QMP: A Cloud-native MLOps Automation Platform for Zero-Touch Service Assurance in 5G Systems

Georgios Samaras *, Vasileios Theodorou *, Dimitris Laskaratos *,
Nikolaos Psaromanolakis *, Marinela Mertiri * Alexandros Valantasis *
* Intracom Telecom, Greece
email: {gsamaras, theovas, dlaskaratos, nikpsarom, marmert, savalant}@intracom-telecom.com

***Abstract*—In this paper we present QMP—an AI-driven platform for proactively assuring Network Slices' Quality of Service (QoS)— and a demo of its MLOps automation features applied on vCDN Slices at the network edge.**

I. Overview

End-to-end automation of 5G networks' and services' management has been identified as a necessity to cope with the complexity and extreme requirements of softwarized, cloudified and agile services that form the basis of network slices [1]. In this direction, several mechanisms have been proposed and even standardised (e.g., the 3GPP 5G network data analytics function (NWDAF)) for the collection of monitoring data and the streamlining of analytics to generate insights and to take action towards network slice optimization and service assurance. In turn, the optimization analysis and the decision making for actuation and adaptation to close the automation loops, is largely performed nowadays in a zero-touch manner, building on Artificial Intelligence (AI) / Machine Learning (ML) techniques for tasks such as 5G performance KPI prediction, anomaly detection, placement and resource scheduling for efficiency and footprint minimization.

Despite an abundance of available AI/ML techniques and algorithms that could be used for automated and optimized network slice lifecycle management (LCM), these operations still entail manual configurations and long implementation cycles by expert teams. Several barriers to automation, including the lack of practical tools to operationalize such assets and the shortage of combined skills in AI/ML and 5G networking technologies, are further reinforced by the restrictive amount of resources needed for the training of sophisticated AI/ML models and their application in the field.

In this demo, we present the *QoS Monitoring and Prediction Platform* (QMP), which is a cloud-native solution for the reliable and efficient proactive detection and mitigation of QoS drops in network slice operation, embodying MLOps automation processes and utilizing minimal containerized resources for the LCM of ML models. To this end, QMP produces insights and notifications about imminent performance degradation of network slices, based on online incoming data about different layers of a 5G System, i.e., networking QoS; compute and storage utilization; and performance and application-layer monitoring metrics. More specifically, such data are fed into QMP, which employs the use of its internal sophisticated

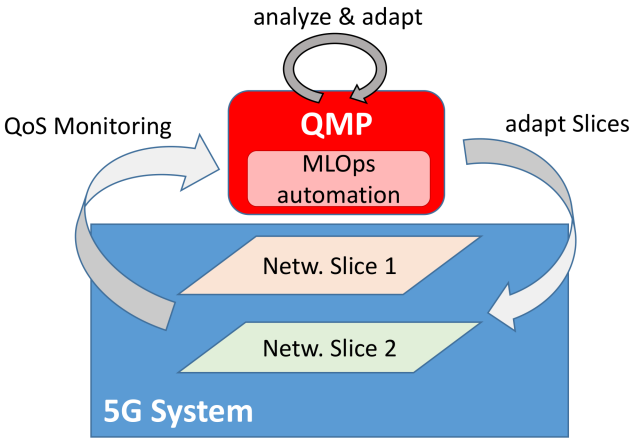

Fig. 1. Quality of Service Monitoring & Prediction (QMP) platform.

Machine Learning (ML) pipeline in order to trigger alarms whenever needed, e.g., if a prediction is below a given QoS threshold for a given performance metric, such as bandwidth. In case an alarm is triggered, subscribed management modules to the event are alerted to make informed adaptations, in order to assure that the future network demands will be served without any deflection in QoS requirements and end users' experience. Examples of such adaptations would be to scale out the network slice's underlying infrastructure resources or to switch users from their assigned network slice to another.

In the present demo, the traffic is monitored and evaluated using Content Distribution Network (CDN) applications, deployed in a 5G-enabled cloud facility. In real world scenarios, this kind of traffic may evolve in such a way that significantly modifies the distribution of the monitoring data that are fed into QMP. For this reason QMP is designed as a platform that continuously analyzes incoming data, and if needed, adapts itself to maintain highest possible accuracy over its provided analysis and prediction. This is done by updating its internal ML pipeline, i.e., selecting best-fit ML models and their hyperparameterization, commencing on the fly model (re-)training cycles etc., in order to reflect on the recent incoming data distribution and characteristics. Such self-adaptation actions of QMP on its internal ML pipeline are performed upon autodetection by QMP of significant drops in its effectiveness to produce reliable results, to avoid unnecessary resource con-

sumption. In the present demo, a 5G Network Slice supporting a CDN service is being monitored with the goal to trigger slice adaptations before the performance of the slice is decreased. QoS Monitoring performed by QMP and Slices' adaptation is illustrated in Figure 1, as well as the QMP MLOps automation and its internal analysis and adaptation feedback loop.

II. Innovation

A. Problem Formulation

We consider a scenario involving users streaming online video content from deployed CDN slices in a 5G cloud infrastructure as depicted in Figure 2, where video chunks are streamed according to HTTP Live Streaming (HLS) protocol. At the same time, it is assumed that the content is provided concurrently at multiple CDN slices, which for the case of our demo are represented by two 5G network slice. Moreover, the edge caches are located in the same geographic area, as it is assumed that there are two different CDN providers serving the same region. More specifically, the end-users should be served by the end to end (E2E) CDN slices that can deliver the agreed QoS. A slice consists of a) the centralized CDN component, i.e. "Headend & Mgmt VNF", which is responsible for the management and administrative functionalities of CDN (such as end-user registration etc.), and b) the "Edge Cache VNF", which is the Edge component responsible for temporarily storing and streaming the video to the end-user. It should be noted that Edge Cache VNFs can form a layered hierarchical network of caches to optimally distribute video chunks to geographically distributed end users, but for the sake of simplicity in this demo, we assume only one layer of Edge Caches. Also note that gNodeB is able to redirect the content from any CDN slice, deployed at its adjacent (micro-)cloud, to the end-users to which it provides mobile access. Hence, intelligent optimization and adaptation of the CDN-based streaming services is in place for multi CDN selection, i.e., the proactive redirection of users from one CDN slice to another, whenever the former is assessed as incapable to support expected Quality of Experience (QoX) in the near future. In order to tackle this challenging task, QMP that will be presented in the following subsection, is used to predict services' QoS and evaluate their compliance with expected end users' QoX. QMP is getting data, in real time, from a monitoring service that collects data from various sources. In this way, the monitoring service gathers application and network metrics from both slices and passes them to a message broker reachable by QMP, as per QMP's request.

B. Novelty

QMP is a cloud-native MLOps automation platform for zero-touch service assurance in 5G networks. The platform has an ML models' repository populated with trained models, which have been trained offline on real datasets collected from an operational CDN [2] deployment in US with real users. QMP receives data from the cloud-native monitoring service, shown in Figure 2, in order to produce useful QoS insights, which trigger *prediction* jobs. Initially QMP enters an online

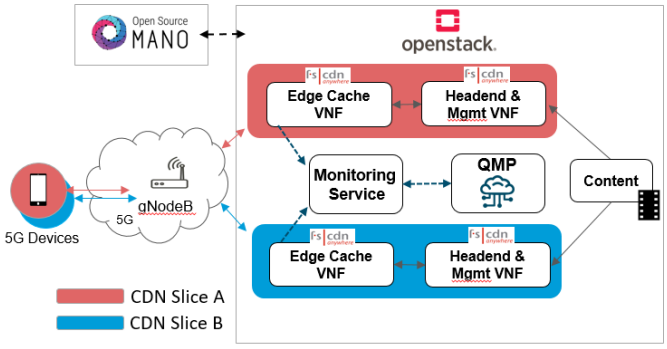

Fig. 2. Use case high-level architecture.

model-selection phase, where the data are fed in every model associated with the given performance metric. In parallel, each of these models produce predictions which are collected for post analysis. Once enough predictions are made, the average accuracy of each model is calculated and the best reporting model against the online data is selected as the default model.

As time progresses, QMP collects and stores the data received by the 5G monitoring service for potential later processing. Periodically, e.g. every 10 predictions, the platform will automatically access the accuracy of its past recent predictions in the current period. In case the accuracy threshold (e.g. 95%) is not met, a Training job is triggered. QMP will, in parallel, load the associated ML models from the repository, preprocess the online data stored by QMP, and start retraining the models. Models are then evaluated and if the accuracy is satisfactory, the models are stored in the repository, ready to be used for prediction. If a retrained model does not pass the evaluation phase, it will get retrained with potentially more data or/and different hyperparameters, from a given set of empirically acquired hyperparameter values that our experiments have shown that work well with the real datasets mentioned before. QMP's default model will be replaced by the best performing model on the newly seen online data. QMP's workflow of its closed-loop automation architecture [3] is shown in Figure 3.

This event-based framework allows QMP to be a truly Zero-touch MLOps platform, which will automatically analyze its accuracy performance and adapt itself automatically through the *training* job. As a result, QMP is capable of evolving and adapting to any data distribution change that may affect the accuracy performance of the insights QMP produces. Moreover, the platform exposes API abstractions for seamless and uniform ML model management. Data analytics mechanisms for data preparation and/or aggregation are implemented, which effectively makes QMP easily configurable, e.g. with forecasting horizons from 1 minute-based to hourly or even to daily predictions. Furthermore, any ML model that implements the minimal abstract API can be plugged into the platform, which allows seamless enhancement and extension of QMP's ML model suite.

The ML model suite of QMP contains three state of the

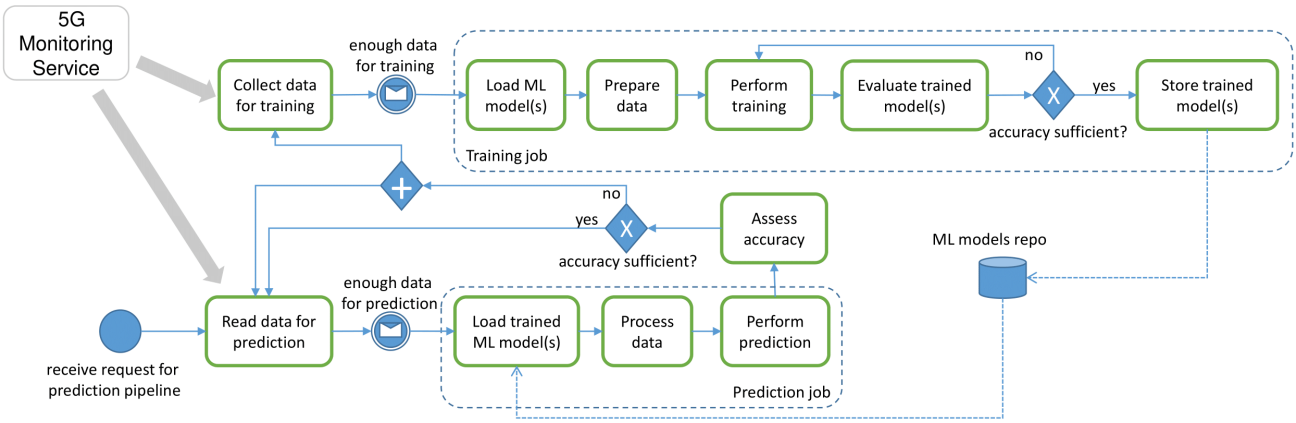

Fig. 3. QMP workflow.

art ML models, written in Python. The first one is the Long short-term memory (LSTM), a Deep Learning (DL) Recurrent Neural Network (RNN), which is rather popular in QoS Monitoring applications. We have the LSTM shipped with Tensorflow ML library [4]. Then we have the Support Vector Regression (SVR) regression model, which is very flexible in defining a model error threshold. Scikit-learn ML library [5] was chosen for a light SVR implementation. Finally, the Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) model is deployed in QMP's ML model suite. N-BEATS is a deep neural architecture based on residual links and a very deep stack of fully-connected layers. We use the N-BEATS implementation found in Darts [6], a library focused on Time Series Forecasting.

C. Technologies used

Regarding technologies used, QMP uses the *Kubernetes* system as its container orchestration system to automatically deploy, scale, and manage its "containerized" jobs, including the ML prediction and inference jobs. *Argo Workflows* container-native workflow engine is used for workflow management and automation, triggering in an event-based manner the instantiation of services and coordinating their message-bus-based communication. This cloud-native architecture allows QMP to use only the required resources in a pay as you go manner, which is critical for a cost effective ML solution. Training and inferencing of ML models are often rather resource intensive operations, especially for DL models. Regarding Monitoring Service, we use a *Kafka* bus as message broker for the ingestion and collection of monitoring metrics, as well as *Prometheus* application along with a *Grafana* platform in order to record real-time metrics and visualize them in interactive dashboards, accordingly. Furthermore, while QMP is agnostic of the Network Functions Virtualization Infrastructure (NFVI) and technologies used for the 5G network slice deployment, those are created and orchestrated with the use of Network Function Vitrualization (NFV) Management and Orchestration (MANO) tools. More specifically, the VNF orchestration and management is done through Open Source MANO (OSM), while Openstack is used as the Virtualized Infrastructure Manager (VIM) of 5G infrastructure. Finally, the CDN application VNFs are 5G-ready customized deployments of Intracom Telecom's full-service CDN commercial solution, namely the fs—CDN™ Anywhere Solution [2] for the realization of live, adaptive (e.g. MPEG DASH or HLS) streaming/TV services.

III. Relevance

With the plethora of emerging 5G applications, demand for meeting the agreed QoS is constantly increasing. To this effect, optimal network management becomes highly critical. A monitoring service which provides robust and accurate insights that effectively allows for proactively avoiding a QoS drop enables intelligent optimization and adaptation of the network. Many researchers already focus their efforts on design and implementation techniques and architectures that will be able to take advantage of such new 5G enabled capabilities. This paper intends to contribute to this direction by evaluating a modern architecture which utilizes cloud-native MLOps automation to its full potential, providing a zero-touch service assurance for 5G networks.

IV. Demo Summary

The main objective of the prototype is to demonstrate the zero-touch nature of the platform as described in the previous section. Specifically, it will revolve around the automatic ML model selection and the re-training when the selected model's accuracy drops significantly, using the network data provided by two pre-deployed CDN E2E slices. The demo will be started by manually executing a simple HTTP POST request, requesting a forecast service. The request's payload will contain the type of data that will be received as input for the ML models and the source of the data, among others. Based on the type of data, QMP will assign a number of ML models that have been trained on that specific type. After this initial configuration, an automated end-user emulation script will be used to generate artificial traffic to the data source, thereby

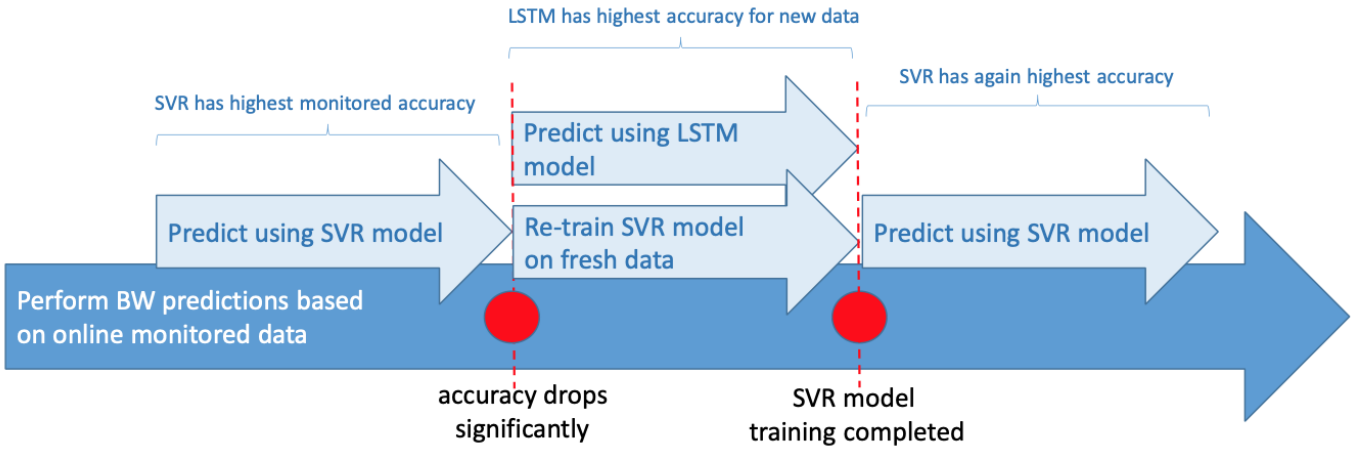

Fig. 4. ML prediction and training scenario.

producing bandwidth metrics that are subsequently received by QMP. It should be noted that the virtual traffic generated follows the pattern of data originating from a commercial deployment of the CDN platform. The ML models that were assigned initially, will start producing their respective forecasts on the incoming input, and every forecast will be compared against the new input to acquire the accuracy. After a number of such forecasts has been reached, the model with the highest accuracy will be selected to carry on the predictions. We will then intentionally fluctuate the traffic from the CDN source drastically in order to force the ML model to produce forecasts with low accuracy. The result will be that after several consecutive low-accuracy forecasts, QMP will order the model to be re-trained with the data aggregated up to that point, while a model with better accuracy will be selected to continue the forecasts. This flow is depicted in Figure 4. The aforementioned implementation architecture is also enriched with metrics monitoring functionalities. Regarding our use case scenario, the metrics are gathered from the QMP component and references the accuracy, error and prediction of QMP's ML models, namely LSTM, SVR and N-BEATS.

V. Demo Processing

The platform will have already been deployed on the Kubernetes cluster before the demo and equipped with pre-trained models that are uploaded on a local database. Additionally, the automated script to generate the traffic to the CDN components will have been set running before the start of the demo. The entirety of the flow described above will be demonstrated through the logging facility of the platform, informing of the result of every step, including the values of the incoming monitoring data and the forecasts of the models.

VI. Demo Material

A recording of the demo can be found in this video[1].

[1]https://youtu.be/2iMYpHHIavo

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 856691 (5G-SOLUTIONS).

References

[1] "ETSI Zero touch network & Service Management (ZSM)." [Online]. Available: https://www.etsi.org/technologies/zero-touch-network-service-management

[2] "Intracom Telecom - full-service content delivery network fs—cdn™ anywhere." [Online]. Available: https://www.intracom-telecom.com/en/products/telco_software/iptv_multiplay/fs_cdn.htm

[3] V. Theodorou *et al.*, "Blockchain-based Zero Touch Service Assurance in Cross-domain Network Slicing," *EuCNC & 6G Summit*, 2021.

[4] M. Abadi, A. Agarwal, P. Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] J. Herzen, F. LÃ¤ssig, S. G. Piazzetta *et al.*, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1177.html