A comparative study for Time Series Forecasting within software 5G networks

Pousali Chakraborty  
Fraunhofer FOKUS Institute  
Kaiserin-Augusta-Allee 31, 10589,  
Berlin, Germany  
pousali.chakraborty@fokus.fraunhofer.de

Marius Corici  
Fraunhofer FOKUS Institute  
Kaiserin-Augusta-Allee 31, 10589,  
Berlin, Germany  
marius-iulian.corici@fokus.fraunhofer.de

Thomas Magedanz  
Fraunhofer FOKUS Institute  
Kaiserin-Augusta-Allee 31, 10589,  
Berlin, Germany  
thomas.magedanz@fokus.fraunhofer.de

Abstract—5G has a very flexible network architecture due to virtualization and will come with various customisations based on different use cases. 5G also promises to provide intelligent networks with high bandwidth and low latency. One of the tradeoffs for this is the complexity of network monitoring and resource management of 5G; making availability, reliability and performance a challenge. The adoption of Software Defined Networking (SDN) and Network Function Virtualization (NFV) concepts ensure availability of network data and flexibility in architectural decisions for 5G. Because of the availability of data and advanced computing capabilities usage of ML (Machine Learning)/Artificial Intelligence (AI) can be envisaged in the control and management of 5G networks by predicting the load on the network. This article proposes a solution to integrate time-series based predictive analytics with 5G Core and shows a comparative study between two Time Series Forecasting Models—AutoRegressive Integrated Moving Average (ARIMA) and Facebook Prophet. Fraunhofer FOKUS Open5GCore is used as the reference 5G testbed toolkit for validating the proposal.

Index Terms—Machine Learning, Time Series Forecasting, 3GPP 5G Core, Open5GCore Toolkit, Failure Prediction.

I. INTRODUCTION

ML enables the capability for a system in scrutinizing data and gaining knowledge out of it. ML works beyond simple learning to improve knowledge over time and identifying hidden patterns in the data. ML can work on disparate data, which is an important aspect for forecasting. As it can learn, the accuracy of prediction gets increased over time. Time Series Forecasting is an essential field of ML. It can be seen as a regression problem, where one can analyze current and historical facts to find out future events. Time series modeling is a wide research area and various time series forecasting models have evolved over time like AutoRegressive Moving Average (ARMA), ARIMA, Holt-Winters.

In next-generation networks, the major challenge will be network management due to the complexity. In regard to this context, ITU Telecommunication Standardization Sector (ITU-T) is working on updating recommendations regarding Quality of Service (QoS) and users’ Quality of Experience (QoE). Integrating ML with networks is one of the major recommendations for better network management. ITU-T Y.3172 specifies an architectural framework for ML in future networks which includes ML pipeline, management and orchestration functionalities [2].

5G has applications in different fields like smart home, smart cities, industrial automation etc. For different types of usages, different levels of functionality, performance and reliability will be needed and these requirements will be served through network slicing. Each type of usage can have different Radio Access Technologies (RAT) with a wide range of Network Functions (NF) based on the required latency and bandwidth. The use of SDN, NFV and network slicing concept will enhance the flexibility of the 5G core network. On the other hand, it will add complexity in network management and that will be increasing with different use cases. Use of ML with automation will be beneficial to control the network.

This article proposes a solution to integrate ML-based analytics with 5G Core to enhance the reliability and performance of the core network. In order to use ML with 5G Core, time-series data will be collected from the 5G core, which will be mostly the operation statistics from 5G core and resource usage of the 5G core components. The Fraunhofer FOKUS Open5GCore testbed toolkit ([13],[14]) will be used as the base for the solution. Fig. 1, shows the outline of the proposal on integrating analytics with the platform. The monitoring module from the platform will be used to collect and send data from the core network to Prometheus. By introducing the time series based predictive analytics in the 5G platform, forecasting of the network data will be performed. In this article two different forecasting models, ARIMA and FB Prophet will be used.

![Fig. 1. The components for time-series based analytics in 5G network](image-url)

The remainder of the paper is constructed as follows, Section II provides the background for the proposed solution, where Section III defines the design concept. Section IV discusses the implementation of the solution on top of Fraunhofer FOKUS Open5GCore and Section V is used to evaluate the solution and Section VI concludes the paper.
II. BACKGROUND

The performance of a network depends directly on the customer experience. Better service can be provided to the customers if there is visibility in the network traffic information and analytics applied with the traffic to have better insights. There are many types of research going on for applying analytics in cellular networks. In the paper [4], the authors proposed a network traffic feature analysis method using multiple time series data mining. An SDN Framework for Distributed Network Analytics (DNA) [9] is proposed by Alexander Clemm et al. for the orchestration of network analytics across the network, so that users are able to interact with the network as a whole. CellIQ [10] proposed by Anand Padmanabha Iyer et al. is a real-time cellular network that supports analysis tasks. The use of data analytics in 5G networks has emerged as a hot topic for the researchers to improve the performance of 5G networks. In the paper [11] a complete framework for network access is presented for resource management and traffic steering challenges in 5G.

In mobile networks, monitoring the network and ensuring availability and reliability is a crucial part of network management. The complexity in the 5G network is much higher than the previous generation of networks and it has a wide area of use, which makes the management more challenging. Because of the complexity of the environment and high operation load on the core network, there is a high risk of failure at the 5G core side, which needs more attention for improving performance of the system.

The 5G core within the 5G architecture consists of network functions like Authentication Server Function (AUSF), Access and Mobility Management Function (AMF), Session Management Function (SMF), Unified Data Management (UDM), User Plane Function (UPF) etc. [1]. The User Equipment (UE) and (Radio) Access Network ((R)AN) are not part of the core. Because of user authorizations and faster services to devices, lots of traffic will be created in the core network.

A. 5G Core

5G Core is the implementation of new 5G components evolved from the Evolved Packet Core (EPC). The main 5G core components related to the article (shown in Fig. 2 along with the interfaces), are discussed below.

![Fig. 2. 5G Core Components](image)

AMF The Access and Mobility Management Function receives all connection and session related information from the UE, but it handles only connection and mobility management tasks for the UEs.

SMF Session Management Function is an important part of 5G service-based architecture. It interacts with the decoupled data plane, by managing all the Protocol Data Unit (PDU) sessions related functionalities like establishment, modifications, release in association with UPF [1]. A sequence of NG tunnels in 5G core and multiple radio bearers on the radio interface, together makes a PDU session.

AUSF Authentication Server Function works as the front-end for the UDM to execute the authentication properly.

UPF User Plane Functions are responsible for data plane functionalities of 5G system [1].

UDM Unified Data Management in 5G has a front end, which is AUSF and has a User Data Repository. UDR stores subscriber information and also policy profiles for PCF.

PCF Policy Control Function (PCF) enforces policy rules for control plane functions which include mobility and roaming management, network slicing.

Currently, different monitoring tools are used by the operators to do continuous monitoring within a mobile network, where ongoing network events and resource usages are monitored to figure out any vulnerabilities, anomalies or threats within the network. It requires the integration of core network with intelligent monitoring system, that helps in maintaining network resources more efficiently. There are already many monitoring tools available like Zabbix, SolarWinds, Prometheus with Grafana for better visualisation, which can be configured with the system to collect network traffic load and they are able to provide metrics for network resource utilization like CPU load, memory usage, disk space consumption. These monitoring tools can be used for monitoring services running on clouds also.

Most important tasks of the monitoring systems are evaluating the network status by collecting statistics from the network traffic, gathering traffic flow samples at different time intervals and also checking the response of the system for any kind of events triggered by the core or the radio layer (RAN or UE). For a network like 5G which itself has a complex architecture and will be generating a massive amount of data within the network, using only monitoring system to capture network statistics will be very challenging and that will cause much more overhead and delay in the whole procedure. Integrating time-series based predictive analytics with the monitoring system can be a good solution to address this problem.

Apart from the advancements in ML methods, the availability of data is the major factor for ML getting applied in different fields of research. Time series modeling is used to collect time-series data, which is a sequential set of data points \( x(t) \) collected over successive times (where time periods \( t = 0, 1, 2, \ldots, n \)) and study past observations rigorously. Thus, an appropriate model can be developed to describe the structure of the data and generate future values for the series. Using Time Series Forecasting, traffic can be analysed to...
predict the network load beforehand and that can help the network administrators to manage the network more effectively, which will result into improved reliability and availability of the system. There are already different ML approaches available for forecasting like – ARIMA, Recursive Neural Network (RNN), Seasonal ARIMA (SARIMA), FB Prophet; but selecting the best fit for the training data is a challenge.

III. CONCEPT

This article proposes a concept of using Time Series based analytics with 5G Core as shown in Fig. 3. The solution is designed to handle the issue of 5G core network failures and improve system reliability. The design is mostly based on the traffic generated within the control plane of the 5G core. The testbed that has been used as the base is Open5GCore, which is designed and implemented by Fraunhofer Fokus.

The whole idea revolves around traffic load on the 5G core. A Benchmarking Tool has been designed as a component where multiple UEs and RAN have been emulated, for generating the traffic in the 5G core. The captured traffic log is then sent to the monitoring tool Prometheus with the help of the monitoring module that is already present in Open5GCore, from there the data is stored in the Database (we have used InfluxDB here). The main advantage of using Prometheus over other monitoring tool is operational simplicity, and Prometheus is really easy to configure in any system. The stored data is then processed in batches and ML algorithm is executed on the datasets, to predict the future time steps from the time series data.

![Fig. 3. Design concept](image)

The analytics module, which is responsible for the time series forecasting is designed considering four important metrics. These metrics are directly related to the system performance and reliability of the 5G core. The Key Performance Indicators (KPIs) for evaluating the system performance are listed below.

- CPU usage and memory usage of the core components.
- Variation in operation duration for different procedures performed in the core.
- The rate of operation failures over time.

Before applying the ML algorithm on the collected metrics from the 5G core, the raw data needs to be processed and transformed into meaningful data, so that it can be fitted to the time series forecasting models; this process is known as data preprocessing. The pipeline used for designing the time series based forecasting models follows the standard ML pipeline defined in IMT-2020 [7].

This article concentrates on how well future time steps can be predicted from core network traffic logs using ARIMA and FB Prophet, such that, system failure due to anomalies can be predicted beforehand and proper actions can be taken. So only the ML models and their behaviour will be discussed in details in the later sections. Other steps from the ML pipeline will be skipped in this article.

In order to use Time Series Forecasting Models, time-series data needs to be collected from the 5G core. We have used Fraunhofer FOKUS Open5GCore platform as the base of the design. As already discussed a Benchmarking Tool from the Open5GCore platform is used to trigger periodic network events in 5G Core like – Registration, PDU Connection, Deregistration. The generated data is formatted in time series data and sent to Prometheus as a metric. Before applying ML models, first, the data selection needs to be done.

A. Dataset Selection

Prometheus pushes the data to remote storage InfluxDB. To perform time series forecasting we have selected datasets for the Resource Usages (CPU Usage, Memory Usage for core components) and for Operational statistics (for each type of procedures average duration and avg. failure rate over time). In this article, we will only compare the predicted value for resource usages. Here we have selected datasets only for core components AMF, SMF, UPF; as the load is very high on these components when multiple operations are going on in the core network. These datasets are then pre-processed before using as the input to ML models.

B. ML Models

The processed datasets are non-stationary and there are many irregular patterns. From the Open5GCore platform, sub-daily datasets are collected and they have trend and seasonality components in them. Any ML model that can work on seasonality in sub-daily data and predict future values can be used for the forecasting. Here time series forecasting model ARIMA, FB Prophet is used. These two models are able to handle sub-daily non-stationary data and make predictions. In the Evaluation chapter, the outcome of the prediction using ARIMA and FB Prophet will be evaluated. The application of Recurrent Neural Network (RNN) can be an efficient solution for the prediction but with limited data, the outcome will not be satisfactory. Depending on the outcome of the Forecasting Models, the predicted values can also be used as the input to verify the system state and take corresponding actions.

IV. IMPLEMENTATION

As already mentioned the Open5GCore platform was used for this implementation as the base. Open5GCore toolkit is a network testbed platform developed by Fraunhofer Fokus for the 3GPP 5G core network. The deployment of the testbed for this implementation was done in the Unix based environment.
We have used Time Series Modelling for the predictions. Time Series Forecasting Model ARIMA and FB Prophet were used and parameters were selected by applying the models on training datasets.

A. ARIMA

The ARIMA model is used for time series forecasting. It works on the basic assumption that the considered time series is linear and it follows a statistical distribution like the normal distribution. ARIMA model finds the autocorrelation in the data for forecasting the future time steps. ARIMA model uses subclasses of other models, such as the AR, MA and ARMA models and handles non-stationary data with the integrated aspect.

The ARIMA model can be defined with the parameters using the standard notation ARIMA\((p, d, q)\), where \(p\) is the order of autoregression; \(d\) is the order of differencing; \(q\) is the size of moving average window. For a stationary time-series having no seasonality, the ARIMA\((p, d, q)\) model can be expressed with the function below [3]:

\[
y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - ... - \theta_q \epsilon_{t-q}
\]

where \(\phi_1, \ldots, \phi_p\) are the slope coefficient of the model, \(\theta_0\) is a constant, \(\epsilon_t\) is white noise, \(\theta_1, \ldots, \theta_q\) are the moving average parameters of the model and \(y_{t-1}\) is the data point at time step \(t - 1\).

To find out the values of the parameters \((p, d, q)\) of the model, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are used. These measures show the relation between the observations of the Time Series. To determine the orders of AR, I, MA terms for the modeling and forecasting using ARIMA, these ACF and PACF plots against consecutive time lags are very useful.

B. Facebook Prophet

Prophet is open-source software released by Facebook for time series modeling. The Facebook prophet uses additive model for forecasting time series data. It is able to fit data having non-linear trends with yearly, weekly, and daily seasonality and also holiday effects. FB Prophet has the capability to work on sub-daily time series data having a trend in it. FB Prophet is designed in such a way that it works pretty well on missing data, outliers, sudden changes in the data series and provides reasonable forecasts on them.

The main components of the FB prophet model are trend, seasonality and holidays. These components are combined in the following equation for the model [6]:

\[
y(t) = g(t) + s(t) + h(t) + \epsilon_t
\]

- \(g(t)\): trend of the model, which shows the increase or decrease in the data series. Prophet models the non-periodic changes in the data, by fitting piecewise linear or logistic growth curve over the trend of the time series data. This linear fitting is used so that the model is not affected by outliers or missing data.
- \(s(t)\): seasonality of the model (with Fourier Series), which shows the periodic changes (e.g. weekly/yearly seasonality) in the data because of seasonal factors (based on the time of the year).
- \(h(t)\): holiday effects, this is provided by the user to apply the effects of holidays or big events on the behaviour of time series data.
- \(\epsilon_t\): error term, this stands for any unusual changes that is not fitted by the model.

Facebook Prophet is getting popularity because of the faster and accurate forecasts in seconds, and also it is able to handle messy data with very less manual effort. Prophet can work well with data having outliers, missing data and sudden spikes in the time series data. There are many parameters for trend and seasonality in the dataset, provided by the model which can be tweaked or adjusted for better prediction [5].

Among other parameters, the changepoint prior scale is a very important parameter as it can be used to tackle two very known issues of forecasting overfitting (occurs when model learns too much about detail and noise in the training data that it has an adverse effect on the performance and the new data) and underfitting (it occurs when the model can not fit the training data properly and can not generalize to new data). As the value gets higher, the model will fit a more flexible curve to the time series [5].

The Open5GCore toolkit is implemented on top of the Phoenix platform of Fraunhofer Fokus. Phoenix project has a predefined memory management system, where chunk managers are used for memory usage based on the usage type. There are four chunk managers used in all the modules-cm_globalP, cm_packetP, cm_sessionP, cm_transactionP. To collect the memory usage of core components usage of each chunk manager was collected after each time interval.

The CPU usage for the 5G core components (AMF, SMF, UPF) was collected for both User Time and System Time. User Time is the CPU time used in executing the process and System Time, which is the CPU time for the process used in the kernel.

The proposed solution was working with univariate time series, for each time step a single variable the usage was observed. The available datasets for resource usages were divided into training and test datasets. In this solution, multi-step forecasting (two or more steps in the future time frame) was performed from the training datasets for both ARIMA and Facebook Prophet. Proper selections of parameters for both the models were also done before performing the forecasts and comparing the outputs.

V. EVALUATION

To evaluate the outcome of using Time Series Forecasting on 5G Core data and performing the comparative study between ARIMA and FB Prophet, the Fraunhofer Fokus phoenix platform for Open5GCore was used as a test environment. This platform follows a modular architecture,
where all the 5G core components are implemented as modules and they interact with each other using bindings. The platform supports multi-threading, so that parallel execution of core functionalities can be supported. The phoenix platform was instantiated as a virtual machine in the computer having operating system Ubuntu 16.04 and the below hardware-level specifications.

RAM: 16GB  
Processor: Intel(R) Xeon(R) CPU E3-1270 v6 @ 3.80GHz(8 Cores)  
Disk: 256 GB

A. Datasets

To evaluate the performance of the mentioned models, forecasting was done using the same set of datasets. Datasets for this test was collected from the 5G core by triggering 20 registration, 10 PDU connection, 10 deregistration, 10 registration and PDU connection and 10 handover procedures using the Benchmarking Tool every 40 seconds with the frequency of 5 operations/second. We used 4 datasets for memory usage for each of the components and compared the outcome of the models for each of the 4 datasets for the components. To evaluate the outcome of CPU usage prediction 2 datasets for each of the components were used.

B. Forecasting with ARIMA

The Auto-Regressive Integrated Moving Average model forecasts future values based on the previous values. Proper selection of values for parameters p(AR), d(I), q(MA) are very important for ARIMA to perform better on a given dataset. AIC (Akaike Information Criterion) was used to find out the best fit of the ARIMA model for the collected datasets. AIC gives an estimation of out of sample prediction error for statistical models for a given dataset and is widely used for selecting predictors for regression problems. When comparing ARIMA models having different values for input parameters (p, d, q), the one with the lower AIC is the better fit for a given dataset. Datasets collected for the test case were fit to the best fit ARIMA models with the parameters chosen using AIC (used parameters are listed in TABLE 1).

C. Forecasting with FB Prophet

Facebook Prophet is an open-source forecasting tool for time series data. It uses curve fitting (additive regression models) for predicting future values. The Prophet is very flexible on data that is fed to the model and it takes by default parameter values if not passed explicitly. In order to improve the performance of the prophet model parameters can be configured. Prophet was fit on the same datasets as used for ARIMA for the comparative study. Hyperparameters were selected for prophet using grid search on the datasets. To find out the best fit of prophet model and the corresponding parameters, GridSearchCV and r2 score from the scikit-learn library were used. GridSearchCV provides grid search to generate parameter sets out of the grid of parameter values specified with the param grid parameter, whereas r2 score helps to find out the proportion of variance in the dependent variable. The best possible fit of the model is when r2 score
is 1. Parameters used to fit the datasets to prophet models are listed in TABLE 1. As the datasets carry sub-daily data points, ‘daily seasonality’ parameter was set to true for all the datasets.

Same as ARIMA, datasets were split into train and test datasets for training models and validating the accuracy of models. Fig. 6, shows CPU usage and Fig. 7, shows memory usage (for chunk managers cm_sessionP and cm_transactionP only) prediction for 5G core component AMF. The blue line is showing the actual values and the green line is showing the predicted values. Prophet has functionalities for cross-validating forecasted values using historical data against actual values. From the fbprophet package performance metrics was used to compute MAPE for evaluating the accuracy of the prophet models used.

While doing the testing using both ARIMA and Prophet, it was seen that both of the models performed well under a proper selection of hyperparameters and they are able to predict for multiple time steps in future (here prediction was done for next 5 minutes). This ensures that there is a possibility to grant additional resources (CPU, memory) to the components if the predicted values report resource depletion. However, the percentage of error in predictions based on different test datasets was overall less in the case of the prophet models used.

ARIMA model depends on the datasets, so based on the datasets parameters need to be passed to make the series stationary and have proper MA coefficient before fitting the model. While generating data using Benchmarking Tool there are many outliers in the datasets. The distribution of data also varies based on operations performed and the frequency of those operations. So, to use ARIMA model one has to compute the parameters and train the model every time to fit the datasets. This is not the case for FB Prophet as it adjusts parameters by itself, and is very flexible on the type of datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>ARIMA Parameters</th>
<th>ARIMA MAPE</th>
<th>Prophet Parameters</th>
<th>Prophet MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Chunkmanager-</td>
<td>(p, d, q)</td>
<td>0.00</td>
<td>changepoint</td>
<td>0.01</td>
</tr>
<tr>
<td>cm_globalP</td>
<td>(4,2,3)</td>
<td>0.89</td>
<td>scale:</td>
<td>3.73</td>
</tr>
<tr>
<td>cm_packetP</td>
<td>(4,2,2)</td>
<td>1.08</td>
<td>changeP</td>
<td>0.84</td>
</tr>
<tr>
<td>cm_sessionP</td>
<td>(4,2,2)</td>
<td>1.21</td>
<td>changepoint</td>
<td>1.08</td>
</tr>
<tr>
<td>cm_transactionP</td>
<td>(4,2,2)</td>
<td>1.21</td>
<td>prior</td>
<td>0.84</td>
</tr>
<tr>
<td>CPU Usage-</td>
<td>(p, d, q)</td>
<td>0.53</td>
<td>changepoint</td>
<td>0.16</td>
</tr>
<tr>
<td>System Time</td>
<td>(4,2,3)</td>
<td>0.9</td>
<td>prior</td>
<td>0.48</td>
</tr>
<tr>
<td>User Time</td>
<td>(4,2,3)</td>
<td>0.76</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: ARIMA and Prophet Model Parameters Along with MAPE for AMF

VI. CONCLUSION

In this article, we have proposed a concept for using Time Series Forecasting on 5G Core network data. Forecasting was done using time series forecasting models ARIMA and FB Prophet. This article also showed a comparative analysis of the outcome of the models and gave us insight into the behaviour of the models for 5G Core network data. It also opens up the possibility of fitting time series model on network data to have a better prediction of future events and enable better resource management, thus avoiding sudden unexpected failures within the 5G network.

The main advantage of FB Prophet is, it works considering the aspects trend, seasonality and holiday effects on the data. One can work with Prophet without having much statistical knowledge. The prophet model is able to handle outliers and missing values in the time series data, unlike ARIMA which needs regular spaced measurements. In this article, it was shown that, Prophet also has the capability to apply seasonality effect on sub-daily data. Apart from that while predicting the future values, it also provides a maximum and minimum range for the prediction, which is very helpful for system failure checking. It can be used as the limit of usage that actual values should not cross (can be recorded as anomalies) in a stable system. However, for highly irregular datasets, prophet does not give very good results. As we have seen in the evaluation, using FB Prophet the prediction was better for CPU usage than memory usage of the components.

On the other hand, ARIMA is capable of working on both stationary and non-stationary data, but it works best with stationary data. One needs to have statistical knowledge to use ARIMA properly. In order to tune the model the correct values of parameters p(AR), d(I) and q(MA) should be identified. ARIMA works with the trend in the data but it is not able to consider the seasonality component. SARIMA can be used
in place of ARIMA to have both trend and seasonality into consideration.

In this article we have worked with small datasets due to lack of data availability. As both ARIMA and FB Prophet have the capability to work on small datasets and provide good predictions capturing different components of time series data, we have selected these two time series models for the evaluation. Other time series models can also be used based on the availability and structure of the datasets.

Network management and orchestration is a challenging work in a complex network like 5G. Machine Learning has the capability to solve this issue. ML can enhance network management; by gaining insight from network-generated data and yielding predictions for supporting optimization in network operations and maintenance. The standard ITU Y.3172 is designed to address this issue and it provides insights of an architectural framework for networks to accommodate Machine Learning with it.

Applying ML on 5G core network is a vast area of research and there can be many possibilities for future work in this direction. Here, the future possibilities are cited taking current Open5GCore as the base platform. This article depicts the possibility of integrating ML with 5G core and improving reliability and overall performance of the system. Subscriber location can also be collected with the operation metric and that can be used to find out system load based on the location. This information can be used for allocating resources and better load balancing on the core network side to ensure network availability during the high amount of usage at the core network.

Shifting from 5G to 6G communication networks is already a hot topic for researchers. 6G networks will be complex than 5G because of being context-awareness, ubiquitous, reconfigurable, and having intelligence in mobile devices and network infrastructure. ML-based solutions/AI will be a big trend in 6G for network resource management, network planning and optimization, also for failure detection.

REFERENCES