# Hotspot Prediction Using Deep Learning: In the Case of Addis Ababa LTE Network

---


By - Gebremedhin Weldemariam

Advisor - Sosina Mengistu (PhD)


A Thesis Submitted to


School of Electrical and Computer Engineering

Addis Ababa Institute of Technology

Addis Ababa University


In Partial Fulfillment of the Requirements for the Degree of Master of Science in Telecommunication Engineering

January 2022

**Addis Ababa University**

**Addis Ababa Institute of Technology**

**School of Electrical and Computer Engineering**

Signed by:

| | | |
|---|---|---|
| Sosina Mengistu (PhD) ________ | ____________ | ________ |
| Advisor | Signature | Date |
| ____________ | ____________ | ________ |
| Internal Examiner | Signature | Date |
| ____________ | ____________ | ________ |
| External Examiner | Signature | Date |

DECLARATION

I, the undersigned, declare that the thesis work is my own and all reference sources used in this thesis have been acknowledged.

Gebremedhin Weldemariam

Name Signature

# ABSTRACT

Due to users demand for high mobile bandwidth, operators increasingly spend money in routine optimization activities, adding cell sites, and deploying new technologies. This is mainly to improve capacity and quality of congested cells in network hotspot areas. In hotspots, Quality of Service (QoS) is highly degraded. Dynamically managing and optimizing network resources improves degraded QoS in real time and decreases capital and operational costs. For dynamic optimization, proactive and accurate identification of hotspot variations is required. Some studies have modeled hotspot prediction based on data traffic usage and others based on user density. However, this may not accurately identify congested cells since cells have different capacity configurations, and network QoS such as throughput are affected by factors such as radio conditions.

This study focuses on developing a model to accurately identify congested cells based on number of users and user throughput by utilizing cell counters collected from Long-Term Evolution (LTE) network. Data preprocessing techniques such as replacing missing values using per cell per hour historical mean, and resampling to reduce class imbalance are applied on the collected data. Long Short-Term Memory (LSTM), a deep recurrent neural network, is used to model hotspot prediction, and performance of the model is evaluated using metrics such as accuracy, precision, and F1 score.

Experimental results show the model performs with an accuracy of 89.13% and F1 score of 85.5%, and predicts for four future hours. Therefore, the model achieves acceptable performance, and it can help operators predict hotspots for dynamic optimization to improve QoS in real time.

*KEYWORDS* – Long-Term Evolution, Congested Cells, Quality of Service, Hotspot Prediction, Active Users Throughput, Long Short-Term Memory

# ACKNOWLEDGMENT

I would like to extend my deepest gratitude to my advisor, Sosina Mengistu (PhD), for her invaluable suggestions, guidance, advice, responsiveness, and readiness to support me during the thesis work.

I would like to thank Kifle Berhane, Abel Demsew, Taddele Teka and Zelalem Mengistu for providing me the required data, domain related information and recommendations, and specifically I am grateful to Kifle Berhane for his unreserved support.

Finally, I would like to express my deepest gratitude to my families particularly my brother, Gezae, and my special appreciation goes to my wife, Selam Mulaw and my son, Semawi.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| | |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| 4G | Fourth Generation |
| AI | Artificial Intelligence |
| ARIMA | Autoregressive Integrated Moving Average |
| CAPEX | Capital Expenditure |
| CDR | Call Detail Record |
| CNN | Convolutional Neural Networks |
| CSFB | Circuit Switched Fall Back |
| eNB | eNodeB |
| eNodeB | evolved Node B |
| EPC | Evolved Packet Core |
| E-UTRAN | Evolved Universal Terrestrial Radio Access Network |
| FDD | Frequency Division Duplex |
| GBR | Guaranteed Bit Rate |
| GRF | Gaussian Random Field |
| GSM | Global System for Mobile Communication |
| IP | Internet Protocol |
| KPI | Key Performance Indicator |
| LSTM | Long Short-Term Memory |

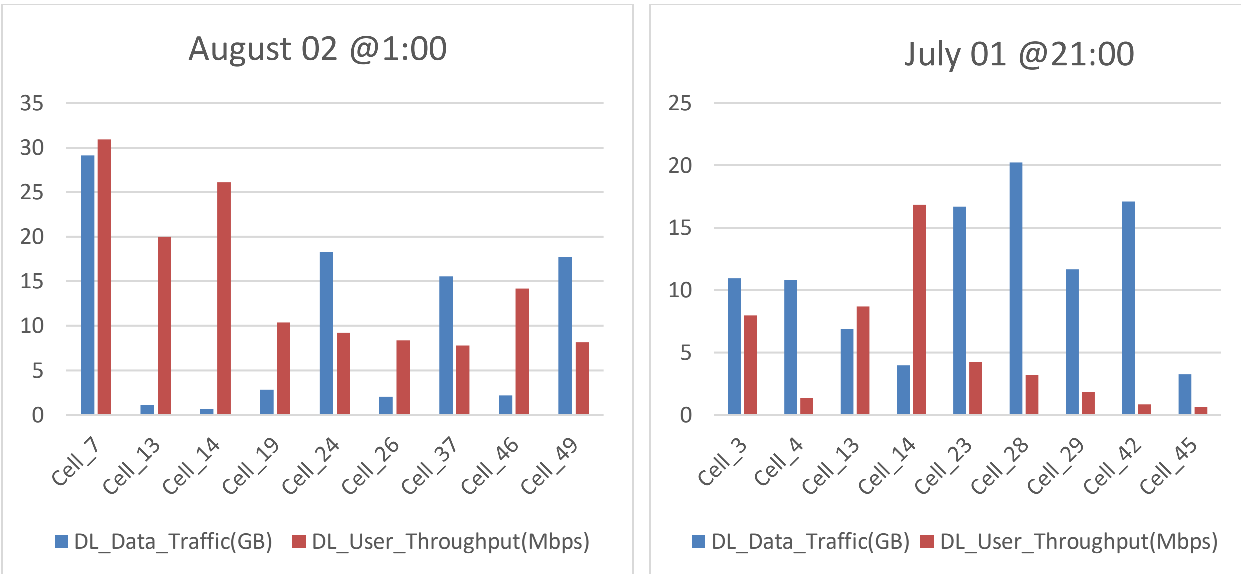
| | |
|---|---|
| LTE | Long Term Evolution |
| MAE | Mean Absolute Error |
| MDT | Minimization of Drive Test |
| ML | Machine Learning |
| MME | Mobility Management Entity |
| OFDM | Orthogonal Frequency Division Multiplexing |
| OPEX | Operational Expenditure |
| P-GW | Packet Data Network Gateway |
| PRB | Physical Resource Block |
| PRS | Performance Reporting System |
| QCI | QoS Class Identifier |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RMSE | Root Mean Square Error |
| RNN | Recurrent Neural Network |
| S-GW | Serving Gateway |
| SMAPE | Symmetric Mean Absolute Percentage Error |
| TDD | Time Division Duplex |
| UE | User Equipment |
| UMTS | Universal Mobile Communication System |

# 1 INTRODUCTION

Due to users demand for high mobile bandwidth and rapid increase of cellular networks traffic, operators are continuously investing in deploying new technologies and more cell sites to expand the capacity of their network [1]. They are also devoting more time and money to improve the quality of their network through various optimization approaches such as balancing cell loads in congested areas. Cell sites in these congested areas with high data traffic and number of users are considered as hotspots, and their users' service quality is compromised [2]. Identifying and handling these hotspots efficiently and in real time helps improve service quality, and decrease expenditures. However, due to users' dynamic mobility behavior, cellular network hotspots highly fluctuate, and currently existing methods and mechanisms to handle and identify these variations in real time is inflexible and inefficient [3]. This is resulting in increasing operators' capital expenditure (CAPEX) and operational expenditure (OPEX), and more importantly causing service quality degradation.

Unfair load distribution among neighboring cells is created due to fluctuation of hotspots. This imbalance can be efficiently handled using load balancing mechanisms by adjusting cell coverage or other optimization parameters of overloaded cells and their adjacent low traffic cells [4]. As a result, resources are utilized efficiently, and good service quality experience of users is maintained. To achieve this, hotspots need to be accurately and proactively identified.

To identify hotspots and their variations, they firstly need to be defined correctly and modeled based on the definition. The authors in [5] defined hotspots as areas in which user density is high. However, [6] did not define hotspots as cell coverage areas in which number of User Equipment (UE) is high but an area with high data volume usage. However, this definition is not guaranteed as high data traffic may not lead to degradation of network QoS such as throughput as shown in Figure 1.1 (a) and (b). Cell_7 has the highest data traffic demand but throughput is also the highest of all the cells in (a). Whereas, in (b), Cell_45 has the lowest data volume but throughput is very bad. In addition, Cell_3 and Cell_4 have almost equal data volume but the throughput at Cell_4 is bad while Cell_3 has very good throughput. This is because throughput is affected by a variety of factors such as radio condition, user mobility, cell load, number of transmit antennas, and user location [7], [8].




(a) (b)

Figure 1. 1- Data traffic vs user throughput (August 02 and July 01, 2021)

So, to identify hotspots and their variations for dynamic and proactive network optimization, they need to be measured differently and more correctly.

In LTE, throughput is one of the most important QoS indicators. When factors such as cell load, radio conditions, and user speed variations cause user throughput degradation, PRB utilization increases resulting in radio resource congestion. In hotspots, throughput is severely degraded due to cell load and according to [9] throughput decreases as number of active users increases as shown in Figure 1.2 below. Thus, hotspots should be monitored and modeled using number of users and user throughput to be correctly identified and handled.

Previously, some research works were conducted to capture traffic variations in Addis Ababa. In [10], data traffic was modeled for Universal Mobile Telecommunications System (UMTS) network. It modeled data traffic usage per partitioned area. Aggregating cell sites with different capacities and congestion levels in to a single area and modeling per area is not a good idea. Besides, modeling data volume variations may not correctly identify hotspot variations if the model is applied for dynamic optimization. So, it is better to identify hotspot variations per cell level rather than per grid area. It is also important to handle hotspot variations based on number of users and their throughput to identify them more correctly.

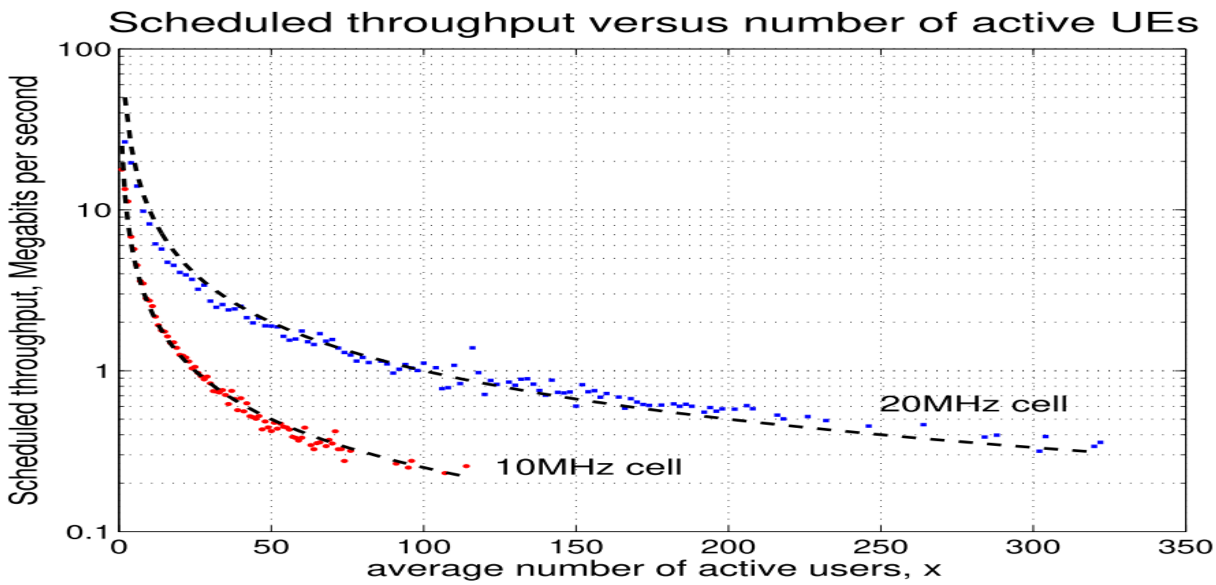



Figure 1. 2- User throughput versus number of active users[9]

This research work develops number of users and their throughput-based hotspot identification mechanism using state of the art deep recurrent neural network for Addis Ababa LTE network cells.

# 1.1 Statement of the Problem

Hotspots in cellular networks occur when cells get congested by users' traffic. Users served by these cells experience degraded QoS resulting in bad QoE. To improve degraded network QoS in hotspots by dynamically optimizing and managing the scarce radio resources, it requires proactive hotspot identification scheme to capture congestion variations.

Operators configure their cell sites with different capacities. Currently in Addis Ababa LTE network, for example, cell sites have different capacity resources configurations: 4T4R+2CC (four transmitters and four receivers with two component carrier's aggregation), 4T4R, 2T2R, and 1T1R. This implies that different cells can handle different number of users without affecting service quality such as throughput. Some cells can also generate more data traffic volume with better QoS than others. So, modeling congested cells based on UE density [5] or data usage volume [11] may not result in accurate identification of hotspots since cells are configured with different capacities, and no exact and direct indication of users in a cell experience bad service quality. Besides, network QoS such as throughput are affected by various factors such as cell user location and radio conditions. However, by taking number of active users and user throughput, hotspots can be more accurately identified.

Therefore, to accurately identify hotspots and efficiently handle them, active users and throughput-based hotspot variations prediction model is required. So, this thesis develops average active users and user throughput-based hotspot identification mechanism by capturing real cell counters and KPIs variations of Addis Ababa LTE network using state of the art deep recurrent neural network algorithm, Long Short-Term Memory.

## 1.2 Objective

### 1.2.1 General objective

To build number of users and throughput-based hotspot prediction model using deep learning to identify congestion variations of Addis Ababa LTE network cells.

### 1.2.2 Specific objectives

The specific objectives of the research are:

- To analyze and build dataset required for deep learning scheme using preprocessing techniques
- To develop hotspot prediction model using deep learning scheme
- To evaluate prediction performance of the employed model

## 1.3 Literature Review

In cellular network hotspots, QoS is highly degraded resulting in bad QoE. Identifying hotspots occurrences and variations proactively and in real time is crucial for dynamic radio resources optimization to keep good service quality experience of users unaffected. So, different literatures related to mobile network hotspot prediction are reviewed here.

Data usage volume variations were captured in [3] to predict the spatiotemporal hotspot variations. data usage volume of base stations was collected in hourly basis for thirty days. Gaussian Random Field (GRF)-based model was employed to capture spatial behavior of data traffic, and Holt-winters model was utilized to predict the data traffic variations temporally. Symmetric Mean Absolute Percentage Error (SMAPE) metric was adopted to evaluate hotspot prediction performance.

Data usage volumes were also used in [6] to model the spatial distribution of hotspots. To determine the location of the mobile data, a combination of different geo-location techniques such as signal delay-based, and pathloss and network configurations-based estimations were applied. It was conducted using data traffic of specific busy hours. The framework did not model real temporal data traffic variations.

In [11], cells were defined and labeled as hotspots or low traffic based on data traffic volume. Cell KPIs were collected and utilized. Support Vector Machine (SVM) algorithm was applied to identify a cell as either hotspot or low traffic using the collected KPIs variations. Evaluation metrics such as accuracy were used to evaluate the performance of the employed classifier algorithm.

However, hotspot identification in [12] was developed by determining the existence of group of users. It collected and used Minimization of Drive Test (MDT) to collect UE coordinates. The paper utilized a clustering algorithm, K-means, to model hotspot identification but user mobility behavior was not considered.

In [5] historical data of Call Detail Record (CDR) numbers reflecting number of users were collected, and hotspots were identified by predicting UE density distributions. Different deep learning and non-deep learning algorithms: autoregressive integrated moving average (ARIMA), LSTM, and ConvLSTM were applied, and their hotspot prediction performance was evaluated using Root Mean Square Error (RMSE) metric.

A summary of the papers is presented in Table 1.1 to summarize emphasis of the studies, dataset used to model the problems, algorithms employed for the problems, experimental results obtained, and limitations of the reviewed studies.

Table 1. 1- Summary of literature review

| Paper | Focus | Dataset | Algorithm | Result | Limitation |
|---|---|---|---|---|---|
| [3] | Predicting hotspots based on data volume usage | Data traffic of base stations for 30 days | - GRF<br>- Holt-winters | SMPE=25. 1% average for every hour | Data usage volume may not correctly identify hotspots as cells are configured with different capacities and QoS such as throughput are affected by radio conditions, cell load |
| [12] | Determining group of clusters based on number of UEs | UEs coordinates from minimization of drive test (MDT) | K-means | Half hotspots are close to 40m | - Users' mobility not considered.<br>- User density may not correctly identify hotspots as QoS are affected by a variety of factors such as radio condition, user cell location, user mobility |
| [11] | Identifying hotspots based on data volume usage | Cell KPIs – hourly basis during a day | SVM | Accuracy: 98.1% | Data usage volume may not correctly identify hotspots as cells are configured with different capacities and QoS such as throughput are affected by radio conditions, cell load |

| | | | | | |
|---|---|---|---|---|---|
| [6] | Determining and tracking hotspots based on data traffic locations | Data traffic of 15 busy hours of 4 days | Framework using variety of techniques such as geolocation of data traffic | Hotspot densities: 4 per km2 for five times mean traffic and 2.5 for ten times mean traffic | - Real temporal data variations were not modeled<br>- Data usage volume may not correctly identify hotspots |
| [5] | Predicting spatiotemporal hotspots based on UE density | Voice & data CDR of active users for two months in hourly basis | ARIMA, ConvLSTM, LSTM | LSTM: 75% of grids within 30 RMSE | User density may not correctly identify hotspots |

To conclude, the reviewed research works help us shape our study objectives and techniques by enabling us to know the methodologies and algorithms of hotspot prediction problems. Various research works used different hotspot identification measurement parameters and algorithms to model hotspot prediction. Some research papers utilized different models based on data traffic volumes to model cellular networks congestion variations. Some have also used number of users distribution to identify hotspots. However, while cell sites are configured with different capacity resources and network QoS such as throughput are also affected by radio conditions, user location, and user mobility, more correct modeling of congested LTE network cells identification needs to be based on number of users and user throughput measurements. Since it is based on users and their throughput, it can be directly identified if a user is experiencing bad service quality. So, average active users and user throughput-based hotspot identification of Addis Ababa LTE network is modeled.

## 1.4 Scope and Limitation

The study focuses on providing cell level hotspot variations identification model for Addis Ababa LTE network which only supports data service. Only data features measurements are considered since voice service is not supported in the current Addis Ababa LTE network. The study is conducted to model hotspot prediction for the downlink.

## 1.5 Contribution

The contribution of our study is providing number of users and their throughput-based hotspot variations identification model that can accurately identify congested cells to be used as input for dynamic optimization to improve QoS and QoE in real time.

Our research also contributes sequence to sequence predictive classification in hotspot identification. It predicts the class of cells as congested or low traffic over four future hours.

Our model can also be used as input during network expansion planning.

## 1.6 Thesis Organization

The rest of the paper is organized as follows. Overview of LTE network and its architecture is described in Chapter 2. LTE quality of services and existing hotspot identification mechanisms in Ethio telecom are also discussed in this chapter. Chapter 3 introduces overview of deep learning. Here, deep learning schemes that can be used in hotspot predictions are explained. The methodologies followed in this research work such as data collection, data preprocessing, and model performance evaluation are explained in Chapter 4. In Chapter 5, experimental setup is described and results are summarized. Finally, conclusions and recommendations of this study are presented in Chapter 6.

# 2 LONG-TERM EVOLUTION NETWORK

## 2.1 LTE Overview

LTE is a Fourth Generation (4G) technology, and it is marketed as 4G. LTE is considered to be evolved from Third Generation (3G) although their air interfaces are not the same. The key requirements of air interface specifications that led to the design of LTE are higher data speed, spectral efficiency, lower latency, and support of multiple bandwidths [13]. LTE band supports different bandwidths ranging from 1.4MHz, the minimum to 20MHz, the maximum. It has higher data rate and lower latency than 3G. In addition, it supports backward compatibility with its predecessor generations: 3G, and Second Generation (2G). LTE provides, in the downlink, up to 100Mbps theoretical data rate for Time Division Duplex (TDD) and 150Mbps for Frequency Division Duplex (FDD). In FDD duplexing method, downlink and uplink use different frequencies. Whereas, in TDD, downlink and uplink use the same frequency. LTE does have also a later release called LTE-Advanced (LTE-A), but it is vital to remember that LTE and LTE-A are the same technology, and the name does not make LTE-A a distinct system from LTE, nor is it the ultimate evolutionary stage in the evolution of LTE [14]. LTE-A includes additional features such as carrier aggregation that enhance LTE data rate.

In Ethio telecom, the bandwidth of the deployed LTE network is 20MHz and the duplexing method is FDD. The LTE network in Ethio telecom has both LTE and LTE-A and here we use LTE to refer for both LTE and LTE-A. Since LTE is all-IP network, it doesn’t support the circuit switched voice service. Ethio telecom provides voice service for 4G subscribers via 3G by Circuit Switched Fall Back (CSFB) mechanism.

## 2.2 LTE Network Architecture

3rd Generation Partnership Project (3GPP) developed LTE architecture is an all-IP network with

Orthogonal Frequency Division Multiplexing (OFDM) air interface [15]. As shown from Figure 2.1, LTE has two main parts: the access part, Evolved Universal Terrestrial Radio Access Network (E-UTRAN), and the core part, Evolved Packet Core (EPC). LTE has simplified architecture compared to UMTS and Global System for Mobile Communication (GSM). There is no radio controller network element in the access part, and the core network part has no circuit switched core, and it is all IP network [16].

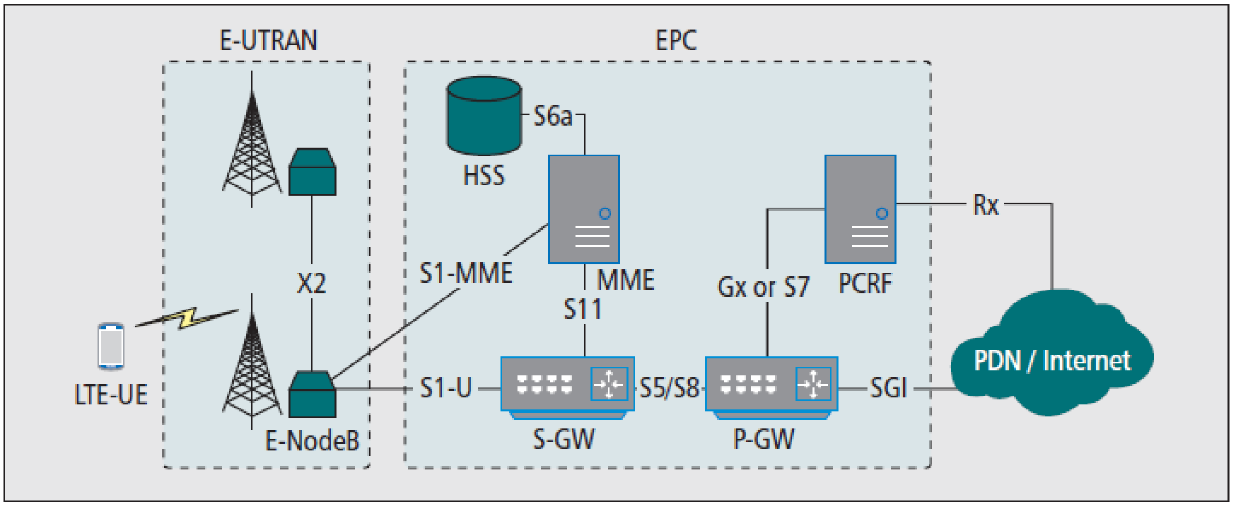



Figure 2. 1- LTE architecture [15]

## 2.2.1 LTE E-UTRAN

E-UTRAN is the radio access part of LTE network. It is a network of eNodeB (eNBs), which support OFDMA and enhanced antenna methods, and each eNB has an IP address which makes it part of the all-IP LTE network [16]. Since there is no radio controller network element in LTE E-UTRAN, its functions such as handover are handled by eNBs.

## 2.2.2 LTE EPC

EPC is the core part of the LTE network and it is IP-based. In the core part there is no circuit switched core. EPC is comprised of core elements such as Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Data Network Gateway (P-GW). MME handles the control plane i.e., mobility signaling for the access network. S-GW and P-GW handle the user plane i.e., user data traffic transportation. The EPC is connected to the access part of the network via S-GW and to the cloud via P-GW.

### 2.2.3 E-UTRAN interfaces

- *LTE Uu* - This is the air interface that connects UE and eNBs to create communication link between them.
- *X2* – an interface that connects eNBs to each other.
- *S1* – this is the interface that creates communication links between E-UTRAN and EPC. S1-U connects eNBs with S-GW and S1-MME connects eNBs with MME.

## 2.3 Quality of Service in LTE

Network QoS can be used to measure service quality experience of a user. Different services require different QoS. For example, real time gaming service requires higher throughput than web surfing service. LTE was designed to handle these varied and increased requirements by providing higher data rates in real time.

When a user connects to the LTE network and requests for a service, LTE instantly assigns QoS Class Identifier (QCI) according to QoS requirement of the service. This is implemented between the UE and P-GW on the set of bearers: Radio bearer, S1 bearer, and S5/S8 bearer, which are all referred to as the EPS bearer [17] as shown in Figure 2.2.

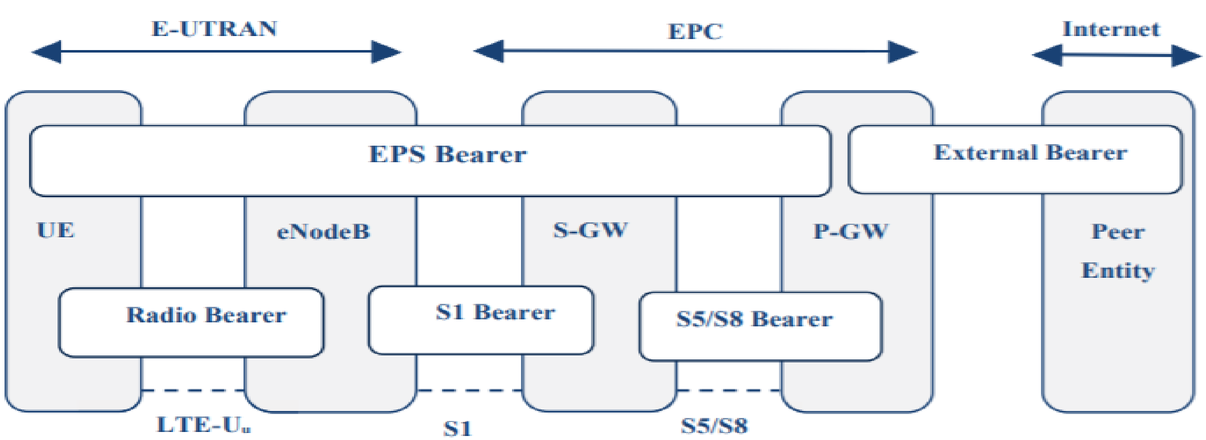



Figure 2. 2- EPS bearer service architecture where LTE QoS is applied [18]

According to [19], LTE has two types of bearers: Guaranteed Bit Rate (GBR) and non-Guaranteed Bit Rate (non-GBR) in which non-GBR is based on best effort but GBR is assigned with dedicated resources. The bearers are also named as default and dedicated bearers. Default bearer is always non-GBR but dedicated bearer can be either GBR or non-GBR.

### 2.3.1 LTE QCI

QCI is an identifier to node specific parameters that handle packet transmission based on resource type, priority, packet delay, and packet loss, and a service is assigned with only one QCI [20]. The LTE standardized QCI characteristics are provided in table 2.1.

Table 2. 1- LTE QCI Characteristics [20]

| QCI | Resource Type | Priority Level | Packet Delay | Packet Error Loss Rate | Example Services |
|---|---|---|---|---|---|
| 1 | GBR | 2 | 100ms | $10^{-2}$ | Conversational voice |
| 2 | | 4 | 150ms | $10^{-3}$ | Conversational video (live streaming) |
| 3 | | 3 | 50ms | $10^{-3}$ | Online gaming (real time) |
| 4 | | 5 | 300ms | $10^{-6}$ | Video streaming (buffered) |
| 5 | non-GBR | 1 | 100ms | $10^{-6}$ | IMS signaling |
| 6 | | 6 | 300ms | $10^{-6}$ | Video, Web, FTP … |
| 7 | | 7 | 100ms | $10^{-3}$ | Voice, Video streaming (live), Interactive game |
| 8 | | 8 | 300ms | $10^{-6}$ | Video streaming (buffered), Web, FTP ... |
| 9 | | 9 | | | |

## 2.4 Handling QoS in Hotspots

Users served by congested cells experience degraded QoS. It is a very challenging task to meet the required QoS in these hotspot cells. One of the reasons is that hotspots are not proactively and correctly identified. Ethio telecom identifies congested cells by conducting regular drive tests, by extracting and analyzing cell KPIs, and from customer complaints. It is very difficult to handle QoS degradations in hotspots using these manual identification mechanisms. So, a model is required to efficiently identify hotspots to tackle QoS degradations in real time.

# 3 DEEP LEARNING SCHEMES

## 3.1 Deep Learning Overview

Machine learning (ML) is a subfield of artificial intelligence (AI). It allows a computer system to learn from data by utilizing a range of algorithms to understand patterns and make predictions by iteratively learning from the data rather than user programmed instructions [21]. To solve a real-world problem using ML, first and foremost, data for the problem in interest must be available. Then an algorithm suitable to the type of problem is selected. The data is fed to the algorithm, and the algorithm learns important patterns and information. Using this trained model, ML prediction is done on new data. This is how ML works.

Deep learning is a subclass of ML that enables computer systems to learn from experience and comprehend the world as a multi-layer of concepts in which complicated concepts are learned from building less complicated concepts [22]. The relationship among AI, ML, and deep learning is depicted in Figure 3.1.

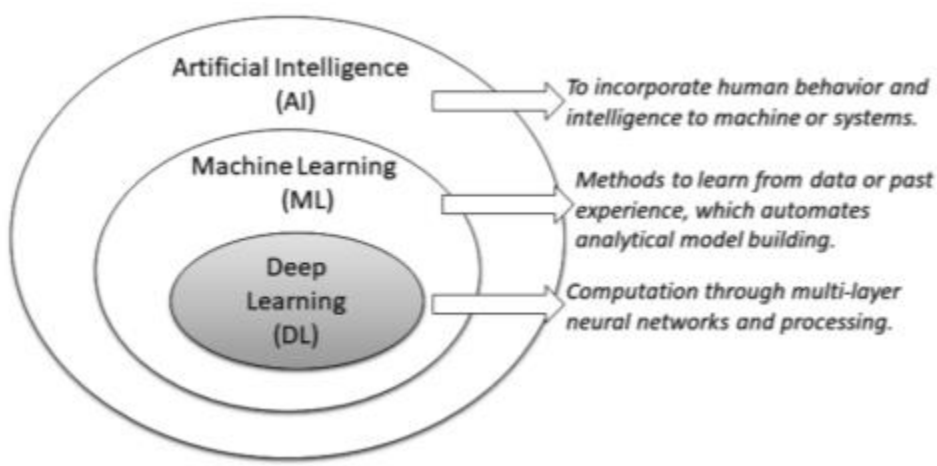



Figure 3. 1- Artificial intelligence types and their relationship [23]

Digital data are enormously increasing due to technological advances, smart phone users, continuous growth of Internet users, and social media [24]. This continuous increase of data is one of the factors that enabled deep learning to gain popularity because the performance of deep learning schemes is better than traditional ML models with big data [23]. The performance trend of deep learning schemes and traditional ML techniques is shown in Figure 3.2. The figure demonstrates that as amount of data keeps increasing, performance of deep learning algorithms is largely surpassing common ML ones. Traditional ML algorithms perform better with small amount of data as Figure 3.2 shows.

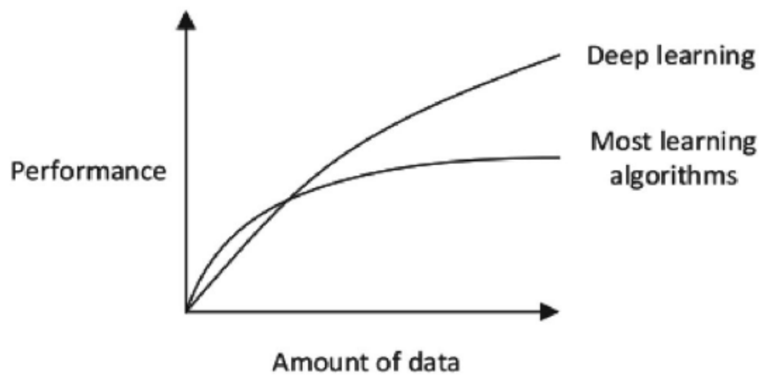



Figure 3. 2- Performance comparison of deep and machine learning with data amount [23]

## 3.2 Neural Networks

The principle of deep learning is based on the working principle of biological neural networks. It takes the concept of brain neurons and their networks to mimic learning capability of humans to solve complex and non-linear prediction problems using computer systems. Neural networks are designed in a layered fashion by grouping neurons into different layers. Fundamentally, neural networks can have three layers: input, hidden, and output layer. However, in the hidden part of the neural network, there may exist one, two or more layers as shown in Figure 3.3. In the input layer, input values, in our case cell feature values, are fed to the neural network algorithm. The hidden layer is in charge of performing mathematical calculations on the input layer's input values. Activation functions on weighted inputs are used to generate the hidden layer's output. The output layer produces the model's output, which is derived from the hidden layer's output.

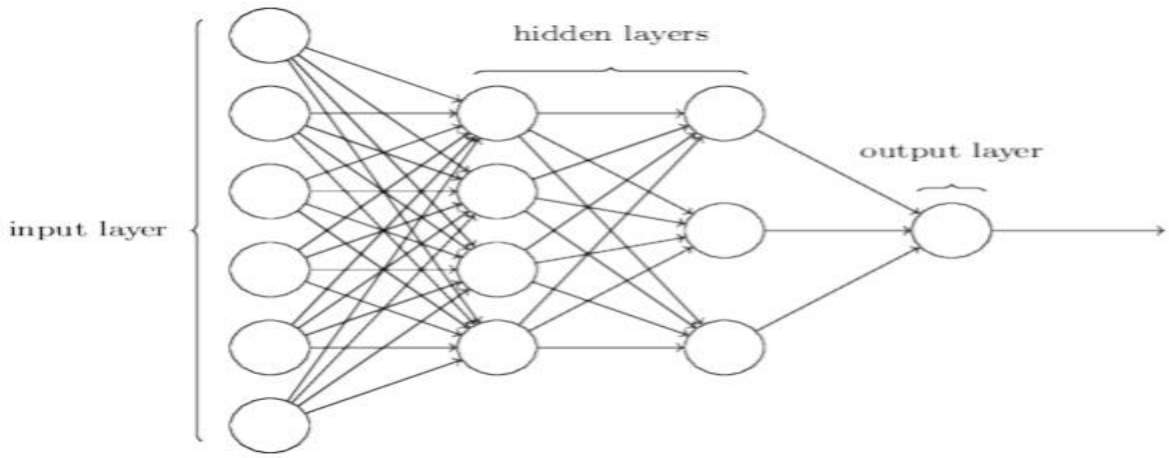



Figure 3. 3- Layers of Neural Networks [25]

There are different types of neural networks. To mention the most common and more important ones are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNN is commonly applied for image processing such as medical imaging, and RNN is commonly used to model sequential data such as time series problems [26].

# 3.3 Recurrent Neural Networks for Hotspot Prediction

RNNs are deep neural networks created to solve real world problems using sequential data. Sequential data are ordered series such as statements in a language created by arranged words based on established rules and time series data ordered based on events in time. In modeling sequential data, future values can be determined based on past observations. Normal deep neural networks cannot model these kinds of problems because it is not clear how a standard deep neural network could use prior observations to determine subsequent ones [27]. However, RNNs have memory and are capable of remembering the past and can predict the future based on past information and current input data. The architecture of an RNN is presented in Figure 3.4, and *A* represents to a neural network. It takes current input at time *t* and previous time step output at time *t-1* to generate output for future time step at *t+1*.

Our study is to build hotspot prediction model utilizing time series LTE cell counters and KPIs. Besides, it is sequence to sequence problem. It takes sequence of past hourly observations and generates outputs for sequence of future hours. This kind of problem is modeled using deep recurrent neural networks.

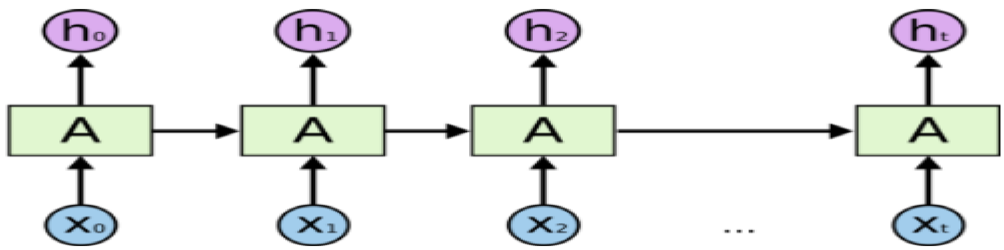
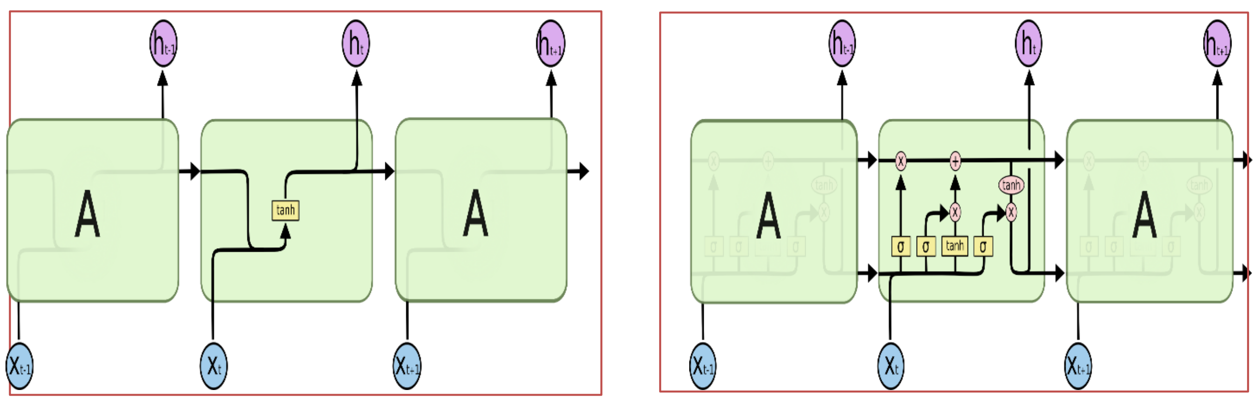



Figure 3. 4- Basic architecture of RNN [27]

## 3.3.1 Long Short-Term Memory

There are different variants of deep recurrent neural networks such as normal RNN, and LSTM. According to [28], LSTM handles long term temporal dependencies of data but normal RNNs face difficulties. As a result, normal RNNs suffer from vanishing gradient problem. This problem occurs when weight is updated to minimize prediction error during backpropagation for long sequences. When number of layers increases, the difference between old weight and new updated weight becomes very small and model no longer trains. This is due to simple and single layer design of normal RNN cell; however, LSTM solves this problem by adding important functionalities and layers in the LSTM cell: cell state, forget gate, input gate, and output gate [27].




(a) Normal RNN (b) LSTM

Figure 3. 5- Comparison of normal RNN and LSTM cells architecture [27]

As shown from Figure 3.5, normal RNN cell has only one tanh layer; whereas, LSTM cell has three sigmoid layers including the tanh in the normal RNN cell.

In LSTM, capturing long term dependencies is achieved with the help of cell state, and the information in the cell state is managed and determined by three gates: forget gate, input gate, and output gate. Forget gate determines which information should be omitted or kept for next time step in the cell state. The input gate determines what new information should be added to the cell state. Then next, the output gate determines what information from the cell state is to be generated as output.

The overall interactions in the LSTM cell are given by the equations stated below.

$$f_t = \sigma(W_f \, . [h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i \, . [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\hat{C}_t = \tanh(W_C \, . [h_{t-1}, x_t] + b_C) \tag{3.3}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t) \tag{3.4}$$

$$o_t = \sigma(W_o \, . [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t * \tanh(C_t) \tag{3.6}$$

The symbols *W* and *b* refer to weight and bias respectively for each respective gate. $\sigma$ refers to sigmoid activation function and *tanh* refers to tanh function. $f_t$, $i_t$, $\hat{C}_t$, $C_t$, $o_t$, and $h_t$ refer to forget gate, input gate, new candidate information to be added to cell state, cell state, output gate and hidden state respectively.

# 4 METHODOLOGY

In this chapter of the study, the detailed methodology followed to achieve the stated objectives is presented. This research work begins with collecting, studying, and investigating relevant papers, books, and other electronic documents related to hotspot identification mechanisms to help us identify and shape the research objectives and techniques.

## 4.1 System Model

Below in Figure 4.1, the system model required to build hotspot prediction is shown.

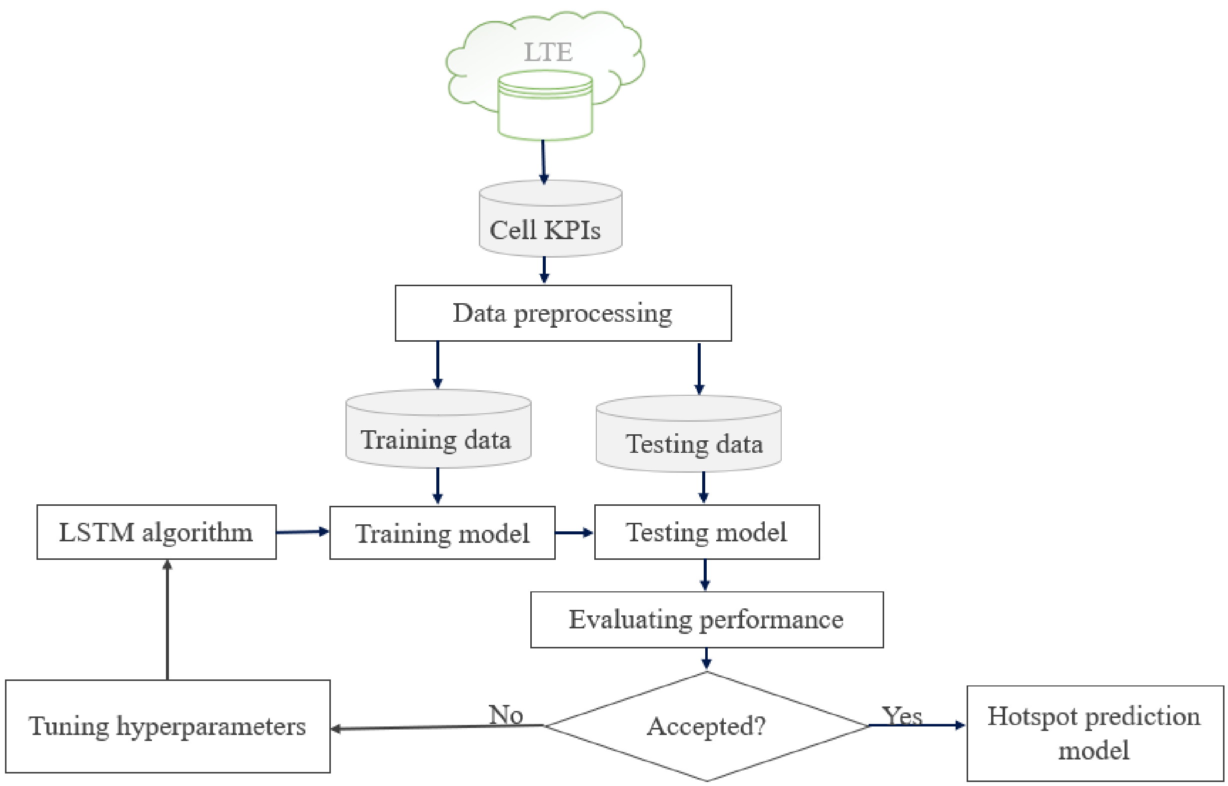



Figure 4. 1- System model

The system processes to develop hotspot prediction model are:

1. First LTE cells KPIs and counters are collected.
2. Next the collected dataset is preprocessed. As a result, training and testing datasets are generated.
3. Using the training set, the LSTM algorithm is trained to develop hotspot identification model.
4. Then using the testing dataset, the trained model is tested.
5. Next the performance of the developed model is evaluated using classification evaluation metrics.
6. If the model meets the required performance criteria, final hotspot prediction model is produced. Unless, the algorithm is optimized by tuning hyperparameters then passes through training, testing, and evaluating steps until the model meets the performance evaluation criteria.

## 4.2 Data Collection

The model is built to identify congestion variations in cell level. So, Addis Ababa LTE cells' counters and KPIs are collected from Ethio telecom Performance Reporting System (PRS). The minimum time dimension in PRS is one hour according to the domain experts. So, cell dataset is collected in hourly basis. Since deep learning algorithms require large amount of data to learn well during training, the data is collected from 200 cells for the duration of nearly two months from July 01, 2021 to August 23, 2021. Since LTE cell sites are layered to 3G sites, the naming of cells contains Radio Node Controller (RNC). So, the cells are selected from areas under RNC4 and RNC5.

## 4.3 Data Preprocessing

### 4.3.1 Data exploration

Firstly, getting familiarized with the characteristics of the collected data is important to proceed with data preprocessing. The dataset has some duplications, counters with all-time zero (0) values, and non-number features. Missing values are also observed in the dataset. Data are missing when there are no recorded data points for a feature or counter in the observation of interest [29] and missing values are caused by various factors such as environmental factors, measurement sensor failure, and infrastructure failures [30]. Among the infrastructure failures, in our case, are transmission failure, eNB device failure and so on. The data values have also varied values ranging from 0 to 2230694.

Time series data are generally affected by five components [31] and one of them is seasonality. Seasonality characteristics exists when sequential data demonstrates consistent and predictable patterns per time intervals. From Figure 4.2, it is observed that features such as downlink average active users and user throughput exhibit seasonality per 24 hours (daily). This can be used to identify easily predictable past observation time steps to enhance model learning skill.

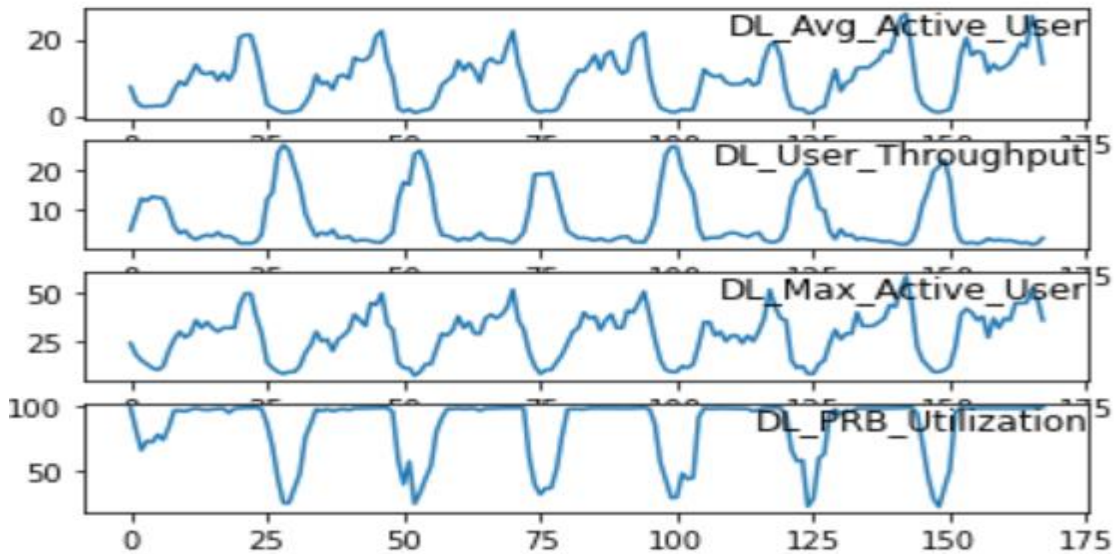



Figure 4. 2- Plot of one week data records of some cell features

### 4.3.2 Data cleaning

Models require correct and quality data to enhance their training skill and accuracy. As mentioned in section 4.3.1 above, the collected data have duplications. So, the duplicated ones are just dropped. Those irrelevant numeric and non-numeric features, and features with all-time zero records are also removed. There are uplink measurements in the collected cell level dataset. However, this study focuses on modeling the downlink congestion variations. As a result, uplink measurements are removed from the dataset. After cleaning the data, 55 counters and KPIs left from 97 collected cell features.

### 4.3.3 Feature selection

Even though, feature selection is not a requirement to be met for deep learning algorithms, there is accuracy and computational time improvement if only relevant features are utilized for deep neural networks [32].

For this and other simplicity reasons, independent features having good correlations with dependent features are selected. The dependent features are the target ones, and they are average active users and user throughput. Note that this is only to select independent features correlated with the target features using cross correlation. It is not about minimizing redundancy between independent features. Figure 4.3 shows sample features correlation. DL_PRB_Utilization, for example, has good correlation with average active user and throughput. It is positively correlated with average active user and negatively correlated with user throughput. Whereas, TA(0-78m)_Percent has almost no correlation with the targets, and it is discarded. After correlation is done with the targets, 31 features are selected from the 55 features. Please note that features are abbreviated and renamed for simplicity and clarity reasons.

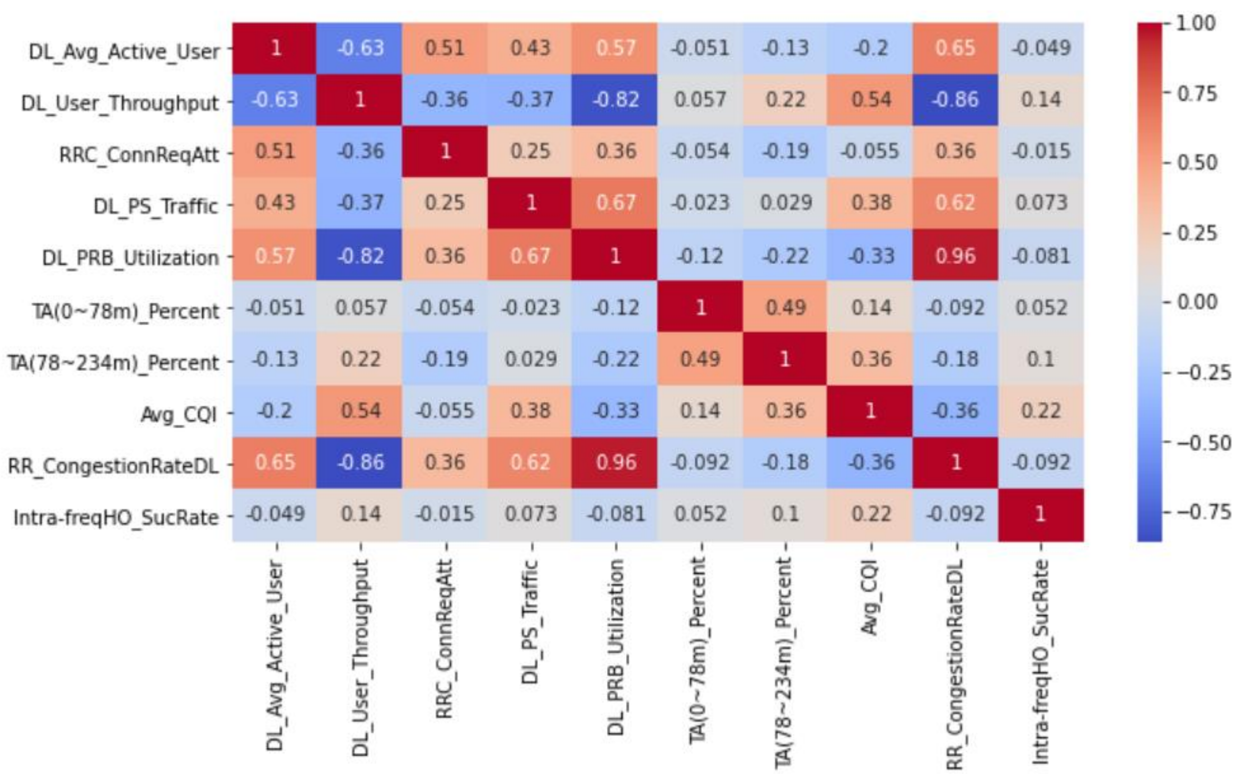


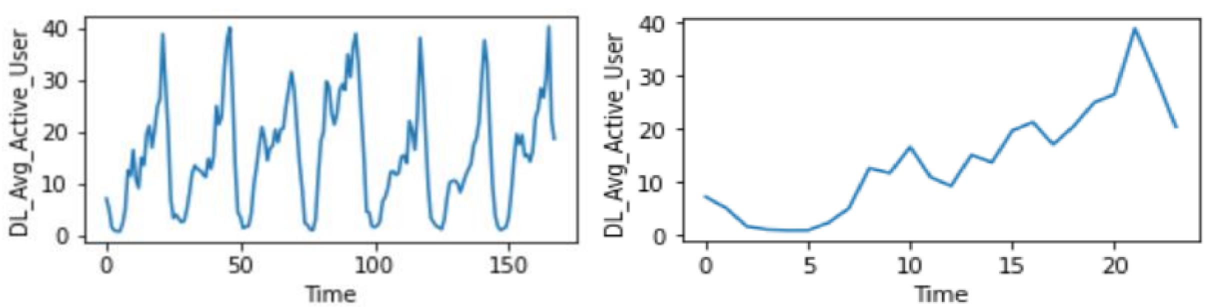

Figure 4. 3- Features correlation heatmap for sample features

### 4.3.4 Handling missing values

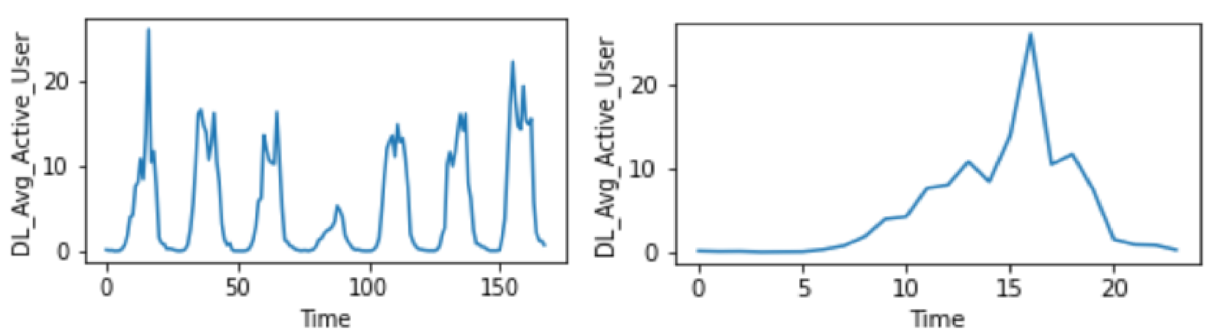
As described above in section 4.3.1, missing values are identified from the collected dataset. In addition, the features exhibit seasonality per 24 hours' time interval.

Cells have their own unique characteristics. As shown in Figure 4.4 below, although both cells exhibit seasonality per every 24 hours, their measurement values are different at different hours. For example, Cell_11 serves relatively high number of users from 18:00 to 23:00 O'clock in the evening. This site may be in residence area. Besides, the maximum average active users served by Cell_11 ranges between 30 and 40 users. Whereas, Cell_14 serves high number of users, most of the time, from 14:00 to 18:00 O'clock in the afternoon. This site may be in business area. The maximum average active users served by Cell_14 ranges mostly between 10 and 20 users.

Therefore, handling of missing data is done based on the characteristics of the cells. As a result, per cell and per hour historical mean is imputed to replace missing values.




(a) One week's active user plot for Cell_11 (b) The first day's active user plot for Cell_11




(c) One week's active user plot for Cell_14 (d) The first day's active user plot for Cell_14

Figure 4. 4- Downlink average active users' behavior comparison between two cells

## 4.3.5 Labeling

Since this study is a time series predictive classification, cells are labeled with 1 for hotspot and 0 for low traffic cells for each observation. Labeling is done based on hotspot threshold shown in Table 4.1 below, which is taken from internal document of Ethio telecom and recommended by domain experts. By determining target user throughput based on baseline speed for video buffering, number of active users is calculated using mathematical formula according to the Ethio telecom and Huawei document. If a cell, at a point of observation, has greater than 12 average active users and less than 4Mbps average throughput, it is a hotspot cell. Otherwise, it is a non-congested cell. In addition to this hotspot threshold, there is also another threshold for normal cells. However, it is not considered in this study for two reasons; all cells are not categorized as normal or high value cells, and there is huge data imbalance between hotspot and low traffic cells.

Table 4. 1- LTE cells hotspot threshold

| User Throughput (Target) | Active User | Hotspot |
|---|---|---|
| 4Mbps | 12 | Active users > 12 and user throughput < 4Mbps |

Source: Ethio telecom (2021)

## 4.3.6 Data imbalance resampling

Imbalanced data in classification is a scenario in which a dataset has much less number of observations of minority class than the majority class [33]. In this scenario, observations from the minority class are misclassified since they don't have enough information for the classifier model. This causes bad classification performance of a model.

In our case, there is huge difference between the hotspot and low traffic classes of the collected data points. The dataset is collected from 200 cells in hourly basis for 54 days. So, the dataset has 259,200 observations. From these data points, only 35665 instances are in the hotspot class i.e., the hotspot cells are 13.76% of the total data points. This imbalance may be caused due to underutilization of the LTE network resource. It is recalled that, expansion project for Addis Ababa LTE network has been done recently.

According to [34], under-sampling which is removing some data points from the majority class or over sampling which is adding data points to the minority class can overcome data imbalance problem. In our study, under-sampling is done on the majority class to reduce class imbalance.

Cells having much smaller number of hotspot class observations are dropped. Since our study is time series classification problem, removing data points randomly can affect the sequential characteristics of our dataset. So, the reduction of the low traffic class, which is major class, is based on removing less congested cells over the 54 days in hourly basis observations of a cell. As a result, our final dataset after resampling of data imbalance is shown in Table 4.2 below. According to [35], the majority of recent works on class imbalance focus on imbalance ratios in the range from 1:4 to 1:100 i.e., from 25% to 1%. So, we can proceed with developing our model having 32.32% of class imbalance as it is not a severe imbalance.

Table 4. 2- Final dataset after resampling

| Data Points | Minor Class Percentage |
|---|---|
| 77,600 | 32.32% |

### 4.3.7 Data scaling

Neural networks become stable and their performance increases during model training if the input features are scaled to small ranges specifically between 0 and 1. This is due to the fact that neural networks use small weights and these small weights are adjusted by gradient descent optimization algorithms to reduce model estimation errors during training. So, if there are features range differences, gradient descent doesn't smoothly converge towards local minimum because learning rate is not updated at the same rate for all features. Model training process becomes slow and unstable.

As mentioned in section 4.3.1 above, the collected data values range from 0 to 2230694. So, data scaling is done by applying data normalization. Normalization is scaling data values to the range between 0 and 1 using the formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

where, x is the value of each feature to be scaled. $x_{min}$ is the minimum and $x_{max}$ is the maximum values.

### 4.3.8 Dataset splitting

To build a model with accepted accuracy performance, it needs to be trained with training dataset and then tested with unseen testing dataset. The collected data is split into 80% for training and 20% for testing sets according to common defaults as shown in the table below.

Table 4. 3- Training and testing sets

| Data Points | Training Set | Testing Set |
|---|---|---|
| 77,760 | 62,208 | 15552 |

### 4.3.9 Dataset reshaping

In this study, we apply LSTM algorithm to model our hotspot prediction problem. LSTM is a deep recurrent neural network in which its input layer takes three dimensions (3D) as input: samples, time steps, and features. However, the collected data has only two dimensions (2D): rows as samples and columns as features. So, the 2D data is reshaped into 3D.

## 4.4 Algorithm

In this study, LSTM algorithm is used to model hotspot prediction. LSTM is an enhanced and special form of RNNs used to handle sequential data. One of the complex problems where LSTM is suitable to apply is time series prediction because LSTM is capable of learning long term dependencies by overcoming vanishing and exploding gradient problems [36]. In addition, this study focuses on modeling sequence to sequence hotspot prediction. Taking sequences of past hourly observations, future sequences of hours are predicted as hotspot or low traffic cells. RNNs are designed to handle for such sequence to sequence problems. So, sequence to sequence LSTM is an ideal solution for this study. The model is developed using Keras on Tensorflow [37] using Python programming language.

## 4.5 Performance Evaluation Metrics

The metrics that are used to measure the performance of the developed hotspot prediction model are explored here. There are different types of metrics available to evaluate model performance. But, the most important classification metrics are identified and described here.

## 4.5.1 Confusion matrix

A confusion matrix contains information about a classification model's actual and expected classifications and the data in the matrix is used to evaluate the performance of the model [38]. It is a matrix used to display false positive (type I) and false negative (type II) errors. Using the values in the matrix, the most important classification metrics: accuracy, precision, recall, and F1 score are derived. Table 4.4 is the confusion matrix.

Table 4. 4- Confusion matrix

| | | Predicted Value | |
|---|---|---|---|
| | | Negative (0) | Positive (1) |
| Actual Value | Negative (0) | True Negative (TN) | False Positive (FP) |
| | Positive (1) | False Negative (FN) | True Positive (TP) |

## 4.5.2 Accuracy

Accuracy is one of the metrics used to measure performance of a classification model. It is the ratio of correct predictions to total predictions. Derived from the confusion metrics, the formula to find accuracy is given by:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \tag{4.2}$$

## 4.5.3 Precision

Precision determines the fraction of actual positives out of the total predicted positives. Here it measures the accuracy of the hotspot cells. Its values range from 0 to 1 where 0 shows no precision and 1 shows perfect precision. The formula of precision derived from the confusion matrix is given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{4.3}$$

### 4.5.4 Recall

Recall measures the fraction of positive predictions out of the total actual positives. Its values range from 0 to 1 where 0 shows no recall and 1 shows perfect recall. The formula of recall derived from the confusion matrix is given by:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{4.4}$$

### 4.5.5 F1 score

F1 score combines precision and recall into one score. It considers both FP and FN errors of the classification model. It is very important metric in imbalanced data classifications. In this study F1 score is considered to be the most important evaluation metric because there is imbalance between hotspot and low traffic cells' class data points. The formula of F1 score is given by:

$$\text{F1 score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \tag{4.5}$$

# 5 EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, experimental results of the hotspot identification model are analyzed and discussed. First, experiment setup and model tuning are described. Next, experimental results are summarized then the effect of data imbalance on model prediction performance is demonstrated.

## 5.1 Experiment Setup

Taking sequence of past hourly observations, our model predicts cells as hotspots or non-hotspots for sequence of hours in the future. Cell counters show patterns of seasonality per every 24-hours. So, the input time step is configured to be 24-hours. The output time step is decided to be for four hours because some decrease of prediction accuracy is observed if time step is extended beyond four hours. As a result, the model is trained using 15,264 sequences.

To implement this, sequence to sequence LSTM is employed i.e., encoder decoder LSTM. The encoder encodes the inputs of past sequence of observations i.e., the 24-hour data points into vector representation, and LSTM decoder decodes the representation and generates all sequence outputs. This is fed to fully connected dense layer to interpret the decoder outputs for each timestep i.e., the four future hours. Then final output is generated from the output layer.

To check model skill while tuning hyperparameters, 10% of the training data is used for validation. In addition, to calculate prediction errors, cross entropy function is used since the study is a classification problem.

## 5.2 Model Optimization

To update LSTM network weights, Adam optimizer algorithm is used. Adam is practically better than other gradient descent algorithms [39]. Different hyperparameters such as number of nodes, learning rate, and batch-size are tuned to obtain the optimized classification model. Finally learning rate is chosen to be 0.0001 and the parameters of the sequence to sequence LSTM hotspot prediction model are listed in Table 5.1.

Table 5. 1- Parameters of hotspot prediction LSTM model

| Parameter | Final Value | Range of Values |
|---|---|---|
| Encoder neurons | 32 | 20, 32, 48, 64 |
| Decoder neurons | 32 | 20, 32, 48, 64 |
| Fully connected dense layer neurons | 32 | 20, 32, 48, 64 |
| Dropout | 0.2 | |
| Hidden layer activation function | Default | |
| Output layer activation function | Sigmoid | |
| Optimizer | Adam | |
| Learning rate | 0.0001 | 0.1, 0.01, 0.001, 0.0001 |
| Input timestep | 24 | |
| Output timestep | 4 | |
| Epochs | 120 | 40, 60, 80, 100, 120 |
| Batch-size | 64 | 32, 64, 128 |

## 5.3 Experimental Prediction Results

After a trained model is built, it is tested using a real world and never seen before test datasets to evaluate its performance using evaluation metrics. During testing the model, 3816 sequences of datasets are used, and different metrics such as accuracy, precision, recall and F1 score are employed to evaluate the performance. The model predicts with 89.13% accuracy as shown in Figure 5.1. Accuracy is a crucial metric to measure performance of a classification model. But it is not enough metrics in case there is class imbalance. In our study, hotspot and low traffic classes are not equal in number of instances. As a result, there is data imbalance and F1 score is considered to be the most important metric to evaluate in such a case. The hotspot prediction model performs great with 85.5% F1 score. So, the experimental finding shows that the hotspot prediction model performs very good according to the accuracy and F1 scores.

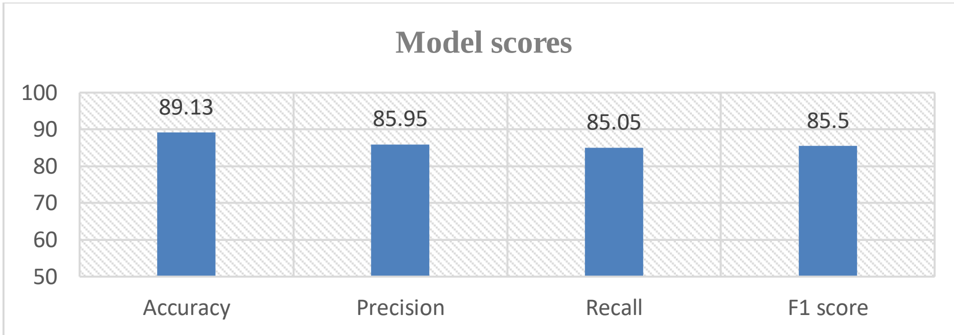
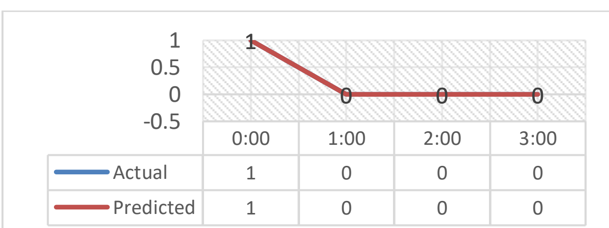
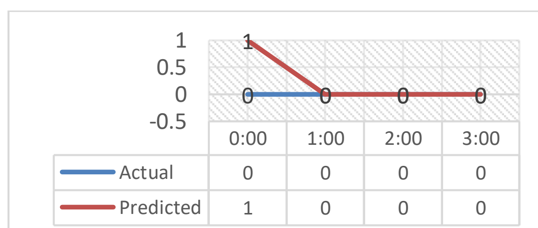



Figure 5. 1- Model performance scores

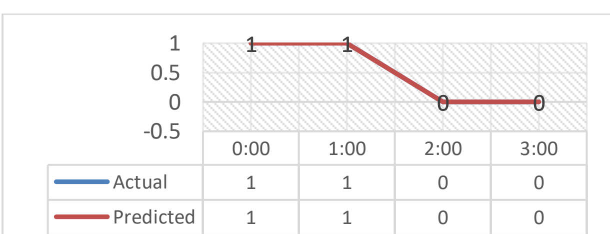
In Figure 5.2, prediction result samples are presented. Actual and prediction values variations over a given four hours for one cell are demonstrated. In Figure 5.2 (a), the cell is congested in the first hour of the sample sequence. In the remaining three hours the cell is not a hotspot one. The actual and predicted values are the same and the graph shows they are overlapped. The sequence here is correctly predicted. The next day for the same sequence of hours, the cell is actually not congested as shown in Figure 5.2 (b). But the first hour of the sequence is wrongly predicted. Figure 5.2 (c) shows the cell is hotspot cell in the first and second hours of the sequence and the model predicts here correctly. As we can see from the figures 5.2 (a) to 5.2 (d), hotspot variations of a cell are observed and the model identifies the hotspot variations with very good performance.



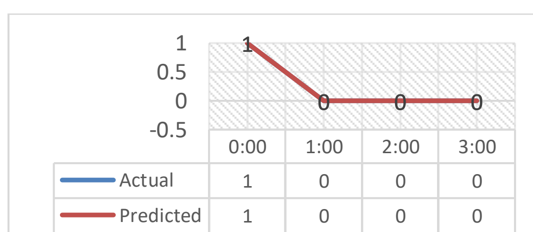

(a)




(b)




(c)




(d)

Figure 5. 2- Actual and predicted values variations over a given four hours for four days

## 5.4 Effect of Data Imbalance on the Model

In classification problems, class imbalance is a critical issue because it affects model performance. During learning process, models learn mostly the major classes. This is because they don't obtain enough information from the minor classes.

In this study, experimental results show that the hotspot identification LSTM model is sensitive to class imbalance. The hotspot prediction model performs well when the percentage of hotspot class is 32.32%, in our final model case, as shown above in Figure 5.1. However, when the hotspot class percentage decreases from 32.32% to 23.11%, the model performance is affected provided that number of training sequences and parameters configuration are the same. For 23.11% of the hotspot cells percentage, the accuracy is 94.09% and F1 score is 74.81%. As shown in Figure 5.3, the increase in accuracy and the decrease in F1 score indicate the class imbalance is affecting the performance of the hotspot prediction model.

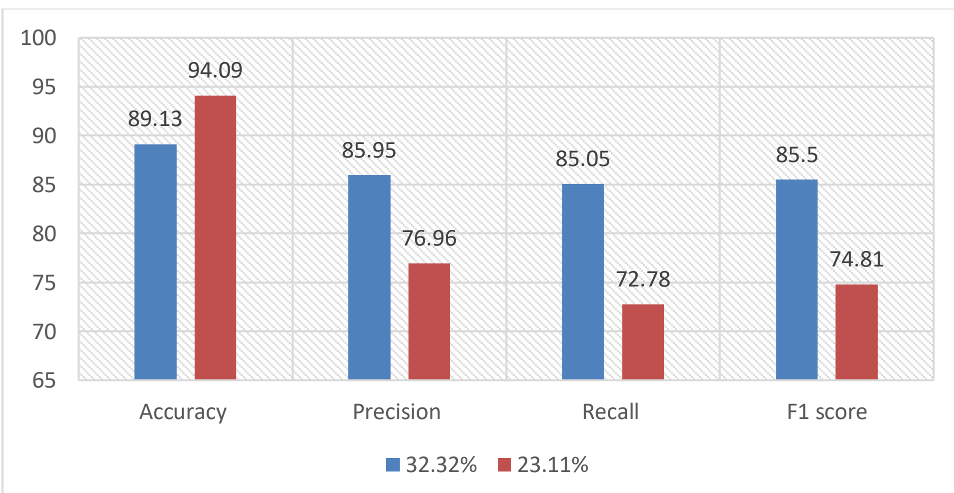



Figure 5. 3- Data imbalance impact on hotspot prediction LSTM model

# 6 CONCLUSION AND RECOMMENDATION

## 6.1 Conclusion

In cellular network hotspots, QoS is highly degraded. To improve network QoS in these congested areas, operators routinely perform optimization actions. But it is challenging task to efficiently identify hotspots because they are volatile and depend on mobility behavior of users. To proactively identify them, some studies modeled hotspots based on data traffic volume variations and others based on user density distributions. However, this may not identify hotspots correctly as cell sites are configured with different capacities and QoS such as throughput are affected by radio conditions, cell user location, and other factors. This study proposes number of users and user throughput-based hotspot identification mechanism using deep neural network LSTM algorithm.

Taking sequence of twenty-four hours past observations, the model is trained to predict cells as hotspots or non-hotspots for sequence of future four hours. After training the model, it is validated using testing data and the model performs with an accuracy of 89.13% and F1 score of 85.5%.

Experimentation is conducted to show the impact of class imbalance problem on the model. The collected data has huge data imbalance between hotspot and low traffic cells. Resampling technique is used to minimize the imbalance problem but still exists class imbalance. The percentage of the minority class in our final model data is 32.32%. The effect of class imbalance is examined by decreasing the percentage of the minority class to 23.11%. According to the experiment result, F1 score decreases from 85.5% to 74.81%. This shows class imbalance has significant impact on the LSTM model.

To conclude, this study delivers number of users and user throughput-based cell level hotspot prediction model with accepted performance. If implemented, operators can benefit from our model by using it as input for network optimization processes to improve QoS in real time, to efficiently utilize scarce resources such as radio resources and energy, and to reduce capital and operational costs.

## 6.2 Recommendation

This study develops a model for LTE downlink hotspot prediction as network performance is mainly determined by downlink users. Future works are recommended to include uplink KPI measurements to model hotspot variations in the uplink so that hotspot identification of cells is handled in both the downlink and uplink of the network.

The experimental result in this study exhibits that the LSTM model is sensitive to class imbalance. Therefore, using balanced class datasets with increased data points could help improve the performance of the model.

# BIBLIOGRAPHY

[1] GSMA, “GSMA-Mobile Spectrum,” no. June. 2015, [Online]. Available: https://gsmaintelligence.com/analysis/2014/1/4g-driving-dta-usage-but-not-all-markets-reaping-the-rewards/412/.

[2] A. Cao, Y. Qiao, K. Sun, H. Zhang, and J. Yang, “Network traffic analysis and prediction of Hotspot in cellular network,” *Proc. 2018 6th IEEE Int. Conf. Netw. Infrastruct. Digit. Content, IC-NIDC 2018*, no. 1, pp. 452–456, 2018.

[3] X. Zhang, F. Zhou, J. Ning, P. Yu, and W. Li, “Hotspot localization and prediction in wireless cellular networks via spatial traffic fitting,” *IEEE/IFIP Netw. Oper. Manag. Symp. Cogn. Manag. a Cyber World, NOMS 2018*, vol. 1, pp. 1–6, 2018.

[4] T. Warabino, S. Kaneko, S. Nanba, and Y. Kishi, “Advanced load balancing in LTE/LTE-A cellular network,” *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, pp. 530–535, 2012.

[5] Y. Liu, X. Wang, G. Boudreau, A. Bin Sediq, and H. Abou-Zeid, “Deep Learning Based Hotspot Prediction and Beam Management for Adaptive Virtual Small Cell in 5G Networks,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 1, pp. 83–94, 2020.

[6] H. Klessig, V. Suryaprakash, O. Blume, A. Fehske, and G. Fettweis, “A framework enabling spatial analysis of mobile traffic hot spots,” *IEEE Wirel. Commun. Lett.*, vol. 3, no. 5, pp. 537–540, 2014.

[7] D. X. Zuo, “A Deep Learning Approach to Downlink User Throughput Prediction in Cellular Networks,” 2020.

[8] J. Salo, M. Nur-Alam, and K. Chang, “Practical Introduction to LTE Radio Planning,” *Eur. Commun. Eng. Ltd, Espoo, Finl.*, pp. 1–20, 2010, [Online]. Available: http://www.eceltd.com/lte_rf_wp_02Nov2010.pdf.

[9] J. Salo and B. Eduardo Zacarías, “Analysis of LTE radio load and user throughput,” *Int. J. Comput. Networks Commun.*, vol. 9, no. 6, pp. 33–45, 2017.

[10] Y. Abera and D. Hailemariam, “Spatio-temporal Mobile Data Traffic Modeling Using Fourier Transform Techniques,” *9th Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Powered by Smart Intell. ICTC 2018*, no. October 2018, pp. 20–24, 2018.

[11] L. Zhou and X. Chen, “SVM Hotspot Identification for Cellular Networks,” *2019 IEEE 5th Int. Conf. Comput. Commun. ICCC 2019*, pp. 1103–1107, 2019.

[12] T. V. Santana, A. Galindo-Serrano, B. Sayrac, and S. M. Lopez, “Dynamic network configuration: Hotspot identification for Virtual Small Cells,” *Proc. Int. Symp. Wirel. Commun. Syst.*, vol. 2016-Octob, pp. 49–53, 2016.

[13] C. Cox, *An Introduction to LTE*. Wiley, 2014.

[14] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*. 2014.

[15] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, “Cellular software defined networking: A framework,” *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 36–43, 2015.

[16] Tektronix communications, “LTE Networks : Evolution and Technology Overview,” *Evolution (N. Y).*, pp. 1–10, 2010.

[17] BEC TECHNOLOGIES, “Quality of Service (QoS) in LTE.” [Online]. Available: https://bectechnologies.net/wp-content/uploads/2019/12/QoS.pdf.

[18] F. Furqan, “Quality of Service ( QoS ) in 4G Wireless Networks Fatima Furqan Supervised by,” 2015.

[19] M. Alasti, B. Neekzad, J. Hui, and R. Vannithamby, “Quality of service in WiMAX and LTE networks,” *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 104–111, 2010.

[20] ETSI, “GSM, UMTS, LTE Policy and Charging Control Architechture (3GPP TS 23.203 version 16.2.0 Release 16).” 2014.

[21] J. Hurwitz and D. Kirsch, *Machine Learning for Dummies*. John Wiley & Sons, Inc., 2018.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. 2016.

[23] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–20, 2021.

[24] Y. Riahi and S. Riahi, “Big Data and Big Data Analytics: concepts, types and technologies,” *Int. J. Res. Eng.*, vol. 5, no. 9, pp. 524–528, 2018.

[25] Michael A. Nielsen, *Neural Networks and Deep Learning*. 2015.

[26] S. Patel and A. Patel, “Deep Leaning Architectures and its Applications A Survey,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 6, pp. 1177–1183, 2018.

[27] C. Olah, “Understanding LSTM Networks,” 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[28] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-Temporal Wireless Traffic Prediction with Recurrent Neural Network,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 4, pp. 554–557, 2018.

[29] H. Kang, “The prevention and handling of the missing data,” *Korean J. Anesthesiol.*, vol. 64, no. 5, pp. 402–406, 2013.

[30] A. Dubey and A. Rasool, “Clustering-Based Hybrid Approach for Multivariate Missing Data Imputation,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 710–714, 2020.

[31] R. Adhikari and R. . Agrawal, “An Introductory Study on Time Series Modeling and Forecasting Ratnadip Adhikari R. K. Agrawal,” *arXiv Prepr. arXiv1302.6613*, vol. 1302.6613, pp. 1–68, 2013.

[32] N. Promoteur and P. Facult, “University of Liege Feature selection with deep neural networks,” 2017.

[33] P. Gulati, “Hybrid Resampling Technique to Tackle the Imbalanced Classification Problem,” 2020.

[34] M. M. Rahman and D. N. Davis, “Addressing the Class Imbalance Problem in Medical Datasets,” *Int. J. Mach. Learn. Comput.*, no. April 2013, pp. 224–228, 2013.

[35] J. Brownlee, “A Gentle Introduction to Imbalanced Classification,” 2019. https://machinelearningmastery.com/what-is-imbalanced-classification/.

[36] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, 2020.

[37] F. Chollet, *Deep Learning with Python*. Manning Publications Co., 2018.

[38] A. K. Santra and C. J. Christy, “Genetic Algorithm and Confusion Matrix for Document Clustering,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 322–328, 2012.

[39] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.