System Failure Prediction within Software 5G Core Networks using Time Series Forecasting

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 network is very flexible because of the two concepts Network Functions Virtualization (NFV) and the Software Defined Networks (SDN). There are various use cases for 5G technology and for different cases different configuration of the network will be needed. 5G Technology will bring intelligence within the network. The ability to support massive connectivity across diverse devices will result in enormous data volume within the 5G network. Continuous monitoring and traffic log analysis in such a complex architecture will not be sufficient to ensure availability and reliability within the network. The integration of data analytics within the 5G network can leverage the potential of automation. By introducing automation in the monitoring process better Quality of Services (QoS) can be achieved and analysing the network traffic load for better bandwidth utilization within the network. This article proposes a solution to integrate time series based analytics with 5G core and predicting any threats within the system which can lead to system failure. To validate the proposal Fraunhofer FOKUS Open5GCore toolkit is used.**

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

I. Introduction

5G technology is a total paradigm shift from 4G/Long Term Evolution (LTE). The architecture is composed of multiple layers of virtual functions, virtual and physical Radio Access Network (RAN) assets, spectrum usage, distributed computing nodes, which are based on mainly SDN and NFV concepts. The convergence of different Information and Communications Technologies (ICTs) will make the way for lots of applications with varying service requirements. As a whole, the environment will be more complex for 5G compared to 4G/LTE. Monitoring in such a complex network will be very challenging and it will need an efficient, cost-effective solution for the network management.

Machine Learning (ML) is a hot topic in the research industry. ML is used in different applications. For years ML has been used to automate network management related tasks mostly in the context of traffic prediction, routing and classification, resource and fault management, network security etc. Due to improved ML techniques, advanced computing capabilities and mostly for better availability of data; ML has become a popular approach to handle the complexity of next generation networks. ITU-T Y.3172 specifies an architectural framework for ML in future networks which includes ML pipeline, management and orchestration functionalities [2].

Current monitoring systems, which are used for network management is mainly deployed in specific locations in the mobile network, and they have delays and overheads for capturing traffic logs. 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 and managing network resources will be very challenging for the operators. This process will cause much more overhead and delay in the whole procedure.

The ability to support massive connectivity across diverse devices will result in enormous data volume within the 5G core network. The availability of data at the 5G core can help in visualizing and analysing the system state at each time step if captured in an informative way. The availability of data enables the use of ML in 5G core, which can result in better network management by predicting issues like security threats, anomalies, failures within the core network and taking proper actions for them.

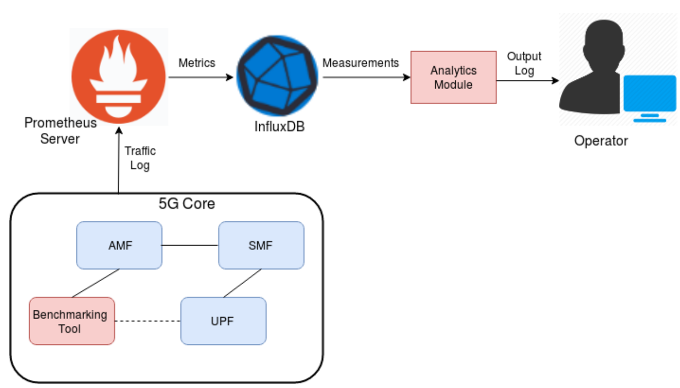

Fig. 1. The architecture for time-series based analytics in 5G network

In this article a solution is proposed to integrate time series based analytics with the 5G Core system in order to detect any possible failures within the system which will result in enhanced reliability and performance of the core network. The outline of the proposed solution is shown in Fig. 1. Fraunhofer FOKUS Open5GCore ([11],[12]) testbed will be used as the base of the solution. To predict any failures within the 5G Core network traffic log from 5G core will be collected as time series data and will be pulled by the monitoring tool Prometheus. From Prometheus metrics will be pushed to the database InfluxDB. Time series based analytics will be integrated with the platform to forecast any possible errors in the system.

The remainder of the paper is constructed as follows, Section II gives the background for the proposed solution, where Section III shows the design concept. Section IV depicts the implementation of the solution on top of Fraunhofer FOKUS Open5GCore. In Section V evaluation of the proposed solution is performed and Section VI concludes the paper.

II. BACKGROUND

In any kind of cellular network, the availability is very important because with higher availability, better customer services can be provided. To improve QoS within a network better visibility of the network is very essential. ML based analytics enables the capability for a system in scrutinizing data and gaining knowledge out of it.

Many types of researches are happening on the application of ML in various contexts of cellular networks for better QoS and Quality of Experience (QoE) to the customers. Network traffic is an important part of the network and prediction of traffic can play a key role in network management for complex networks. To predict network traffic, Yu et al. were the first to apply ML using Multi-layer Perceptron Neural Networks (MLP-NN) [3]. Whereas, Li et al. focused on network traffic flow instead of traffic volume by proposing a frequency domain based method to predict incoming and outgoing traffic volume on an inter-data centre link [4]. Resource management is another important aspect of the network, which includes controlling vital resources like CPU, memory, disk, switches, routers etc. For wireless networks, Piamrat et al. proposed an admission control mechanism for resource management based on subjective Quality of Experience (QoE) perceived by end-users [5]. Another crucial aspect is fault management by predicting upcoming network failures and performance degradation. One of the first ML-based approaches were proposed by Maxion et al. for detecting anomalous events in communication networks [6].

In next-generation networks like 5G, network management will be very complex. The 5G architecture is very complex and there is a high chance of failure of the system because of heavy operation load on the core network. So network management becomes very challenging to ensure availability.

A. Open5GCore Platform

Open5GCore platform (shown in Fig. 2.) which is used in this solution is an implementation of 3GPP 5G core network. It mirrors a prototype from the 3GPP Release 15 for the core network functionality and its integration with 5G New Radio (Standalone and Non-Standalone)[11]. In this platform, 5G core network functions are implemented with a modular design on top of the Phoenix platform of Fraunhofer Fokus. Each module exposes an API, through which other modules can exploit the functionalities of this module. Network functions which are related to this article are discussed below.

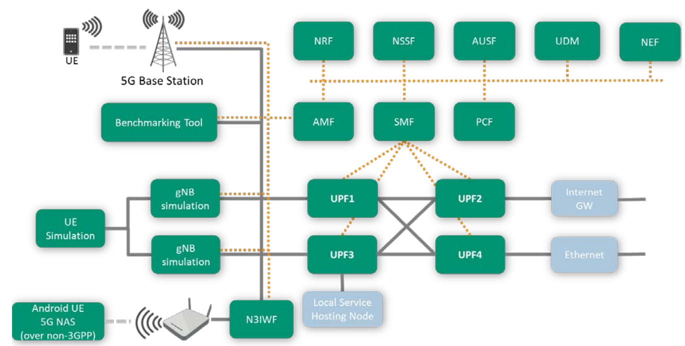

Fig. 2. Open5GCore Platform

AMF The Access and Mobility Management Function which is decomposed from 4G Mobility Management Entity (MME) receives all connection and session related information from the UE. The primary tasks for AMF are connection and mobility management for the UEs.

SMF Session Management Function which is also decomposed from 4G MME manages all the Protocol Data Unit (PDU) sessions related functionalities like establishment, modifications, release in association with UPF [1].

UPF User Plane Functions are responsible for data plane functionalities of 5G system [1]. Packet Forwarding Control Protocol (PFCP) is used for communication between the control and user plane functions.

Another important component of the platform which is used in the proposed solution is the **Benchmarking Tool**. It has been designed as a component where multiple User Equipment (UE) and RAN have been emulated, for generating the traffic in the 5G core.

B. Time Series Forecasting

A time series is a series of data points collected in a timely order, i.e. during equally spaced time intervals. Time series forecasting which works on time series data is one type of prediction problem where one has to analyze current and historical facts to find out future events and has a time component [7]. Time series Forecasting is a wide research area and various time series forecasting models have evolved over time like AutoRegressive Integrated Moving Average (ARIMA), Holt-Winters, Facebook (FB) Prophet.

C. Facebook Prophet

FB Prophet used in this solution is an open source time series forecasting tool which uses curve fitting for predicting future time steps from time series data. The main components of the FB prophet model are trend, seasonality and holidays. These components are combined in the following equation for the model [8]:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

- $g(t)$**:** 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 curves 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.
- ϵ_t**:** error term, this stands for any unusual changes that are not fitted by the model.

III. CONCEPT

In this article the concept of using Time Series based analytics to predict failures within 5G core network is proposed. The design is based on the concept of collecting traffic load from Open5GCore platform and predicting future values from that to identify any potential threat in the core network in accordance with system failure.

The Benchmarking Tool (BT) within the Open5GCore platform is used to trigger network events and generate network traffic from the core network. BT emulates the UE together with gNodeB for the 5G core and provides functionalities to perform 3GPP standard specific operations.

The most important requirement behind the proposed solution is to predict 5G core network failures so that network reliability can be improved. To address this issue the time series based analytics is designed for the Open5GCore. Four important metrics, which are directly related to the system performance and reliability of the 5G core, are considered as the parameters for the prediction algorithm.

- CPU usage of the core components.
- Memory usage of the core components.
- Average operation duration for different procedures performed in the core over a specific time interval.
- The rate of operation failures over a specific time interval.

The above metrics are used as the Key Performance Indicators (KPIs) for evaluating the system performance. The pipeline used for designing the time series based forecasting models follows the standard ML pipeline defined in IMT-2020 [9]. The load is very high on the core components AMF, SMF and UPF when multiple operations are going on within the core network, so here the resource usages are mainly collected for these components to predict system failure.

Time series forecasting model FB Prophet is used for the failure prediction. Because of limited set data availability Recurrent Neural Network (RNN) could not be applied on the datasets. Here the solution is working with univariate time series where for each time step a single variable is observed and multistep forecasting (two or more steps in the future time frame) is performed from the history data.

The failure prediction is done in few stages and different approaches are taken for the four metrics (CPU usage, memory usage, operation duration and failure rate). For the resource usages and operational statistics future time steps are predicted and they are compared with the resource provided or the actual statistics to check if there are any anomaly captured which can lead to system failure. If there is any possibility of failures within the system and that can be predicted earlier proper actions can be taken to mitigate the failure and enhance system's availability, reliability and overall performance of the 5G core network system.

IV. IMPLEMENTATION

The proposed solution was implemented on Open5GCore platform as mentioned earlier. For the implementation the deployment of the Open5GCore testbed was done in the Unix based environment, as it has support for machine learning algorithms and flexible for different programming languages.

The proposed solution has two main implementation aspects-

- Generating Traffic logs using Benchmarking Tool and from core components. Collecting them as metrics using Prometheus-2.9.1 and InfluxDB-1.7.7
- Applying ML algorithm on the data batches for analysing and finding anomalies in the traffic logs for system failure prediction.

A. Dataset Collection

BT was used to trigger network events periodically and collect the metrics. For each of the procedures performed within the core network the operation statistics like average duration and failure rate were collected using BT and from the 5G core components resource usages were collected. Using Prometheus these metrics were pulled and pushed to InfluxDB database for the ML algorithm to execute on. Before applying the ML algorithm for predicting system failure data preprocessing was done on the stored data in InfluxDB.

Fraunhofer Fokus Phoenix platform on which Open5GCore toolkit is implemented, has a predefined memory management system. In this platform for all the 5G core components chunk managers are used for memory usage based on the usage type. The chunk managers used in all the module based implementations of core components are cm_globalP, cm_packetP, cm_sessionP, cm_transactionP. The memory usage of the core components are gathered by collecting the chunk manager usages after each time interval.

To collect the CPU usage for the 5G core components (AMF, SMF, UPF) User Time and System Time usages were collected for each of the components. Where User Time is the CPU time used in executing the process and System Time is the CPU time for the process used in the kernel.

B. Forecasting

Time Series Modelling was used to predict the resource usages for the next time steps. In this proposed solution multiple-step forecasting in the future time frame was done for predicting the usages. Network traffic may have unexpected behaviour in time and the model should be able to handle that to continue the prediction task. The processed datasets were non-stationary and there were many irregular patterns.

In this solution sub-daily datasets were collected and they had trend and seasonality components in them. The Facebook prophet is more robust to missing values in the time series and outliers. The article [10] shows the outcome of the comparison between ARIMA and FB Prophet for the selected datasets, and FB Prophet performed better compared to ARIMA model. Another advantage of FB prophet is that it not only forecasts the value in future time steps from the previous observations but also provides the range (min and max of predicted value) of the prediction based on the confidence level. This range is very useful to check the predicted maximum usage of resources. The proposed solution is handling UE initiated events in the core network mostly and Prophet works very well for events having human interaction.

C. Failure Checking Algorithm

To do the forecast FB Prophet was used and the predicted values from the FB prophet model were used for the system failure check. The design of the algorithm is based on checking the resource usage and anomalies in the system. The checkings which are done by the algorithm described in the below mentioned steps and are shown using Fig. 3.

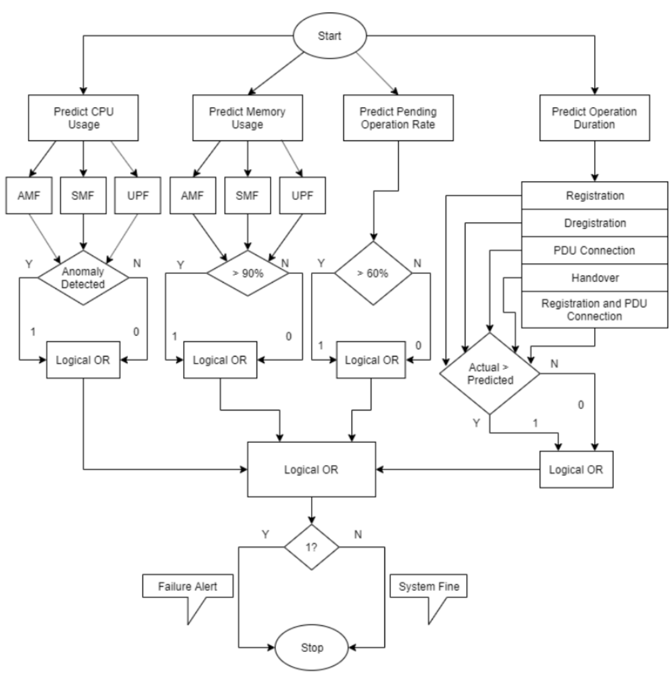

Fig. 3. Flowchart for the System Failure Check Algorithm

- For the CPU Usage Check, the maximum values of the system time and user time predicted for the components are taken. They are compared with the actual values for that time interval. If any of the actual value is greater than the maximum usage that is shown as failure alert in the CPU usage graph and considered as an error in the log.
- For Memory Check, the predicted chunk numbers used for each chunk manager is taken for the components and that is multiplied with the corresponding chunk size. Then a check is done on the total usage of that chunk against the total size allocated for that chunk manager. If the predicted size is more than 90% of the allocated size, that is shown as a failure alert in the memory usage graph and error in the captured log.
- For Operation Failure check, the predicted pending operation rate is taken for the next time intervals and check if the average pending operation rate during the next time steps are more than 60%. In that situation operation failure rate graph shows failure alert and the log captures that event as an error.
- For the Operation Duration Check, the maximum value predicted for the duration is taken for each operation type and that is compared with the actual value for that time interval. If the actual value is greater than the maximum possible predicted duration, failure alert is shown in the operation duration graph. In the log, that condition is considered as an error.

V. EVALUATION

To evaluate the performance and usability of the proposed solution for predicting system failure in 5G software networks, Fraunhofer FOKUS Open5GCore platform was used. The platform has the capability to support multi-threading and supports parallel execution of 5G core functionalities. To do the evaluation, the 5G core environment was instantiated by instantiating Phoenix platform 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

In order to evaluate the performance of the algorithm discussed in the implementation chapter for predicting the failures different tests were performed. Datasets were collected from 5G core for each of the test scenarios to feed into the algorithm. Benchmarking Tool from the Open5GCore platform was used to trigger different 5G procedures such as registration, PDU connection, handover, registration and PDU connection continuously after a predefined time interval, that was mentioned as configuration parameter interval_sec in BT. The generated metrics from the tests were stored in InfluxDB through Prometheus. The algorithm was executed on the stored data for each test case. To evaluate the performance, for each of the core components 4 datasets for memory usage and 2 datasets for CPU usage were collected and were fed to the prediction algorithm in all the test scenarios to compare the outcome of the model.

B. Modelling 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. Modelling with FB prophet is very flexible on datasets that is feed to the model and it takes by default parameter values if not passed explicitly. Parameters can be configured in order to improve the performance of the prophet model. Based on the results from the article [10] FB Prophet was chosen over ARIMA model for the failure prediction algorithm. Hyperparameters were selected for prophet using grid search on the datasets. Best fit of prophet model and corresponding parameters were find out using GridSearchCV and r2 score from the scikit-learn library [10]. The datasets selected to execute the algorithm contained sub-daily data points, so 'daily seasonality' parameter was set to true for all the datasets.

C. Failure Prediction

Different test scenarios were performed for evaluating the accuracy of the implemented ML algorithm for failure check within software 5G networks.

Test No.	Actual Outcome	Predicted Outcome
1	Failure Procedures failed SMF stopped responding	Failure Anomaly detected SMF CPU usage crossed maximum predicted usage. procedure durations longer than expected
2	Failure Procedures failed SMF stopped responding	Failure Anomaly detected SMF CPU usage crossed maximum predicted usage.
3	Failure UPF Crashed Out of Memory for cm_sessionP	Failure Predicted chunk usage for UPF cm_sessionP crossed 90% of allocated
4	No Failure	Failure Higher SMF CPU usage than predicted
5	Failure UPF Stopped responding	Failure Higher UPF CPU usage than predicted
6	Failure SMF and UPF stopped responding	Failure Higher SMF, UPF CPU usage than predicted
7	Failure AMF cm_transactionP was not accessible	No Failure could not predict failure
8	No Failure	Failure No available UEs in BT for further registration
9	Failure System failed within 10 minutes of procedures initiation	No Output Could not train prediction models insufficient data
10	Failure AMF and SMF stopped responding	Failure Higher AMF, SMF CPU usage than predicted

TABLE I
TEST CASES WITH THEIR ACTUAL AND PREDICTED OUTCOME

The actual outcome (system failure encountered or not) and predicted outcome (system failure check algorithm output) are listed in Table 1, corresponding to the test numbers for few of the test cases performed. For each of the test scenarios, collected datasets for resource usage, operation statistics were split into train and test datasets. The train datasets were fed to the ML algorithm to forecast future time steps. The proposed ML algorithm worked on the predicted values and data points from test datasets to find out any potential case of failure within the system.

Output of the algorithm for memory usage and cpu usage check for future values are shown using figures here. Fig. 4. shows the output of memory check on a dataset captured from test 3 for SMF and UPF components. In this test UPF crashed because of not having enough memory chunks to execute the processes. In Fig. 4. the output of the algorithm also shows that predicted chunk usage for chunk manager cm_sessionP is more than the available chunks in that chunk manager for UPF because of which UPF can go out of memory and lead to failure. So, corresponding predicted values are shown using a red line to indicate the predicted failure. As memory usage in SMF were within limit, so the predicted values are shown using a green line to indicate no failure condition.

The output of CPU check on the dataset captured from test 6 is shown in Fig. 5. System encountered failure due to high CPU usage in SMF and UPF. The prediction of CPU usage for system time in SMF and for both system time and user time in UPF, using the algorithm shows anomaly with regards to the actual CPU usage before the failure occurred. Actual CPU usage gets higher than the predicted usage (based on the previous values) for the components and stops responding to any procedures which results in failure of procedures in the 5G core. So, the corresponding predicted curves are shown using red lines. In case of user time CPU usage for SMF, the actual usage is within the limit of predicted usage which depicts normal behaviour of the system so the predicted curve is shown using green line to indicate no failure condition.

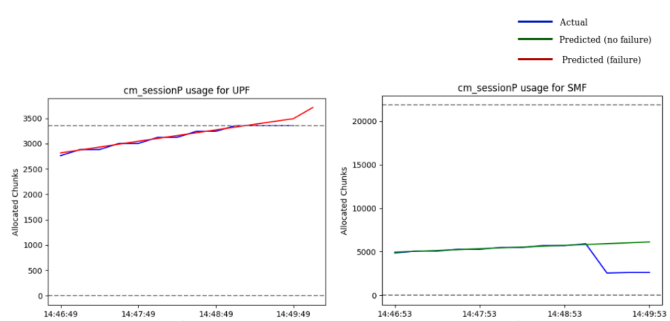

Fig. 4. Failure check output on session memory usage for UPF and SMF

To evaluate the accuracy of the system failure check algorithm- Precision, Recall, F1 Score metrics were calculated from the outcome of the algorithm. These metrics can be calculated with the below mentioned formulas.

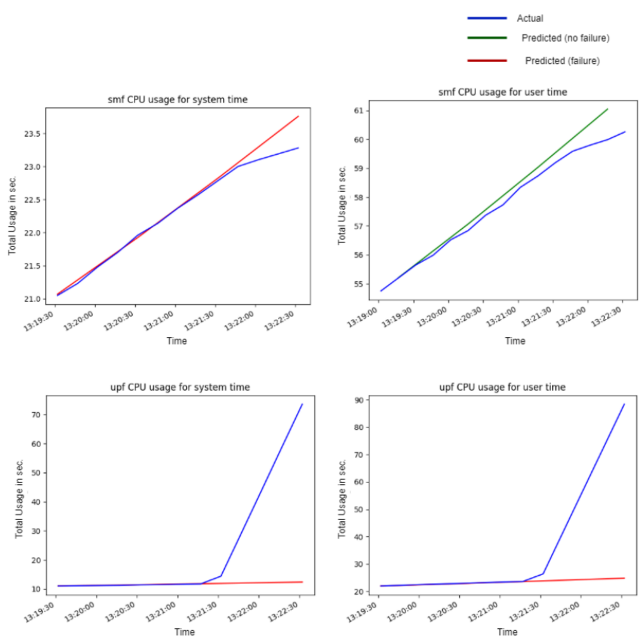

Fig. 5. Failure check output on CPU usage for SMF and UPF

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

True Positive (TP) in this case is 6 as the algorithm could predict failure 6 times out of 8 failures encountered.

True Negative (TN) is in this case 0 as there were 2 cases where the system did not encounter any failures but the algorithm predicted failures.

False Positive (FP) in this case is 2, as the algorithm predicted in 2 cases failures though there were no failures encountered.

False Negative (FN) is in this case 2, as for 2 cases algorithm could not detect failure that encountered in the system.

Using the formulas mentioned for Precision, Recall, F1-Score, the metrics were calculated for the algorithm based on the outcome of the test scenarios. The Precision of the algorithm came as 0.75, Recall was also 0.75 and the F1 Score came also as 0.75. Because of limited datasets, 10 test scenarios were performed to calculate the above mentioned metrics. With higher test scenarios the output of the metrics can vary based on the predicted and actual outcome.

VI. CONCLUSION

In this article a concept for integrating time series based analytics with the 5G Core network was proposed. Traffic log was collected from the core network as metrics using Benchmarking Tool and Prometheus tool. On the collected metrics time series based forecasting was performed and verification was done using ML algorithm, to predict cases of system failures. Time series model Facebook Prophet was used to predict future network events to forecast failures beforehand to avoid unexpected failures within the 5G network which enhances availability of the system.

5G technology will bring a massively improved platform for delivering scalable and reliable connectivity to the world. Because of high data-rate and low-latency that 5G will offer, a wide range of applications will be deployed and that will result in huge number of connected devices to the network. More devices will result in massive amounts of data traffic inside the 5G core network. Because of the increased load on the core network there is a high chance of failures within the network and maintaining availability and reliability of the network will be very challenging for the operators.

Based on the evaluation of the proposed solution the accuracy of the system failure check algorithm had a positive outcome. This leads to the possibility of integrating the solution with 5G core and improving reliability and overall performance of the system with efficient use of the information provided by the solution.

The main problem encountered during the whole process was to collect the data from the 5G core. Because of the system instability and also different behaviours of the system data collection was a huge problem. Due to limited datasets many time series models were not possible to fit. Facebook Prophet provides good prediction with small datasets having outliers, missing values. It also works well with datasets having seasonality and trend characteristics in them. In this solution FB Prophet worked pretty well for predicting resource usages which results into better performance of the system failure check algorithm.

As already discussed only small datasets were collected for the current implementation and evaluation. If more data points can be collected then it will open way for wide area of case studies and will help in identifying more failure cases. Thus, the algorithm will have more detail coverage on failure scenarios with improved outcome. This will also enable other ML models (especially neural networks) to be applied on the datasets, which could not be tested because of lack of data and limited time. The use of other forecasting models to find out the output of the proposed algorithm and doing comparative study with existing methods is beyond the scope of this paper and can be part of future work.

Another possible improvement is, the ML algorithm can be integrated with 5G core and the algorithm can be extended to collect data on a periodic basis and show system status in a continuous manner. This can help in receiving continuous report from the network which can help the network operators to manage the network more efficiently by mitigating any chances of failure within the system. 6G network will be more complex than 5G network and because of context driven decisions in 6G network ML based solutions will be a big trend in the management of the 6G network.

REFERENCES

[1] 3GPP. "System Architecture for the 5G System." Available: https://www.etsi.org/deliver/etsi ts/123500 123599/123501/15.02.00 60/ts 123501v150200p.pdf(2019).

[2] Marco Carugi. "Distinguishing features - and high level requirements of 5G/IMT-2020 networks." April 2019. Available: https://www.itu.int/en/ITU-D/Regional-Presence/ArabStates/Documents/events/2019/ET/S1-%20ITU%20Reg%20Forum-Tunis-5G%20IMT2020-presentation-Marco-Carugi-v1.pdf.

[3] Yu, Edmund S., and CY Roger Chen. "Traffic prediction using neural networks." Proceedings of GLOBECOM'93. IEEE Global Telecommunications Confer- ence. IEEE, 1993.

[4] Li, Yi, et al. "Inter-data-center network traffic prediction with elephant flows." NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2016.

[5] Piamrat, Kandaraj, et al. "QoE-aware admission control for multimedia applications in IEEE 802.11 wireless networks." 2008 IEEE 68th Vehicular Technology Conference. IEEE, 2008.

[6] Maxion, Roy A. "Anomaly detection for diagnosis." [1990] Digest of Papers. Fault-Tolerant Computing: 20th International Symposium. IEEE, 1990.

[7] Jason Brownlee. "Components of Time Series." Aug 2019. Available:https://machinelearningmastery.com/time-series-forecasting/ (2019).

[8] Taylor, Sean J., and Benjamin Letham. "Forecasting at scale." The American Statistician 72.1 (2018): 37-45.

[9] ITU-T. "Architectural framework for machine learning in future networks including IMT-2020." July 2019. Available: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=13894lang=en (2019).

[10] Chakraborty, Pousali, Marius Corici, and Thomas Magedanz. "A comparative study for Time Series Forecasting within software 5G networks." 2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS). IEEE, 2020.

[11] Fraunhofer FOKUS. "Open5GCore – The Next Mobile Core Network Testbed Platform." Available: www.open5gcore.org(last visited on 31.08.2020).

[12] M. Corici et al. "Paving the Way for Local and Industrial 5G Networks and Testbeds." April 2019. Available: https://ercim-news.ercim.eu/en117/special/paving-the-way-for-local-and-industrial-5g-networks-and-testbeds.

[13] He, Weisong, et al. "Applying multiple time series data mining to large-scale network traffic analysis." 2008 IEEE Conference on Cybernetics and Intelligent Systems. IEEE, 2008.

[14] Kim, Eungha, and Young-il Choi. "Traffic monitoring system for 5G core network." 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2019.

[15] Al Mamun, SM Abdullah, and Mehmet Beyaz. "LSTM recurrent neural network (RNN) for anomaly detection in cellular mobile networks." International Conference on Machine Learning for Networking. Springer, Cham, 2018.

[16] Benslimen, Yosra, Hichem Sedjelmaci, and Ana-Cristina Manenti. "Attacks and failures prediction framework for a collaborative 5G mobile network." Computing (2021): 1-17.