APP Usage Traffic Prediction

Accurately predicting the APP types that typically utilize the most traffic under each cluster BSs during a specific period can provide more important guidelines to mobile network operators[21]. Operators should properly reserve valuable resources for valuable service typically requested by video and news APPs containing streaming content.

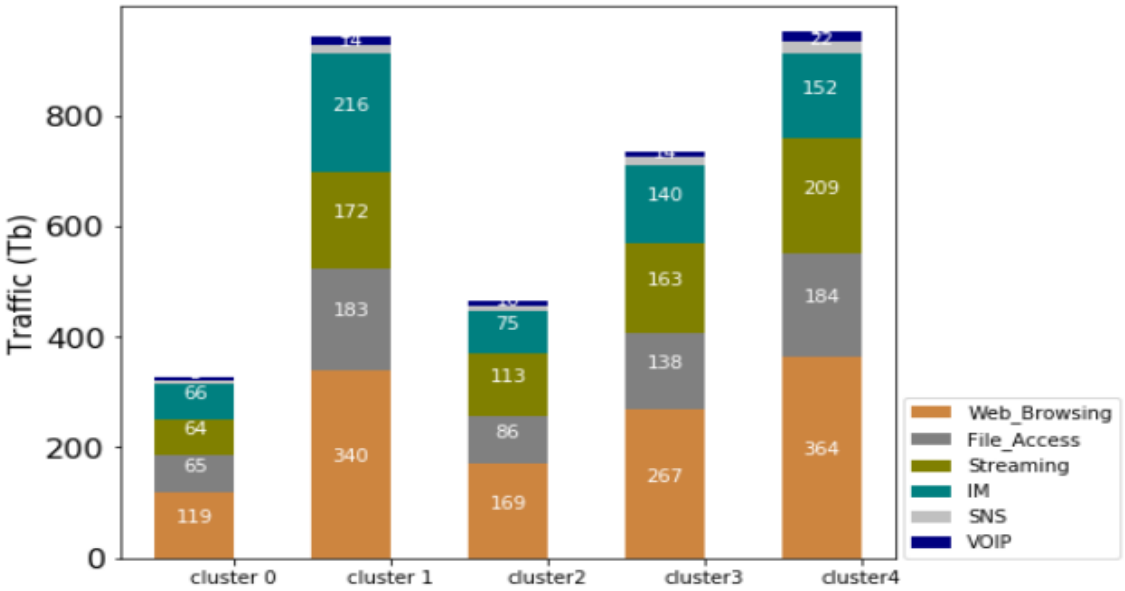

Figure 4.3: APP Distribution per each cluster

For the resource requested by game APPs, the mobile operators should guarantee low packet loss probability.

APP usage traffic prediction performs a key role in network operations for effective management of today's increasing network data traffic[29]. It makes an accurate prediction of future traffic and carefully constructs regression model that is capable of formulating an accurate relationship between future traffic volume and previously observed traffic volumes.

As shown in Figure 4.3, the proper distribution of the specific number of logs generated by various categories of APP types under each specific cluster of BSs is unrelated in a specific time. Meanwhile, the temporal characteristics of traffic consumed by different categories of APP types typically differ on typical weekdays and weekends. We propose to predict the top three APP types in each cluster during a specific time, by using the XDR record data. We select the top three APP types to accurately predict because we typically find that the top three app types make up almost 60% of ordinary traffic under each BS cluster when preprocessing the dataset used in our experiment.

To predict the top three APP types in each cluster during a specific time, we decompose the problem into two steps. Initially,we rank the predicted traffic consumption and

get the top three app types in each cluster. Next, we predict the traffic usage of each APPs type under each cluster during a specific period time.

4.3.1 Model Selection

We select the RF regression model, among the most popular machine learning algorithm .RF has a good accuracy, strength and, ease of use. In addition, the random forest model is very good at handling tabular data with numerical features, or categorical features with less than hundreds of categories. Unlike linear models, random forests can capture non-linear interaction between the features and the target.

RF is an ensemble learning algorithm. The basic premise of the algorithm is that building a small decision-tree with few features is a computationally cheap process [32]. If we can build many small, weak decision trees in parallel to overcome the accuracy of simple prediction and to avoid possible over fit, we can then combine the trees to form a single, strong learner by averaging or taking the majority vote. In practice, random forests are often found to be the most accurate learning algorithms to date. The working principle is illustrated in Algorithm 1.

We will build the model on the training set and check the accuracy of the model by using it on the testing set. Using the python Splitter packet, the data is split in such a way that 80% of records fall under the training set and, 20% of the records are used for testing the model. Our resulting training set has 134 observations, and the testing set has 34 observations. After splitting, the set is used to train the Random Forest model and also for tuning the Random Forest parameters. The tuned model is then used to predict APP traffic on the test set.

4.3.2 Finding Best Parameters

Machine Learning models usually have a set of parameters that should be tuned for a given collection to achieve the maximum possible value[1]. Parameters in the random forest are used either to increase the predictive power of the model or to make it easier to train the model. We used K-Fold Cross Validation not only to find the best model but also to come up with the correct set of parameter values. Here we will tune the parameters while we run K-Fold Cross-Validation. In this work, a good way of visually proving

potentially optimized values of model parameters,a Grid Search Cross-Validation and Curve fitting cross validation parameter tuning meted are used.

k-Fold Cross-Validation

Cross-validation is a re-sampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that the given data sample is to be split. As such, the procedure has regularly called k-fold cross-validation. When specific value k has chosen, it may be used in place of k in the reference to the model, for this experiment we have choosing k=10 .

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, use a limited sample to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on new data. This is the basic idea for a whole class of model evaluation or tuning parameter methods.

Here's a basic diagram of 10-fold cross validation:

So, as you can see, the dataset is split into 10 parts. The model is then trained-and tested 10 times, each time on a different 9 parts, and tested on the tenth. By the end, you can aggregate all of your test results into a dataset, and compare the predictions with the true values. For regression, when you compare prediction with true values and it suggests using r-squared values.

4.3.3 Validation Curve With k-fold Cross-Validation

A validation curve can be plotted on a graph to show how well a model performs with different values of a single parameter. Following are the parameters description, parameter tuning,validation curve of each parameter and parameter selection are we will be talking about in more details.

n_estimators: It defines the number of decision trees to be created in a random forest. Generally, a higher number makes the predictions stronger and more stable, but

Figure 4.4: Basic diagram of 10-fold cross validation

a very large number can result in higher training time.However, adding a lot of trees can slow down the training process considerably, therefore we do a parameter search to find the sweet spot.

max_features : It defines the maximum number of features allowed for the split in each decision tree. Increasing max features usually improve performance but a very high number can decrease the diversity of each tree. There are multiple options available in Python to assign maximum features: auto, None and sqrt. Auto and none are the most frequently used options. Both mean taking all the features. Here we simply do not put any restrictions on the individual tree.Sqrt This option will take the square root of the total number of features in an individual run. For our work we set the max feature value to auto.

max_depth: Random forest has multiple decision trees. This parameter defines the maximum depth of the trees. The default value for max_depth is None, which means that each tree will expand until every leaf is pure. A pure leaf is one where all of the data on the leaf comes from the same class.

min_samples_leaf: This defines the minimum number of samples required to be at a leaf node.Leaf is the end node of a decision tree .Smaller leaf size makes the model more liable to capturing noise in train data. The default value for this parameter is

1, which means that every leaf must have at least one sample that it prediction.

min_samples_split: This parameter specifies the maximum number of leaf nodes for each tree.The tree stops splitting when the number of leaf nodes becomes equal to the max leaf node.The default value for this parameter is 2, which means that an internal node must have at least two samples before it can be split to have a more specific prediction.

n_jobs: This indicates the number of jobs to run in parallel. A value of "-1" means there is no restriction whereas a value of "1" means it can only use one processor. We chose -1 n_jobs values for this experiment.

random_state: This parameter is used to define the random selection. It is used for comparison between various models.A definite value of random_state will always produce same results if given with same parameters and training data. I have personally found an ensemble with multiple models of different random states and all optimum parameters sometime performs better than individual random state.

4.3.4 K-Fold Cross-Validation With Grid Search

Another way to choose which parameters to adjust is by conducting a grid search or randomized search. A grid search takes in as many parameters as you would like, and tries every single possible combination of the parameters as well as as many cross-validations as you would like it to perform. A grid search is a good way to determine the best parameter values to use, but it can quickly become time-consuming with every additional parameter value and cross-validation that we add.

In grid search, the possible values for each of the variables are specified and, based on those, all the potential combinations are generated and tested. For instance, if we have the parameters and values a = [1, 2, 3], b = [0.1, 0.2,0.3], the grid search will generate 9 possible configurations for the regression with each possible combination: [a 1, b 0.1, a 1 b 0.2, ... a 3, b 0.3]. Furthermore, cross-validation is applied to evaluate and select the best setting. All this process is very well supported in python using sklearn.

To find the optimal number of estimators is by using GridSearchCV, also from sklearn. I just give it an estimator, param_grid and define the scoring, along with how many cross_validation folds. What I have done here is make the GridSearchCV find the max_depth,n_estim

and min_samples_leaf by giving its range, and making random state, n_jobs and max_features values 0,-1 and auto respectively.

4.4 Building RF Model

After tuning the parameters, we chose the best model based on the smallest RMSE and highs r^2 score value of the parameter tuning methods. We are select the model having less RMSE and, highs r^2 value. By using the tuned parameter result we build the RF model. Once the parameter selection and the model selection task is completed, the next step is to train the selected algorithms and, building the predictive model.

4.5 Evaluation Metric

Evaluating and comparing the predicting performance of the algorithms, how well they predict the traffic demand of the APP category; is the main objective of this research. Two techniques, RMSE and R square, have been used to evaluate the performance of the algorithms.

RMSE

The most commonly used metric for regression tasks is RMSE (Root Mean Square Error). This is defined as the square root of the average squared distance between the actual score and the predicted score:

$$RMSE = \frac{1}{n}\sqrt{\sum_{t=1}^{n}(y_t - y_p)^2}$$

Here,y_t denotes the true score for the i-th data point, and y_p denotes the predicted value. One intuitive way to understand this formula is that it is the Euclidean distance between the vector of the true scores and the vector of the predicted scores, averaged by$\sqrt{n}$, where n is the number of data points[33].

The MSE metric measures the average of the squares of the errors or deviations. MSE takes the distances from the points to the regression line (these distances are the “errors”) and squaring them to remove any negative signs. MSE incorporates both the variance

and the bias of the predictor.

MSE also gives more weight to larger differences. The bigger the error, the more it is penalized. For example, if your correct answers are 2,3,4 and the algorithm guesses 1,4,3, then the absolute error on each one is exactly 1, so squared error is also 1, and the MSE is 1. But if the algorithm guesses 2,3,6, then the errors are 0,0,2, the squared errors are 0,0,4, and the MSE is a higher 1.333. The smaller the MSE, the better the model's performance.

R2

R-squared (r^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable[34]. The r^2 value varies between 0 and 1 where 0 represents no correlation between the predicted and actual value and 1 represents complete correlation.

$$R^2 = 1 - \frac{\text{Explained Variation}}{\text{Total Variation}}$$

Or

$$R^2 = 1 - TotalVariation$$

Limitations Of R-Squared will give you an estimate of the relationship between movements of a dependent variable based on an independent variable's movements. It doesn't tell you whether your chosen model is good or bad, nor will it tell you whether the data and predictions are biased. A high or low R-square isn't necessarily good or bad, as it doesn't convey the reliability of the model, nor whether you've chosen the right regression. You can get a low R-squared for a good model, or a high R-square for a poorly fitted model, and vice versa.

Chapter 5

Results and Discussion

In this Chapter, the results of the experiment will be discussed. The experiments were done according to the procedures outlined in Chapter Four.

5.1 Analysis Results

APP usage distribution per time and day of week

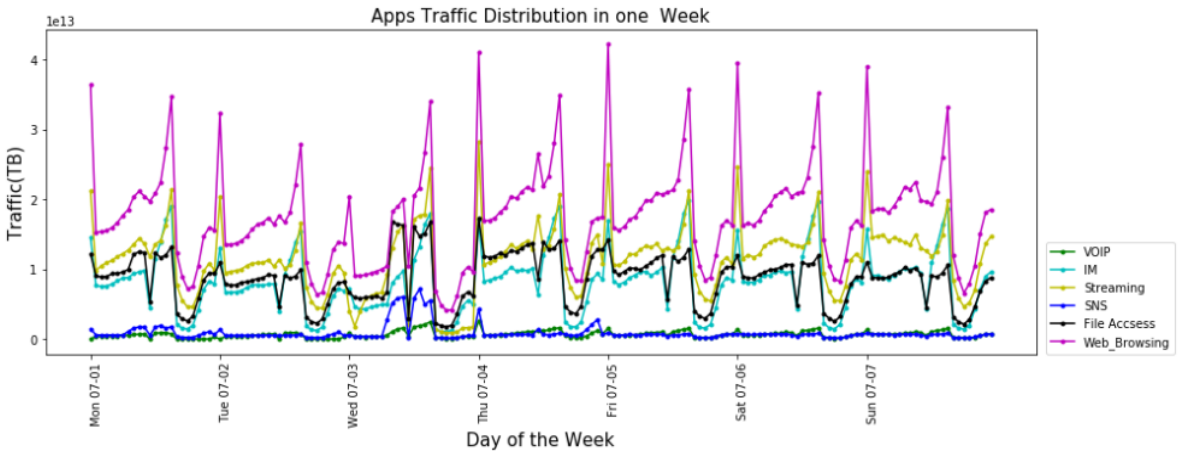

Figure 5.1: APP traffic distribution in one week

Figure 5.1 shows the data traffic volume of different APPs for seven days. The traffic has a repeatable pattern with a one day cycle. This could be explained by the actual fact that almost all users have regular APP usage traffic demand on week. In addition to that

the data traffic usage increase after 10:00 PM of the day until midnight and decrease from midnight to 08:00 AM(see Figure 5.2). At the data traffic increment time , users usually finish their daily work and browse different APP services like scanning daily news, social media and, watching movies. The increase in user APP usage causes high traffic within the cellular network. The above analysis proves, APP usage traffic demand depends on the time of the day and, day of the week.

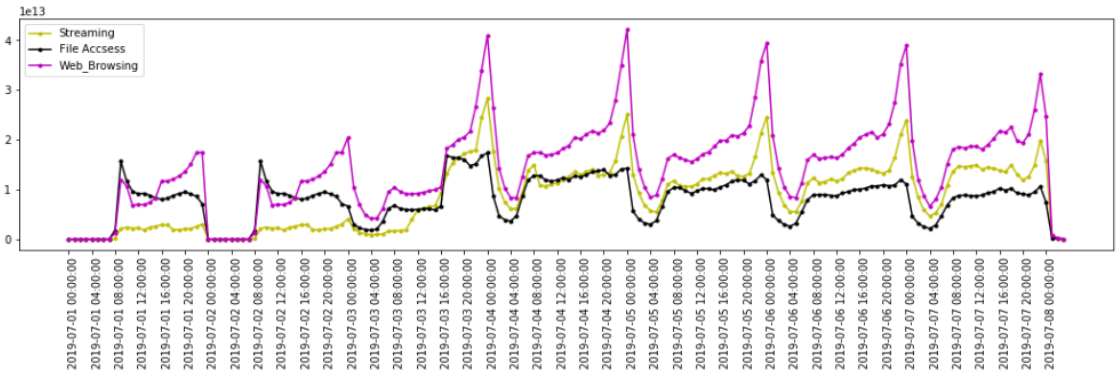

Figure 5.2: Streaming, File Access and Web_Browsing traffic distribution in one week

APP Usage Distribution per Location

Figure 5.3 shows the APP usage distribution in two locations. The selected locations have different population settlement. This helps to understand APP usage in different locations with different characteristics. The first selected location is business area (Mercato), represented as a blue solid line. At this location, we observe peaks throughout operating hours whereas, the second location traffic distribution in the more residential area (Jemo), denoted as a green line has reasonably higher throughout in the evening hours. Such a spatiotemporal non-uniform property of traffic demands is a good opportunity for operators to optimize the resources prices and operational expenses of their network infrastructures.

On one hand, the capability of every base station reaches its peak traffic capacity, resulting in a high operation price. On the other hand, the capability within the individual base station has been wasted different hours. For a business area like merkato resource is wasted at night time whereas for residential area, resource is wasted during the day time. Generally, APP usage depends on location and understanding APP usage per loca-

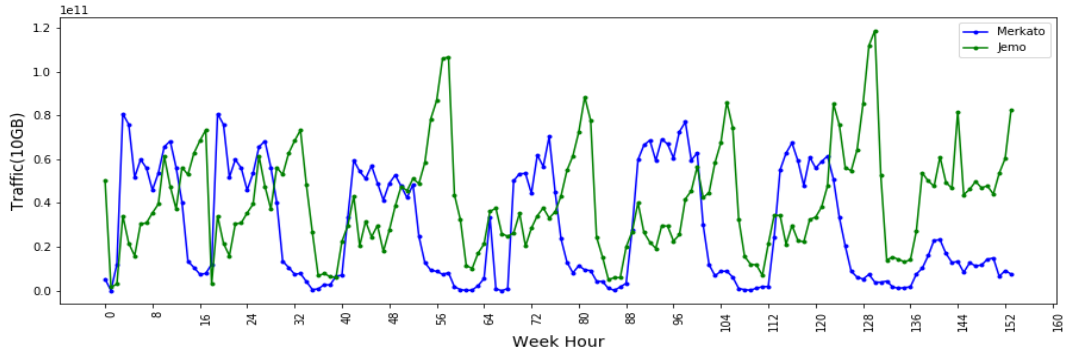

Figure 5.3: Selected sites Total Traffic Distribution

tion has important impact on optimizing, and handling the network resource properly.

The data traffic generated in each base station is highly dynamic at a different time of day and on different days of the week. For example, the traffic volume in business districts may be higher during the working hours than at midnight, and be lower at weekends than on weekdays in other districts(see Figure 5.3). The traffic pattern of the residential area is opposite to that of the business areas. Therefore, we need to build appropriate data traffic profiles to characterize the data traffic patterns of base stations under different temporal contexts.

5.2 Cluster Analysis Result

In this paper, we have used existing ethio telecom base station clusters based on RNC. The scatter plot of the cluster is shown in Figure 5.4. It indicates, Cells clustered into five and,the clusters contain 871, 1912, 988, 1156 and, 2043 number of Cells in cluster 0 to cluster 4 in respective order. The distribution of the Cells is dense in the center and it gets sparse as we go away from the center.

The traffic usage pattern per cluster is shown in Figure 5.5. The cluster's traffic usage pattern the same in all cluster but its traffic volumes are different from one other. The traffic pattern is the same because each cluster contains different points of interest. point of interest can be associated with specific urban and economic functions such as shopping,education, or entertainment. As such,point of interest characteristics the so-

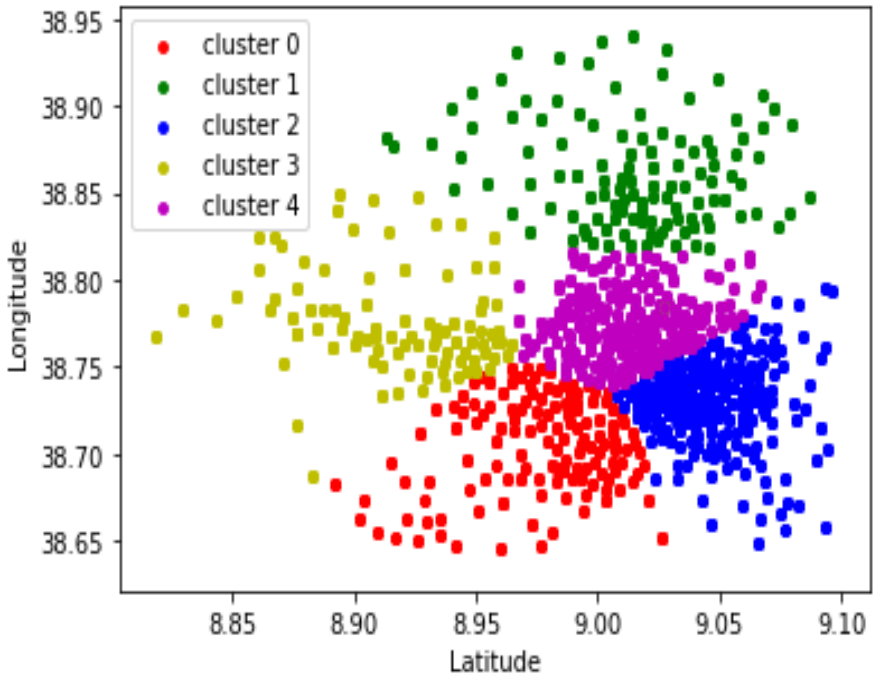

Figure 5.4: Cells cluster based on RNC

cioeconomic function of a location serve [6].

Each cluster Usage has a group of different points of interest like businesses, shops, and the education area. Their combined usage becomes similar to one cluster to other. The traffic demand difference results from the different number of Cells. The traffic demand are increases when increasing number of BSs(see Table 5.1). In another direction, the user's data traffic demand depends on the user's APP usage interest of the location.

Addis Ababa traffic distribution per cluster per APP category has shown in Table 5.1, each cluster has different APPs category interest, traffic demand and number of Cells. The traffic usage of each cluster increases when the number of cells increases. This indicates Ethiotelecom increase traffic demand by implementing additional cells based on the location traffic demand. It has its disadvantages, like increases implementation cost and, network interference. A better way of managing the existing network resource by implementing a resource pulling system or cache server based on user APP usage and, predict traffic demand per location and, time. That will decrease implementation cost and network interference.

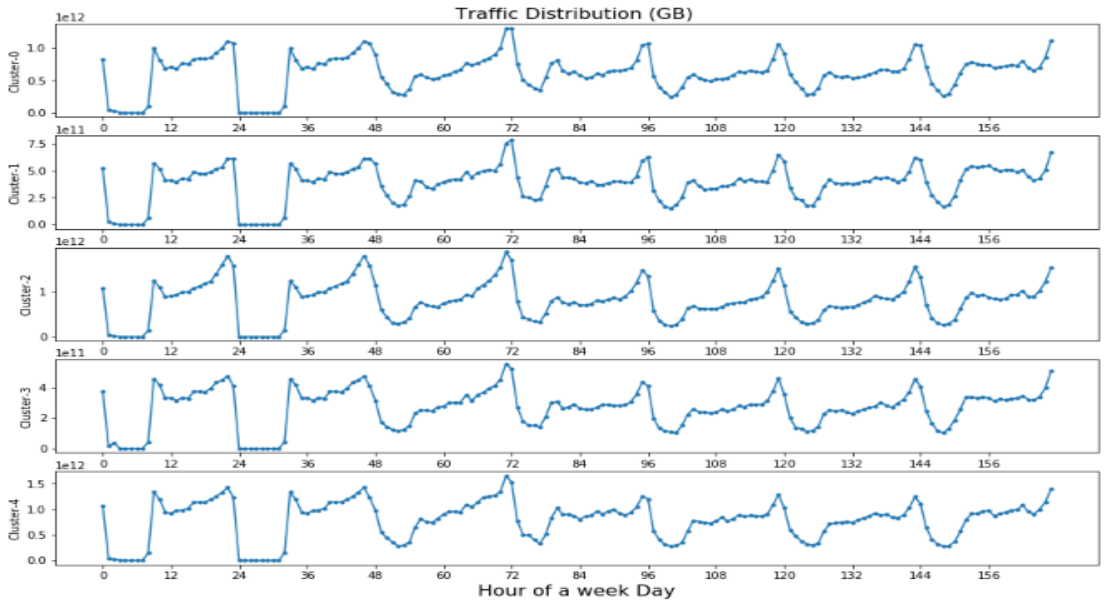

Figure 5.5: Traffic Distribution pattern per cluster

Table 5.1: Traffic Distribution per cluster per APP category

Cluster	No cells	Total traffic	File Access	Streaming	SNS	VOIP	Web browsing	IM
0	871	328TB	65.2TB	64.3TB	5.85TB	6.72TB	120TB	66TB
1	1912	942TB	183.04TB	173TB	14.5TB	14.4TB	341TB	216.302TB
2	988	464TB	86.4TB	114TB	10.3TB	7.89TB	170TB	75.6TB
3	1156	735TB	139TB	163TB	14.1TB	11.5TB	268TB	140TB
4	2043	950TB	184TB	209TB	22.5TB	17.5TB	364TB	15.3TB

5.3 Parameter Tuning

The validation curve for n_estimators is shown in Figure 5.6. This validation curve was created with the values [100, 200, 300, 400 ... 900] as the different values to be tested for n_estimators. In this figure, we see that when testing the values, the best value appears to be 600. It is important to note that even though there appears to be a large difference between the training and cross-validation score, the training set had an average score of 97.5% for each of the ten cross-validations and the cross-validation had set 80% accuracy for all the values of n_estimators, which shows that this model is very accurate irrespective of the number of estimators used.

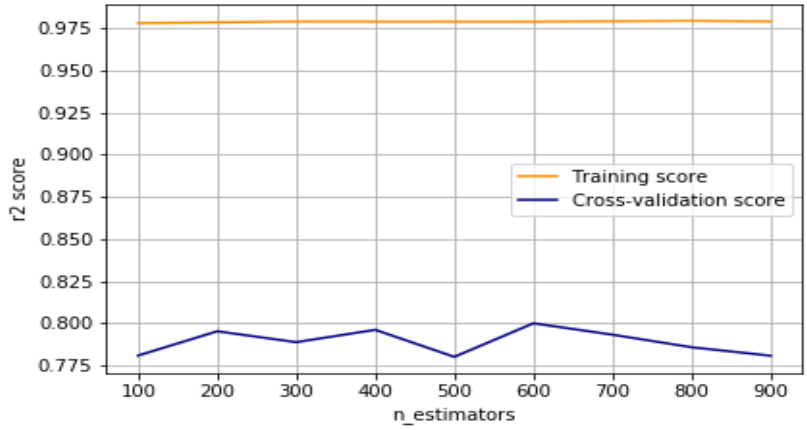

Figure 5.6: n_estimator validation curve

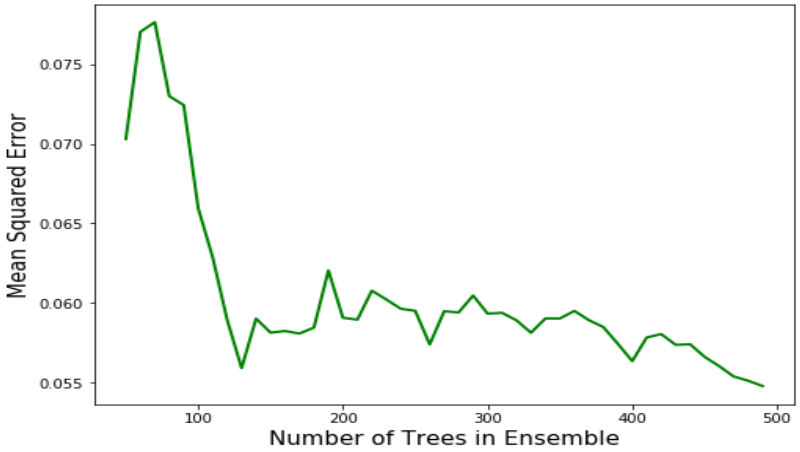

Figure 5.7: Error Vs n_estimators Curve

Generally, as shown in Figure 5.7 n_estimators increases as error value decrees and the model sterility increase . A higher number of n_estimators makes the predictions stronger and, more stable but a very large number can result in higher training time. To avoid this problem we choosing the value 200 for our model.

In Figure 5.8, the validation curve was created with values [5, 10, 15, 20... 45] as the different values to be tested for max_depth. We see that the highest score value on the cross-validation is close to 86% when the max_depth is set to 15. Hence, for our model, we used a max_depth value of 15. While it may seem better to choose a max_depth of 25, because that value has the highest accuracy for the training score, we have not selected

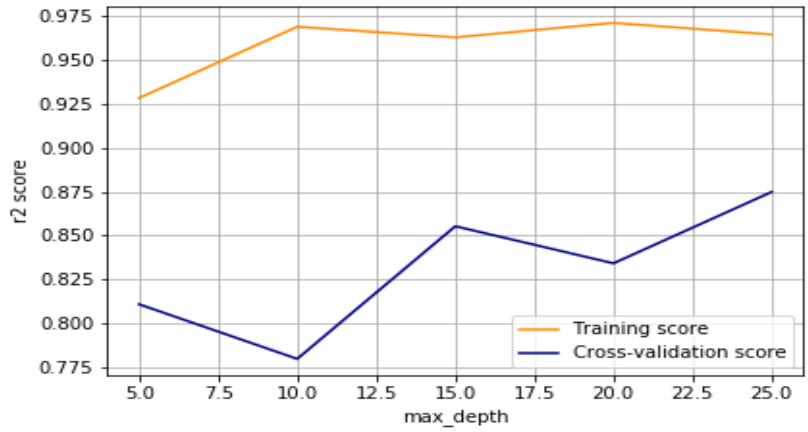

Figure 5.8: max_depth validation curve

this value to prevent our model from over-fitting in the training data.

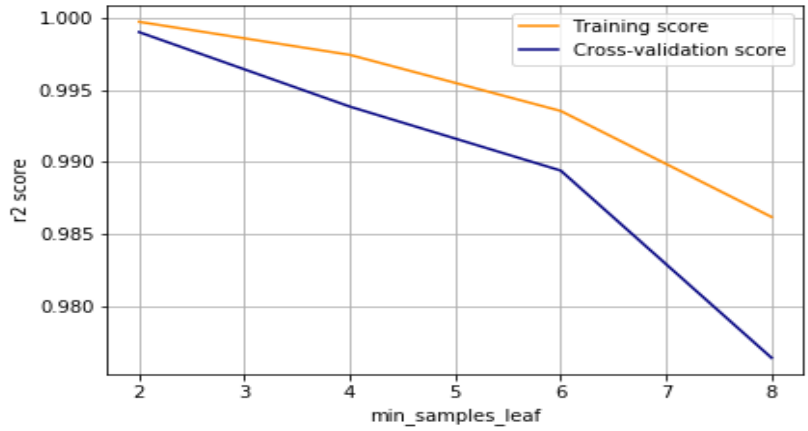

Figure 5.9: min_samples_leaf validation curve

In Figure 5.9, we see that the score goes down for both the training and cross-validation sets for each additional increase in the value of min_samples_leaf. Hence, we will choose 1 for the value of min_samples_leaf parameter, which is also the default value.

In Figure 5.10, we see that the score goes down for both the training and cross-validation setsas the value for min_samples_split is increasing. Hence we will choose 2 as our value for min_samples_split. In this case, it makes sense that we would want a lower value for min_samples_split. The default value for min_samples_split is also 2. As

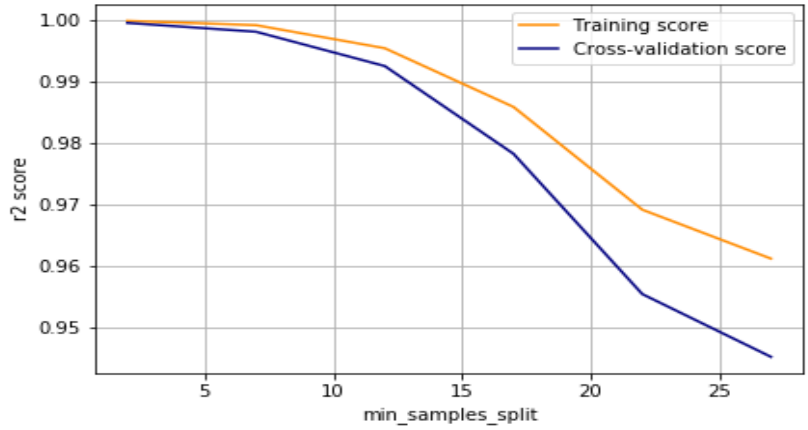

Figure 5.10: min_samples_split validation curv

we choose higher values for the minimum number of samples required before splitting an internal node, we will have more general leaf nodes, which would harm the overall score of our model.

A model would be constructed using the chosen parameters and the training set , and then would be tested on the testing set to see how accurately the model can predict the value. When tested on the testing set with the default values for the parameters, the values of the testing set were predicted with an r^2 and Root Mean Square Error value of 0.99153 and 0.08477 respectively. The tuned model resulted in an accuracy of 0.99307, which was more accurate than default model, by .0015.

By using Grid search cross-validation result, which gives the lowest error value from agiven list, the best parameters are as follows, max_ depth = 5, min_ samples_ leaf = 1, min_ samples_ split = 2 and, n_ estimators = 800.

Table 5.2: Parameter Tuning result

Parameter	**Default Value**	**Validation Curve**	**Grid Search**
max_depth	None	15	5
min_sample_leaf	1	1	1
min_sample_split	2	2	2
n_estmators	10	600	800
RMSE	0.0353	0.018	0.0185
r^2_score	0.996	0.998	0.999

The results of the grid search and the validation curve tuned parameter resulted as shown in Table 5.2 . We have chosen the grid search parameter tuning result, this improved our Default model by decreasing RMSE on the testing set by 0.0800. Grid search Parameter tuning can be advantageous than the validation curve by creating a better prediction model and, very time consuming as well[35].

5.4 Popular APP Prediction Result

The APP usage under different clusters is investigated in this subsection. As shown in the Table 5.3 in cluster one 36.% of traffic usage is due to the web browsing, while 20.40% of data traffic is due to the streaming APP category. We can observe that the traffic distribution of various app types, SNS and VOIP category have the lowest traffic usage under all clusters.

Table 5.3: Percentage Traffic distribution per cluster per category

Traffic category	**percentage traffic per cluster**				
	cluster 0	**cluster 1**	**cluster 2**	**cluster 3**	**cluster 4**
Web browsing	36.49%	36.19%	36.58%	36.39%	38.31%
Streaming	19.67%	18.33%	24.55%	22.24%	22.03%
File Access	19.88%	19.44%	18.63%	18.85%	19.39%
IM	20.12%	22.97%	16.31%	19.05%	16.07%
SNS	1.78%	1.54%	2.22%	1.91%	2.37%
VOIP	2.05%	1.53%	1.70%	1.56%	1.84%

The popularity of different categories of APPs used is the same under different clusters. The Table 5.3 shows the distribution of the number of logs generated by different categories of APP types under five clusters locations. Web browsing APPs generate, the most logs and, traffic in every cluster (see Figure 5.11) thus, web browsing APPs can be regarded as the most popular APP type in this city. Generally we have observed that web browsing,streaming and File Access APPs category popular in all clusters.

Due to similarity, we only show the top three APP category prediction plots for cluster zero. Figure 5.12 shows the visualization prediction accuracy of Web browsing, Streaming and, the File Access APP category for cluster 0 by using the RF model. The green line indicates the training set while the rad, and blue dote line presented the testes, and the

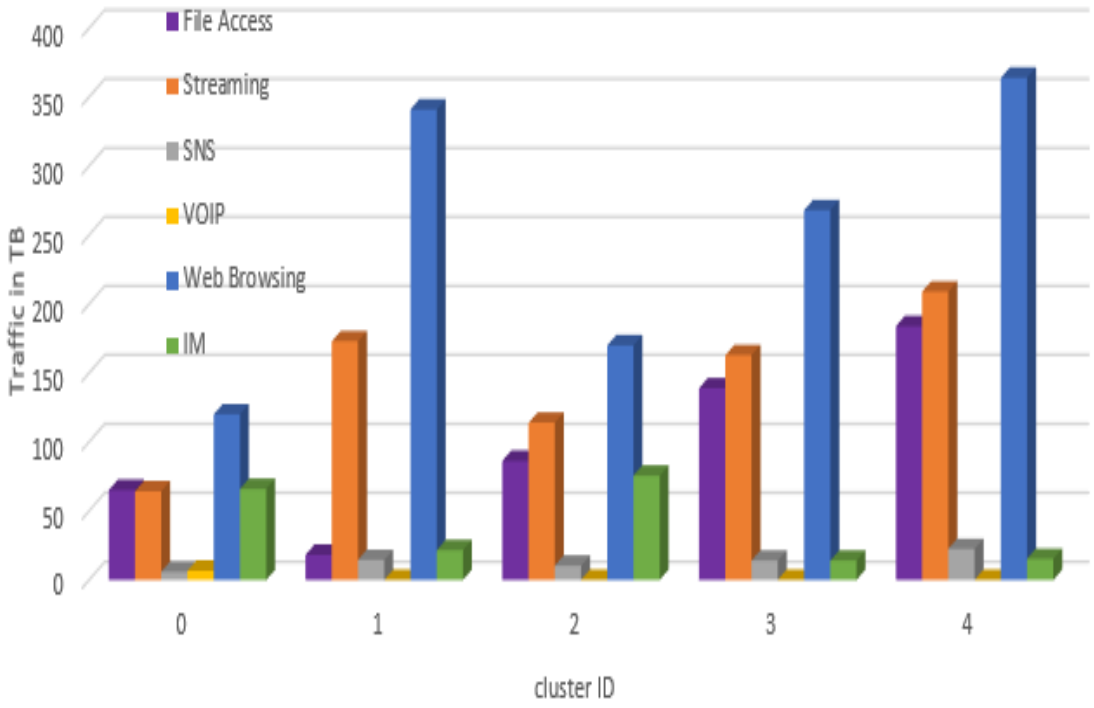

Figure 5.11: APPs category Traffic demand per cluster

predictive result respectively. we have observed that prediction accuracy is good.

Prediction Result Evaluation

The performance of the models has measured by using their r^2 and MSAE values. Two models have compared to MVR and, RF models. In the first cluster the prediction result shows for the File accesses category, the MVR model's score and, MSAE values are 0.45 and 9.78 respectively and for the RF model and prediction score, and MSAE values are 0.95, and 0.089 respectively.

To compare the two prediction models, MVR and RF, we used the Web browsing data from cluster zero. The comparison is done using the error of the fitted model. The error from a fitted model is the differences between the responses observed at each combination values of the test values and, the corresponding prediction of the response. The prediction result plot in Figure 5.13 shows that the prediction did not fit the actual result. This is due to the non-linearity characteristic of the data computed using the MVR. On

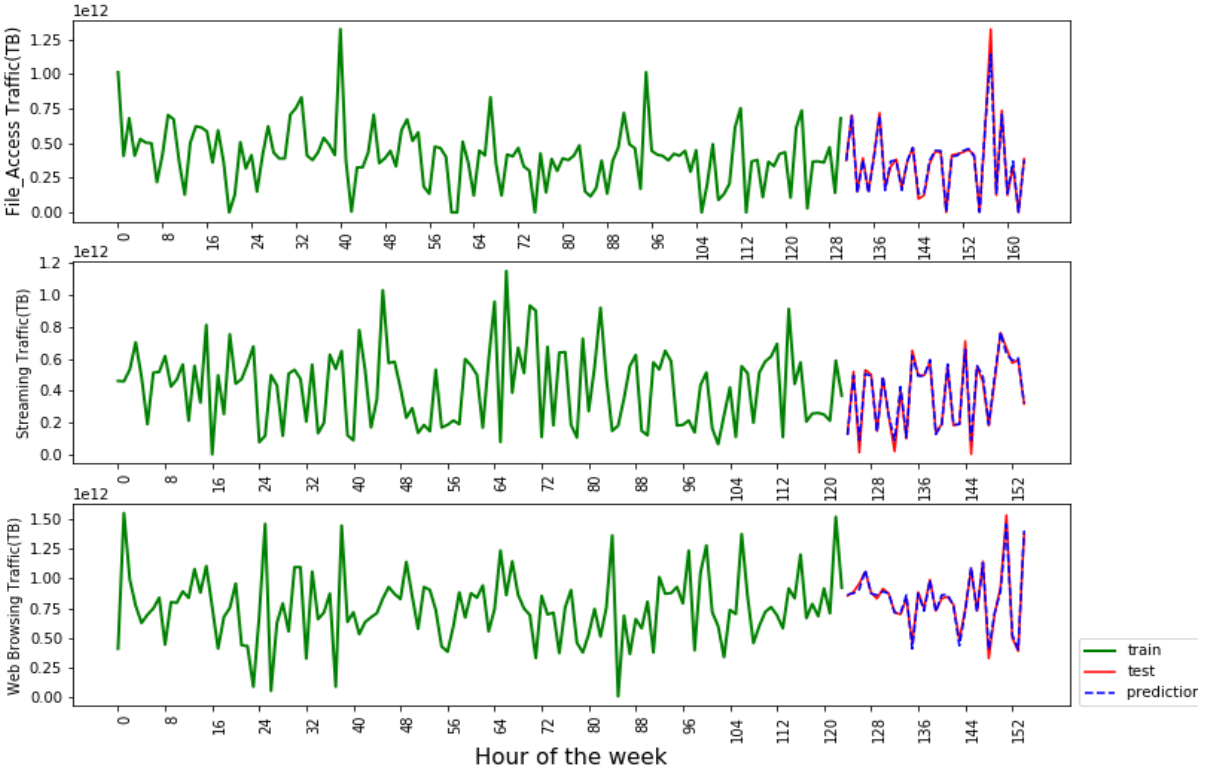

Figure 5.12: Prediction out put of APPs for the first category

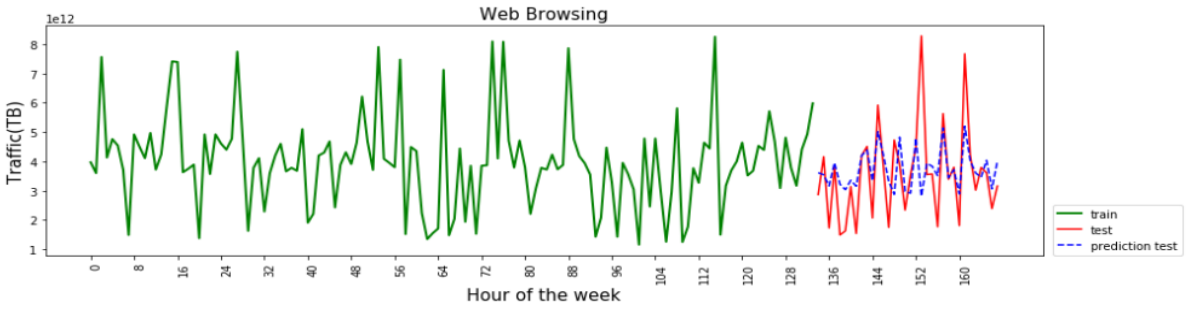

Figure 5.13: Prediction out put of Web browsing by MVR Model

the other hand, prediction performed by using the RF model fits with the actual value (see Figure 5.14).

Hence, in this research, we select RF model for predicting APP traffic usage.Unlike

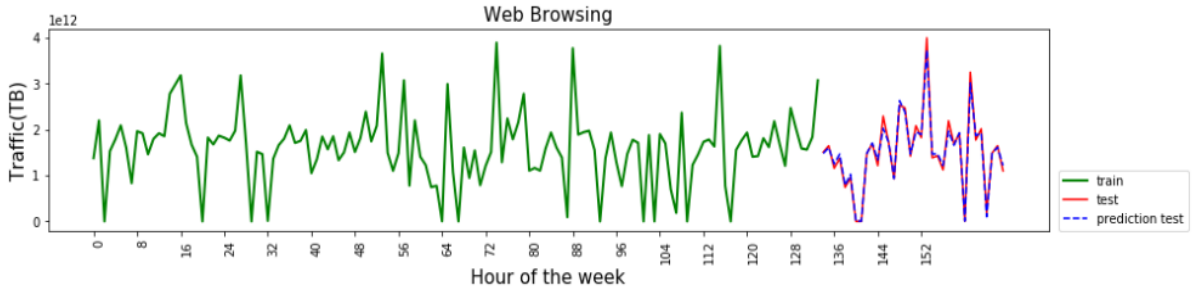

Figure 5.14: Prediction out put of Web browsing by RF Model

the MVR model, RF predicts a higher score value and fewer error (MSAE) values. Based on these performances, RF is recommended for the network traffic prediction task . The remaining cluster and APP category's results in both model are given in Table 5.4.

Table 5.4: Prediction Evaluation Result

cluster	Algorithms	App category score			App category RMSE		
		File Access	Streaming	Web browsing	File Access	Streaming	Web browsing
cluster 0	MVR	0.45	0.33	0.41	9.781	9.761	12.26
	RF	0.95	0.94	0.96	0.089	0.477	0.624
cluster 1	MVR	0.61	0.67	0.38	4.956	4.606	9.37
	RF	0.97	0.95	0.96	0.125	0.437	0.697
cluster 2	MVR	0.34	0.5	0.26	20.559	21.98	10.48
	RF	0.99	0.96	0.95	0.149	0.416	0.129
cluster 3	MVR	0.36	0.44	0.46	10.016	3.31	18.88
	RF	0.93	0.94	0.94	0.125	0.171	0.111
cluster 4	MVR	0.26	0.36	0.21	19.37	8.51	6.73
	RF	0.95	0.98	0.96	0.174	0.531	0.531

APP usage demand prediction for the Network providers aspect plays an important role in resource management, network planning, and optimization. By understanding the traffic usage patterns of APPs, network providers can benefit in allocating radio resources, implementing cache server, optimizing the network, etc. If a large number of APPs are targeted, their traffic volume and access time roughly have a linear correlation with their number of users. Accordingly, cellular providers can estimate and allocate radio resources based on the predict traffic demand to improve their user's experience by controlling call dropping and traffic congestion.

We observe that the top three popular APPs consume the majority of traffic. For example, the APP with the largest traffic volume is more than 50% of the total traffic volume of the APPs category, and the APP with the longest network access time(Higher APPs usage) takes of the total network access time of the APPs category. Understanding the usage patterns of these APPs, network providers may do certain optimizations case by case. The temporal patterns of the APPs category help network providers allocate radio resources. For example, the access time per IP flow helps network providers decide the timers in-state promotion.

In addition, users of certain social Streaming and Web browsing APPs categories are more likely to move around across several base stations. In the future, LTE networks will push the first IP hop forward to base stations, which increases the flexibility of content placement and optimization. However, if users frequently move around, the corresponding APP may increase the complexity to decide where to cache content and what content cache. Therefore understand the usage and popularity of APP is very important to handle those problems.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

The performance data traffic of ethio telecom shows that there is a significant amount of network congestion due to limited channel elements and, congestion is related to the data traffic increase. Furthermore, we have observed that, in both residential and business areas, congestion is the main capacity challenge. In residential areas, the peak traffic occurs at midnight time whereas in business areas the peak occurs between 10:00 and 16:00. These show the network capacity challenges in location and time.

To overcome the above resource challenges, we proposed APP clustering and prediction techniques using Random Forest model.To evaluate the techniques we used ethio telecom XDR data collected from Addis Ababa BSs sites. Clustering has been made by grouping BSs to their RNC. For APP prediction, we used Random Forest and Multi variable regression algorithms. The three popular APP chosen for predictions are Web-browsing, Streaming, and File Access APPs.

In our finding Web browsing APP category has the highest traffic usage all over the clusters and, it's popularity has been uniform throughout the clusters. This shows users' APPs usage points of interest in Addis Ababa city is the same.Because our population settlement has been mixed which means, each cluster's combination of different APP usage interests like, education, businesses, and residential.

For performance evaluation, we used MSAE and r^2 metrics.Prediction score of Web browsing, Streaming, and File Access is between 96% and 99% while the MSAE value

is between 0.450 and0.089. For the three popular APP that constitutes 53% of the total traffic, we get average prediction r^2 (score) of 97%, 95%, and 96%, respectively.

In this research, we have understood the APP usage behaviors and, predicted the top 3 popular APP over BSs based collected data. The purpose of popular APP prediction is to apply network planning optimization, and managing strategies of cellular networks to reduce transmission latency, congestion, call drooping and improve user experience. Our study provides basic insights with a method for network operators to understand the traffic consumption of APP and, design schemes for resource allocation depending on APP usage.

6.2 Recommendation

Understanding and predicting APP based on locations is for caching techniques that store content at the storage devices near the wireless network edge for the eventual use are good APPs to reduce back-haul traffic [7]. A vast portion of back-haul traffic is contributed by transmitting duplicate popular data to multiple users [36]. The duplicate data transfer is mainly caused by transmitting popular content (File Access, streaming, and Web-browser). For example in Table 5.2, 36.9% of total traffic demand is due to one APP that is Web-browser. When the general content is requested by a considerable number of users at various times, the wireless network needs to frequently send the same content to each user. As a result, the duplicate data passes through the back-haul links over and over again, which causes significant back-haul traffic.

If BSs can cache the general content in cache memories installed at the BSs they are capable to directly serve each user. Then, the back-haul links only need to communicate the general content once to the BSs rather than multiple times, which can effectively reduce the duplicate data typically transmitted. More precisely more attractively if can exactly predict the user requests and cache the predicted content in advance they can directly serve the mobile users without demanding the use of the back-haul link[37][38]. As a result, the BSs can provide simultaneous transfer with low back-haul needs, particularly during peak hours. In addition to that, the cost of installing memory storage is much lower compared with that of upgrading the back-haul capacity.

References

[1] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, no. Apr, pp. 1063–1095, 2012.

[2] J. Lempiainen and M. Manninen, *UMTS radio network planning, optimization and QoS management*. Springer, 2003.

[3] C. V. N. Index, "Global mobile data traffic forecast update, 2016–2021 white paper," *Cisco: San Jose, CA, USA*, 2017.

[4] K.-W. Lim, S. Secci, L. Tabourier, and B. Tebbani, "Characterizing and predicting mobile application usage," *Computer Communications*, vol. 95, pp. 82–94, 2016.

[5] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng, "Characterizing user behavior in mobile internet," *IEEE transactions on emerging topics in computing*, vol. 3, no. 1, pp. 95–106, 2014.

[6] M. Zeng, T.-H. Lin, M. Chen, H. Yan, J. Huang, J. Wu, and Y. Li, "Temporal-spatial mobile application usage understanding and popularity prediction for edge caching," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 36–42, 2018.

[7] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A feature-based bayesian method for content popularity prediction in edge-caching networks," *arXiv preprint arXiv:1905.09824*, 2019.

[8] C. Kappler, *UMTS networks and beyond*. John Wiley & Sons, 2009.

[9] J. Kataria and A. Bansal, "Exploration of gsm & umts security architecture with aka protocol," *International Journal of Scientific and Research*, vol. 3, no. 2, pp. 2250–3153, 2013.

[10] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and iot applications," in *2017 8th In-*

ternational Conference on Information, Intelligence, Systems & Applications (IISA). IEEE, 2017, pp. 1–8.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*. Springer, 2012, pp. 62–77.

[13] L. Chen, D. Yang, D. Zhang, C. Wang, J. Li *et al.*, "Deep mobile traffic forecast and complementary base station clustering for c-ran optimization," *Journal of Network and Computer Applications*, vol. 121, pp. 59–69, 2018.

[14] J. D. Olden, J. J. Lawler, and N. L. Poff, "Machine learning methods without tears: a primer for ecologists," *The Quarterly review of biology*, vol. 83, no. 2, pp. 171–193, 2008.

[15] J. M. Benítez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Transactions on neural networks*, vol. 8, no. 5, pp. 1156–1164, 1997.

[16] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, 2006.

[17] A. C. Muller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, 2017.

[18] Y. Zhao, K. Liu, X. Xu, H. Yang, and L. Huang, "Distributed dynamic cluster-head selection and clustering for massive iot access in 5g networks," *Applied Sciences*, vol. 9, no. 1, p. 132, 2019.

[19] C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 173–182.

[20] F. Willnecker, A. Brunnert, and H. Krcmar, "Model-based energy consumption prediction for mobile applications." in *EnviroInfo*, 2014, pp. 747–752.

[21] J. Wu, M. Zeng, X. Chen, Y. Li, and D. Jin, "Characterizing and predicting individual traffic usage of mobile application in cellular network," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 2018, pp. 852–861.

[22] A. Tongaonkar, S. Dai, A. Nucci, and D. Song, "Understanding mobile app usage patterns using in-app advertisements," in *International Conference on Passive and Active Network Measurement*. Springer, 2013, pp. 63–72.

[23] X. Liu, W. Ai, H. Li, J. Tang, G. Huang, F. Feng, and Q. Mei, "Deriving user preferences of mobile apps from their management activities," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, p. 39, 2017.

[24] J. Yan, Y. Qiao, J. Yang, and S. Gao, "Mining individual mobile user behavior on location and interests," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 1262–1269.

[25] Q. Chen, M. Zhang, and X. Zhao, "Analysing customer behaviour in mobile app usage," *Industrial Management & Data Systems*, vol. 117, no. 2, pp. 425–438, 2017.

[26] S. Jiang, B. Wei, T. Wang, Z. Zhao, and X. Zhang, "Big data enabled user behavior characteristics in mobile internet," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2017, pp. 1–5.

[27] K. Huang, C. Zhang, X. Ma, and G. Chen, "Predicting mobile application usage using contextual information," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 1059–1065.

[28] D. Yu, Y. Li, F. Xu, P. Zhang, and V. Kostakos, "Smartphone app usage prediction using points of interest," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, p. 174, 2018.

[29] Z.-X. Liao, Y.-C. Pan, W.-C. Peng, and P.-R. Lei, "On mining mobile apps usage behavior for predicting apps usage in smartphones," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 609–618.

[30] Y. Qiao, X. Zhao, J. Yang, and J. Liu, "Mobile big-data-driven rating framework: measuring the relationship between human mobility and app usage behavior," *IEEE Network*, vol. 30, no. 3, pp. 14–21, 2016.

[31] W. Sun, D. Miao, X. Qin, and G. Wei, "Characterizing user mobility from the view of 4g cellular network," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 1. IEEE, 2016, pp. 34–39.

[32] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.

[33] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[34] M. Čeh, M. Kilibarda, A. Lisec, and B. Bajat, "Estimating the performance of random forest versus multiple regression for predicting prices of the apartments," *ISPRS International Journal of Geo-Information*, vol. 7, no. 5, p. 168, 2018.

[35] M. M. RAMADHAN, I. S. SITANGGANG, F. R. NASUTION, and A. GHIFARI, "Parameter tuning in random forest based on grid search method for gender classification based on voice frequency," *DEStech Transactions on Computer Science and Engineering*, no. cece, 2017.

[36] J. Poderys, M. Artuso, C. M. O. Lensbøl, H. L. Christiansen, and J. Soler, "Caching at the mobile edge: A practical implementation," *Ieee Access*, vol. 6, pp. 8630–8637, 2018.

[37] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5g networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.

[38] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.