IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 July 2024, accepted 2 August 2024, date of publication 5 August 2024, date of current version 11 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439008

RESEARCH ARTICLE

Optimizing UE Power Efficiency: AI/ML Approach for Upgrade Time Determination

KARTHIKEYAN SUBRAMANIAM[1], NAVEEN KUMAR[2], SUDHAKAR BALUSAMY[3], CHITTARANJAN SAHOO[4], AND GANESH CHANDRASEKARAN[5]
[1]Management S/W Group, Samsung Research and Development Institute, Bengaluru 560037, India
[2]Bearer Software, Samsung Research and Development Institute, Bengaluru 560037, India
[3]SDN and Cloud Solution, Samsung Research and Development Institute, Bengaluru 560037, India
[4]Element Management System Software, Samsung Research and Development Institute, Bengaluru 560037, India
[5]Service Management Orchestrator and Slice Manager Part, Samsung Research and Development Institute, Bengaluru 560037, India

Corresponding author: Ganesh Chandrasekaran (ganesh.c@samsung.com)

ABSTRACT The transition of Control Plane (CP) block functions into software entities, as proposed by 3GPP, necessitates periodic downtime for maintenance activities such as software upgrades or failures. This downtime requires the disconnection of all User Equipment (UE) connections to the CP, triggering the UE reattach procedure and resulting in increased UE power consumption and spectrum wastage. To mitigate these challenges, optimal CP upgrade timings should align with periods of low traffic. In this paper, we propose an AI/ML-based procedure to autonomously determine the optimal time to upgrade CP block functions, eliminating the need for manual intervention by operators. Our approach involves analyzing traffic conditions using statistical data from several CP blocks managing base stations across various areas, including residential and non-residential zones like subways, shopping complex and hospitals. Leveraging Seasonal Auto-Regressive Integrated Moving Average (SARIMA) forecasting, we predict bearer statistical data to calculate the optimal CP software upgrade time, validated using Z-Score analysis at the same time. In addition to address the suboptimal upgrade timings, we also proposed CP Outage Handling Procedure (COHP) v2 by preserving UE contexts during CP upgrades. Our results demonstrate SARIMA's high accuracy in predicting lean traffic conditions, with an R-Squared score of 0.99. Furthermore, upgrading CP software during predicted lean periods leads to substantial UE power savings ranging from 80% to 97% compared to manual upgrades.

INDEX TERMS 5G core, cloud native, control plane, user plane, CNF, AI/ML, SARIMA, power optimization.

I. INTRODUCTION

The advent of telco cloud has ushered in a new era of network architecture, integrating software-defined networking, virtualization, and Cloud Native (CN) technology to establish a distributed computing network. Telco infrastructure deployed within a CN environment provides businesses with the opportunity to upgrade and scale operations strategically, maximizing growth potential while minimizing costs. The inherent elasticity of the cloud enables dynamic upgrade and scaling of telco products, facilitating the accommodation of faster time-to-market, high-performance workloads and traffic spikes. To fully capitalize on the benefits of containerization within a CN architecture, it is imperative to align product technologies with the characteristics of distributed microservices [1].

Leading vendors in the 5G and beyond landscape have commenced redesigning their telco workload Network Functions (NFs) in the access and core domains to adhere to the CN and virtualization paradigm [2], [3], [4]. The evolution of telecommunication networks towards virtualization, containerization, and software-defined architectures, as envisioned by the 3rd Generation Partnership Project (3GPP), presents both challenges and opportunities for

The associate editor coordinating the review of this manuscript and approving it for publication was Miguel López-Benítez.

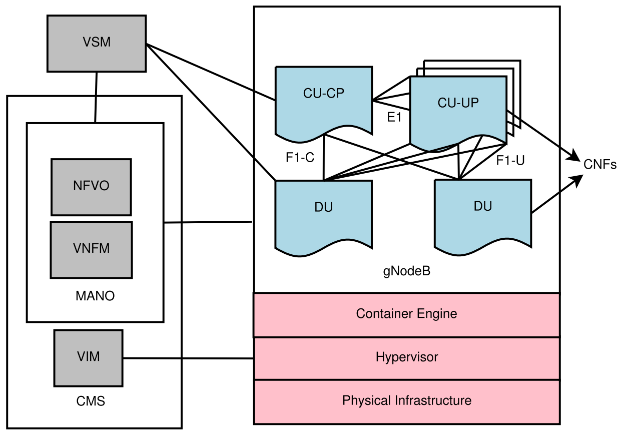

FIGURE 1. **CNF (vRAN) deployment in a NFV environment.**

network management and optimization. A significant aspect of this evolution involves the migration of Control Plane (CP) block functions to software entities, enhancing flexibility and efficiency in network operations. To meet the demand for network capacity [5], 3GPP [6] has proposed dividing the base station, gNodeB (gNB), into two distinct units - the Central Unit (CU) and the Distributed Unit (DU).

The CP and User Plane (UP) blocks oversee signaling and data traffic, respectively, and are implemented as Cloud-native Network Functions (CNFs) within a Network Function Virtualization (NFV) environment for virtual Radio Access Network (vRAN) deployment, as depicted in Figure 1. The CP and UP comprehended the CU, while only DU encamps the UP. Dynamic upgrade and scaling are essential, and brings in elasticity to the NFV environment. When notified of a software upgrade by the Virtualized System Manager, the CP block necessitates temporary downtime, prompting a re-attach procedure for all User Equipment (UEs) connected to the upgrading CP. This interruption results in data service disruptions and unnecessary power consumption during re-attach procedures.

The traditional approaches to CP maintenance often rely on manual intervention by operators, resulting in suboptimal upgrade timings and unnecessary resource wastage. To address these challenges, there is a pressing need for automated procedures that can determine the optimal timing for CP upgrades, aligning with periods of low network traffic and minimizing the impact on UE connectivity and power consumption. In our previous study [7], a CP Outage Handling Procedure (COHP) was proposed to mitigate these issues by preserving UE contexts during CP upgrades. The COHP procedure, while effective for Best Effort (BE) services, still leaves UEs vulnerable to service interruptions and power wastage if the upgrade fails or takes longer than the predetermined stale timer duration. Consequently, there is a need for a predictive procedure to optimize CP upgrade timing, minimizing the impact on UEs.

In this paper, we introduce an AI/ML-based approach, dubbed COHP version 2 (COHPv2), to address the limitations of the existing COHP process by accurately determining the optimal CP software upgrade time. COHPv2 leverages AI/ML techniques to analyze traffic behavior, including seasonality changes, to forecast the optimal lean period for each CP independently. Our approach, i.e. COHPv2 additionally leverages statistical data from multiple CP blocks managing base stations across diverse geographical areas, including residential and non-residential zones such as subways and hospitals. By employing SARIMA forecasting techniques, we predict bearer statistical data to calculate the optimal CP software upgrade time, then we validated our proposal through rigorous statistical analysis. Our key contributions to COHPv2 are as follows,

- Proposal of a three-step automated procedure for CP software upgrade management, integrating AI/ML forecasting and dynamic threshold validation.
- Utilization of SARIMA model for lean period prediction and validation, achieving high accuracy with an R-Squared score of approximately 0.99.
- Implementation of dynamic Z-Score threshold to handle seasonality changes and ensure precise scheduling of CP software upgrades.
- Generation of analytics reports containing lean period information, enabling informed decision-making by NFV Orchestrator (NFVO) and facilitating automated upgrade triggering based on operator preferences.
- Achievement of significant UE power savings of 80-97% using the proposed AI/ML-based procedure in conjunction with COHPv2.

The organization of this paper is as follows: Section II reviews related work on network management and operations, with a focus on AI/ML techniques. Section III introduces the network elements and time series model. Section IV-A describes the challenges posed by manual CP software upgrades and the need for a dynamic network. The time series approach for determining the software upgrade window is detailed in Section IV-B. Experimental results and evaluations are presented in Section V, while the conclusions and the benefits of the proposed software upgrade approach are discussed in the Section VI.

II. RELATED WORK

In this section, we delve into recent advancements concerning analytics in network management, insights from exhaustive surveys [8], [9], [10], [11], [12]. We started with analyzing existing works focusing on 3GPP standards. The 3GPP's technical specification 28.104 sets the groundwork for applying data analytics and its functionalities in 5G architecture through Management Data Analytics (MDA) [8]. With the evolving landscape of 5G services, proactive management of network faults and performance issues becomes imperative. MDA Service (MDAS) harnesses real-time operational intelligence gathered by the MDA Function (MDAF) from managed NFs, streamlining network automation and service orchestration. MDAF, acting as an MDAS producer, furnishes data to corresponding consumers, incorporating internal business logic to exploit performance management, alarm information, and analytics data. MDAS analyzes the collected data to predict potential issues, such as service disruption or degradation.

Prior studies predominantly rely on linear forecasting algorithms to anticipate network traffic patterns and optimize network services for enhanced user experiences. Researchers, such as those cited in [9], delve into the analysis of traffic patterns across millions of cellular base stations, uncovering significant variations in traffic distribution throughout the day. Further explorations into cellular network traffic distributions reveal dynamic variations over time periods. Studies by Wang et al. [10] demonstrate the trimodal distribution [11] of cellular traffic on both spatial and temporal dimensions, while Lee et al. [12] identify log-normal or Weibull distribution as best approximations for traffic density in the spatial domain. While existing approaches establish periodicity for regular traffic patterns, they exhibit limitations in deriving the duration of traffic distributions necessary for network planning activities, such as software upgrades and load balancing.

Thus, there exists a pattern in the cellular traffic data, and we need a mechanism to determine the lean period for network planning activities such as software upgrades. The authors [13] suggest a multi-model fusion prediction method combining feature selection and stacking ensemble learning to predict base station traffic. However, they do not delve into detailed traffic data type collected from thousands of base stations. In contrast, COHPv2 utilizes standard-based Performance Monitoring (PM) traffic data to study base station traffic, ensuring our solution is compatible with any telecommunications vendor. We investigate peak and non-peak traffic distribution to derive the frequency and duration of the lean period (the time duration when traffic demands are generally less over a day). The operator can perform network planning activities during such lean periods, so that the impact of such activities, as explained in Section I, can be minimized. The novel contributions of this work are as follows, a) introduction of a CP software upgrade automation procedure to identify the potential lean periods for upgrade activities, b) examination of the traffic characteristics across CP blocks, considering key bearer features, c) development of a dynamic threshold algorithm to adapt to changes in UP traffic time series patterns, enhancing adaptability and accuracy in the network management.

III. SYSTEM MODEL

In the context of network management, the NFVO plays a pivotal role in orchestrating the lifecycle of virtualized NFs and services. NFVO serves as the central entity responsible for coordinating various network management tasks, including resource allocation, service provisioning, and lifecycle management. In the context of MDAS, NFVO acts as a key consumer of the real-time operational intelligence provided by the MDAS. By leveraging the insights derived from MDAS, NFVO gains valuable information about network performance, faults, and trends, enabling it to make informed decisions regarding network optimization, automation, and service orchestration. This close relationship between NFVO and MDAS ensures seamless integration of analytics-driven insights into network management processes, ultimately enhancing the efficiency, reliability, and performance of virtualized telecommunication networks.

A. TELCO NETWORK MODEL

In this paper, we have collected data from one of the live 5G cellular network operators. We have considered an area of 10 km sq. within a densely populated city. This area consists of 500 base stations (residential as well as non-residential) with a transmission range of 100m approximately. All these base stations are connected to 20 DUs which in turn are connected to 7 CPs as shown in Figure 2.

Further, as discussed in Section I, MDAF is used as a traffic analyzer to analyze the bearer data. This data is consolidated by CP blocks from their associated UP blocks. Let CP $C_i \in \mathcal{C}$, where $\mathcal{C}$ denotes the set of CP blocks. MDAS consumer (NFVO) requests MDAF (managing C_i) to identify the best lean period for C_i to perform software upgrade. MDAF then collects and analyzes data from C_i to generate the MDA Report. MDAF contains AI/ML models and Z-Score statistical method [14] to analyze the PM statistics of bearer. The Z-Score statistical analysis on bearer data sets up a threshold value $Z\_threshold$ to validate the predicted lean period. At last, MDAF generates the MDA report containing lean period information and sends to NFVO. NFVO, which acts as an MDAS consumer, triggers the software upgrade based on the operator's decision. To cater to the daily trends, Periodic Scalar Update (PSU) function updates the $Z\_threshold$ periodically. Further, AI/ML models are retrained periodically for better performance. These are further explained in Section IV-B. Table 2 shows the list of notations used in this study.

B. TIME SERIES MODELLING WITH PM DATASET

As mentioned previously, we have gathered PM data from both residential and non-residential base stations, comprising various features related to bearers as outlined in Table 1. To manage processing load effectively at the management server (MDAF), we have adopted a data collection interval of 15 minutes. Ensuring the dataset's suitability for time series modeling necessitates verifying its stationarity [15]. In the literature [15], the Augmented Dickey-Fuller (ADF) Statistical test emerges as a commonly employed method for assessing stationarity in time series datasets. The ADF test evaluates the null hypothesis concerning the presence of a unit root within the time series dataset. This is expressed mathematically in the equation 1.

$$z_t = c + \beta t + \alpha z_{t-1} + \sum_{n=1}^{p} \varphi_n \Delta z_{t-n} + \epsilon_t \tag{1}$$

where, z_t is the value of the time series at time t, z_{t-1} is the first lag of the model at time $t-1$, α is the coefficient of the

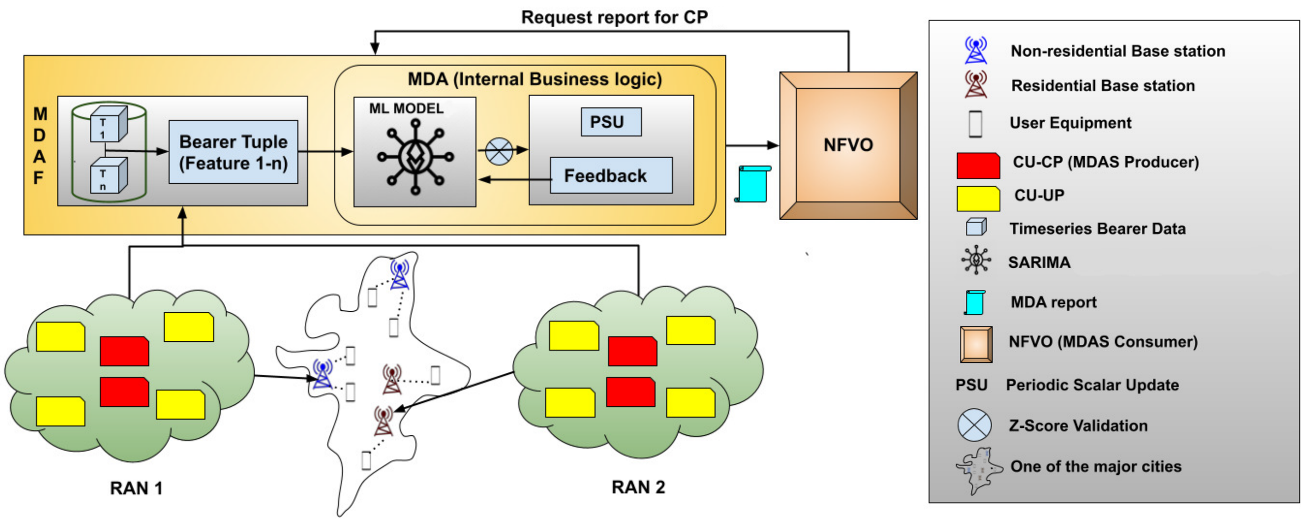

FIGURE 2. Network model.

TABLE 1. Key bearer features influencing CP SW upgrade procedure.

Bearer Features (count)	Traffic analyzer impacts on software upgrade
Active Bearer	Traffic analyzer shall forecast time when the number of active bearer sessions are minimal.
Handover	Traffic analyzer shall forecast time when the less number of bearer sessions are susceptible for handover.
Bearer Modify	Traffic analyzer shall forecast time when the less number of bearer sessions are susceptible for modifications.
Bearer Release	Traffic analyzer shall forecast time when the number of bearer sessions that are susceptible for release are low.

TABLE 2. List of notations.

Symbol	Description
$\mathcal{C}$	Set of CP blocks
C_i	Uniquely identifies the CP with index 'i'
t	Time
$Z\_threshold$	Lean period threshold
x_t	Time series data at time t
w_t	Gaussian noise
p	Number of lagged observations in time series analysis
q	Number of differences used in time series analysis
d	Number of lags of forecast errors in time series analysis
s	Seasonal period
Opt_t	Set of predicted optimal CP software upgrade times

first lag in the time series data, Δz_{t-n} is the first difference of the series at $t-n$, ϵ_t is the noise generated, β is the weighted average at time t, φ_n is the weighted average to the difference at time $t-n$, p is the order of the model and c is the constant. The null hypothesis assumes the presence of unit root (i.e., $\alpha = 1$), and thus, the p-value must be less than a significance level (0.05) [15] to reject the null hypothesis and prove that the dataset is stationary. From the ADF test on bearer dataset, using python statistical models library, the derived p-value is $2.53e^{-7}$. This value is less than the significance level which proves that the time series dataset is stationary. Hence, forecasting models can be applied to the dataset without any differencing terms.

C. CORRELATION ANALYSIS OF BEARER FEATURES

We have collected PM data for key bearer features as depicted in Table 1 for a complete week. Next, we have performed correlation analysis to find the Pearson Coefficient of correlation [15] among different bearer features. The correlation matrix in Figure 3 shows that the least correlation value is 0.85. This value proves that all the features taken into account are highly correlated with each other. Hence, in our work, we have considered one of the features, active bearer for forecasting. In Table 1, we have shown four different types of bearer features, such as, active bearer, handover, bearer modify and bearer release. We have also shown the traffic analyzer impacts on software upgrade under these bearer conditions.

IV. PROBLEM STATEMENT AND COHPV2 SOLUTION

A. UNDERSTANDING NETWORK DYNAMICS AND UPGRADE CHALLENGES

The cellular network is dynamically growing to meet the service demand. A dynamic next-generation network that

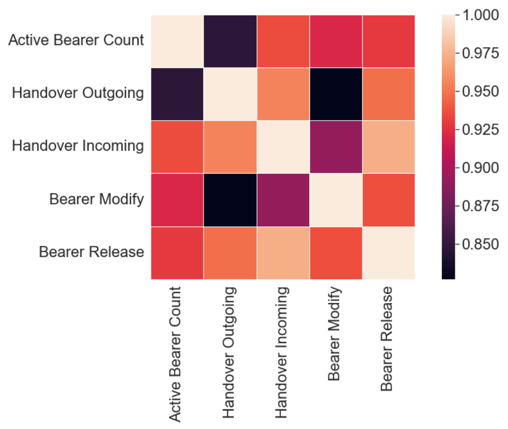

FIGURE 3. **Pearson coefficient analysis on bearer features.**

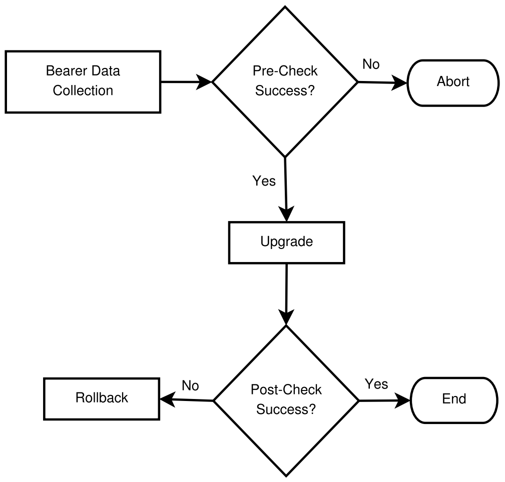

FIGURE 4. **A Typical CU-CP software upgrade cycle.**

supports efficient, robust, and agile management needs to be in place. These robust and dynamic cellular network need to undergo many cycles of software upgrades. However, these upgrades may result in service disruption. Thus, we need to determine an optimal time to perform software upgrade. The time period (lean period) with relatively lesser number of active bearers is the optimal time to perform software upgrade. Generally, lean periods are determined using manual approximation by cellular operator. However, in some cases such as soccer events during midnight and pandemic situations, there is a chance of having more number of active bearer connections during the manually determined lean periods. In such cases, these manually determined lean periods may not be the optimal time to perform software upgrade. Therefore, there is a need for an AI/ML-based procedure which forecasts the accurate lean period to trigger the software upgrade.

Typically, upgrade activity spans entire night period (considering night period as lean period) with a series of validation steps for pre- and post-upgrade. These steps will measure the network performance based on parameters such as service availability, accessibility, retainability, performance, etc. In case of any performance degrade, software upgrade will be rolled back. This is explained in Figure 4. The operator uses a set of Key Performance Indicators (KPIs) to compare pre- and post-upgrade conditions to declare the success or failure of an upgrade. The entire procedure is automated to reduce the manual intervention and improve the robustness of the software upgrade.

B. COHPV2 SOLUTION APPROACH

To solve the above problem, we propose a three-step COHPv2 procedure that automates the CP software upgrade procedure, which is given as follows:

(a) Leveraging AI/ML for Lean Period Forecasting
(b) Enhancing Prediction Accuracy With Dynamic Z-Score Threshold Validation
(c) Streamlining MDA Reporting and NFVO Interaction
(d) Validating Software Upgrade Success With Automated Metrics

The above mentioned COHPv2 solution is described in Algorithms 1 and 2. Algorithm 1 describes the steps involved in CP software upgrade procedure. As stated in Section I, to minimize the impact of the software upgrade, NFVO needs to determine the lean period of CP C_i that needs to undergo software upgrade. Thus, NFVO requests MDAF to provide an MDA report containing the lean period information. To determine the lean period, MDAF first collects the bearer statistical data (Table 1) in *br_info* from C_i (step 2). Next, MDAF calls *Analyze* procedure (Algorithm 2) to analyze the collected *br_info* (step 3). *Analyze* procedure returns a set of optimal upgrade times Opt_t and predicted bearer statistical data *br_pred* during Opt_t for C_i. MDAF shares the MDA report containing these parameters with NFVO (step 4).

On receiving the MDA report, NFVO sends feedback containing operator's decision op_d and selected optimal upgrade time $up_{time} \in Opt_t$. If the operator decides to opt for different lean period, then it sends op_d as negative. The model will further look for the next optimal time to upgrade (steps 5-7). Else if op_d is positive, the algorithm monitors real-time C_i data to check for any abnormal behavior until the current time t_{c_i} reaches the predicted lean period up_{time}. If there is an abnormal behavior in the bearer data, the algorithm warns NFVO not to perform software upgrade (steps 9-17). The Z-Score threshold is updated using *threshold_Update* function if the upgrade is success (steps 20-22). Algorithm 1 also describes the *PeriodicUpdate* procedure (steps 24-27) which updates the *Z_threshold* and retrains the AI/ML models for all $C_i \in C$. This procedure helps to cater to the seasonality changes in the *br_info* dataset.

Algorithm 2 describes the procedure to analyze the bearer statistical data (Algorithm 1, step 3), predict the lean period, and generate the MDA report for C_i. To analyze the bearer statistical data, it is fed into AI/ML SARIMA model [16] (step 2). The working of SARIMA is described in Subsection IV-B1. This model predicts a set of lean periods. The predicted lean periods are validated using *Z_threshold* which is retrieved using *get_Threshold* function (step 3).

1) LEVERAGING AI/ML FOR LEAN PERIOD FORECASTING

In the section III-B, we have proved that the time series bearer dataset is stationary. Thus, we propose to use SARIMA forecasting model [16] for each C_i. SARIMA performs better on long forecast intervals compared to other models [17].

Algorithm 1 NFVO-MDAF Lean Period Prediction

```
1:  procedure Predict($C_i$)
2:      $br\_info \leftarrow collect\_Info(C_i)$
3:      $Opt_t, br\_pred, update\_flag \leftarrow analyze(br\_info)$
4:      $up_{time}, op_d \leftarrow send\_Report(C_i, Opt_t, br\_pred)$
5:      if $op_d == NEGATIVE$ then
6:          $valid(op_d) = INVALID$
7:          goto Step 2
8:      else
9:          $t_{c_i} \leftarrow current\_timestamp$
10:         while $t_{c_i} < up_{time}$ do
11:             $br\_data \leftarrow live\_DataCollection(C_i)$
12:             $pattern \leftarrow monitor(br\_data, br\_pred)$
13:             $wait$(15 minutes)
14:             $t_{c_i} \leftarrow current\_timestamp$
15:         end while
16:         if $pattern == ABNORMAL\_INCREASE$ then
17:             $warn$(“Do not upgrade”, pattern)
18:         end if
19:     end if
20:     if $update\_flag == TRUE$ then
21:         $threshold\_Update(br\_data)$
22:     end if
23: end procedure

24: procedure PeriodicUpdate
25:     $retrain\_All(C)$
26:     $threshold\_Update\_All(C)$
27: end procedure
```

Algorithm 2 Bearer Traffic Analysis

```
1:  procedure Analyze($br\_info$)
2:      $Opt_t, br\_pred \leftarrow ml\_Prediction(br\_info)$
3:      $Z\_threshold \leftarrow get\_Threshold()$
4:      if $find\_ZScore(br\_pred, Opt_t) <= Z\_threshold$ then
5:          $update\_flag \leftarrow TRUE$
6:      else
7:          $update\_flag \leftarrow FALSE$
8:      end if
9:      return $Opt_t, br\_pred, update\_flag$
10: end procedure
```

SARIMA(p, d, q, s) model is composed of four components,

(a) AR component: It models the relationship between the time series and lagged values. p represents the number of lagged observations.
(b) Integrated component: It uses differencing terms to make the time series stationary. d represents the number of differences used.
(c) MA component: It predicts the future values as function of lagged errors in forecasting. q represents the size of the MA window (number of lags of forecast errors).
(d) Seasonal component: It represents the time window used in forecasting and is denoted by s.

The general forecasting equation of SARIMA(p, d, q, s) is written as,

$$\Phi_p\left(B^s\right)\phi(B)\left(\nabla_s\right)^d\nabla^d x_t = \Theta_q\left(B^s\right)\theta(B)w_t \qquad (2)$$

where, $\Phi_p\left(B^s\right)$ and $\Theta_q\left(B^s\right)$ are of orders p and q, and represent the seasonal AR and MA components, respectively. B is the backshift operator. The polynomials $\phi(B)$ and $\theta(B)$ represent the normal AR and MA components, respectively. ∇^d and $(\nabla_s)^d$ represent the normal and seasonal difference components. x_t represents the time series data, and w_t is the Gaussian noise. The order p of the AR model is found using the Partial Auto-Correlation Function (PACF) plot [15], whereas, Auto-Correlation Function (ACF) plot [15] is used to find the order q of the MA model. Orders p and q are the lag values when PACF and ACF plots cross the confidence interval [15] for the first time. After analyzing the data, we have observed that the values of p and q are (1, 1). As the time series bearer dataset is proved stationary in Subsection III-B, the number of differences d is 0. Seasonal period s is calculated to be 96 (data taken every 15 minutes in a day). Thus, the equation 2 can be written as,

$$\Phi_1\left(B^{96}\right)\phi(B)x_t = \Theta_1\left(B^{96}\right)\theta(B)w_t \qquad (3)$$

With the derivation of polynomials Φ_p, Θ_q, ϕ, and θ [15], the equation 3 becomes,

$$x_t = 1 + \Theta_1\left(B^{96}\right) + \Theta_1\theta\left(B^{97}\right) - x_t\Phi_1\left(B^{96}\right) + \theta(B) + x_t\phi(B) - x_t\Phi_1\phi(B)\left(B^{97}\right) + w_t \qquad (4)$$

Thus, the SARIMA forecasting equation 4 predicts the active bearer count *br_pred* which is used to derive the optimal lean period Opt_t (Algorithm 2, step 2) for C_i. This model is retrained periodically using *retrain_All* function to avoid any model drifts (Algorithm 1, step 25).

2) ENHANCING PREDICTION ACCURACY WITH DYNAMIC Z-SCORE THRESHOLD VALIDATION

To build a robust lean period prediction, we need to address the uncommon and rarest of situations. One such example is hosting soccer games. Typically, the soccer events are seasonal and are played in floodlights, resulting in the millions of users connecting to the midnight telecast [18]. The predicted lean period can vary with the actual lean period during this midnight telecast. These uncommon situations demand a periodic and dynamic threshold value to calculate the best lean period. Thus, in this paper, we propose to use Z-Score threshold validation [14] to validate the predicted lean period. With Z-Score threshold validation, the proposed lean period prediction procedure brings robustness to the upgrade schedule (i.e., retraining the models during such

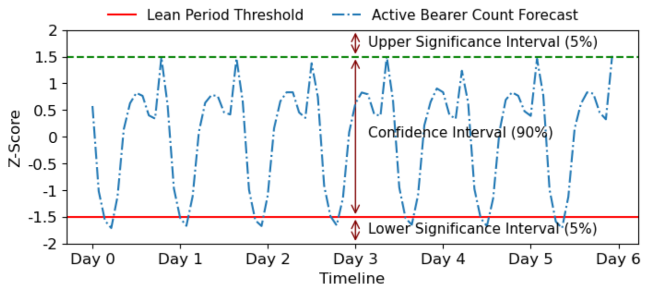

FIGURE 5. **Lean period threshold calculation.**

aforementioned situations, thus less impact on millions of viewers).

The Z-Score analysis [14] on the active bearer count shows how far the datapoint varies from the mean value (μ) calculated on the bearer training data (i.e., the number of standard deviations (σ) lying above or below the mean value). The value of Z-Score Z can be calculated using the equation 5.

$$Z = (X - \mu)/\sigma \tag{5}$$

where X is the data point.

From the Figure 5, we have noticed that σ lies between -2 to $+2$. To validate the predicted lean period, the lean period threshold $Z\_threshold$ is calculated based on the Z-Score of the significance level $Z_{\alpha/2}$ [14] with standard confidence intervals such as 90% and 95%. In this paper, we have calculated the $Z\_threshold$ with 90% as confidence interval and 10% as α level (Figure 5). The other standard significance levels [14], such as 95% confidence interval, exceed the minimum Z-Score of active bearer count. The $Z\_threshold$ is calculated using the equation 6.

$$Z\_threshold = Z_{\alpha/2} \tag{6}$$

The best time to perform software upgrade is the time when the Z-Score of active bearer count is less than Z-Score threshold, as they are far away from μ. Let $Z_t^{br\_pred}$ denotes the Z-Score for predicted active bearer count at time t. Time t is predicted as lean period if the criteria mentioned in the equation 7 is met.

$$Z_t^{br\_pred} <= Z\_threshold \tag{7}$$

From the Figure 5, we infer that the lean period threshold $Z\_threshold$ is initially set as -1.5 (Algorithm 2, step 3). The algorithm uses $Z\_threshold$ to validate the predicted lean period Opt_t. If the Z-Score of br_pred during Opt_t is not less than the $Z\_threshold$, $update\_flag$ is set to true. This flag will update $Z\_threshold$ only if the upgrade is success (steps 4-8). The lean period window size can be changed by modifying $Z\_threshold$. The scalars (μ and σ) are updated periodically using $threshold\_Update\_All$ function to accommodate the abnormal trends for short period and the seasonal changes (Algorithm 1, step 26).

TABLE 3. **MDA report format.**

Parameters	Description
Timestamp	Time when the report is generated
C_i	Uniquely identifies the CU-CP
Opt_t	Set of possible times at which the CP software upgrade can take place
br_pred	Set of active bearer count mapped to the set of predicted upgrade times

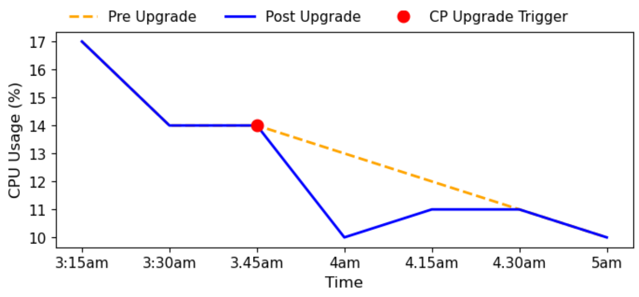

FIGURE 6. **CPU usage comparison of cp across pre- and post-upgrade.**

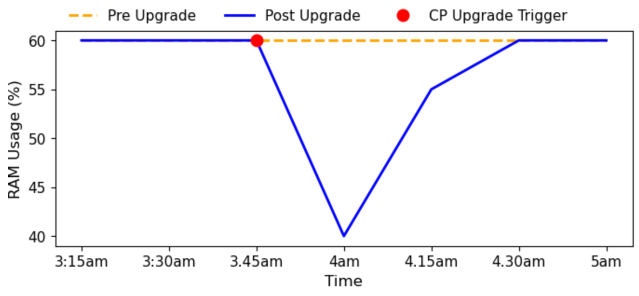

FIGURE 7. **RAM usage comparison of cp across pre- and post-upgrade.**

3) STREAMLINING MDA REPORTING AND NFVO INTERACTION

After the lean period prediction, MDAF generates the MDA report in format of $\{C_i, Opt_t, br\_pred\}$, as given in Table 3 and sends to NFVO using $send\_Report$ function (Algorithm 1, step 4). The operator decides whether the CP software upgrade has to be executed based on the MDA report. The operator may also deploy his strategy to override MDA report to suitably trigger the upgrade. For example, to avoid complete blackout in an area where all cells are connected to various C_is and are undergoing software upgrades concurrently. In such case, the NFVO sends negative feedback to MDAF. Further, MDAF sets the previously predicted Opt_t as invalid and predicts the new optimal upgrade time (steps 5-7).

4) VALIDATING SOFTWARE UPGRADE SUCCESS WITH AUTOMATED METRICS

Apart from identifying the best time to perform software upgrade with AI/ML-based procedure, we also propose to extend automation for software upgrade validation. The

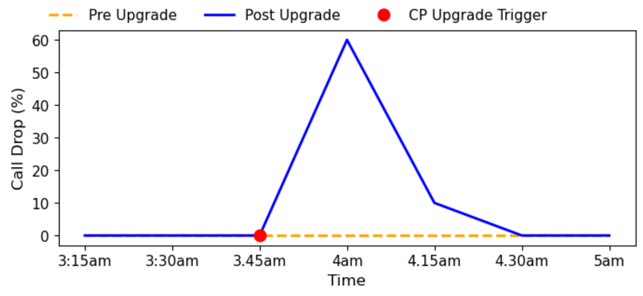

FIGURE 8. Call drop comparison of cp across pre- and post-upgrade.

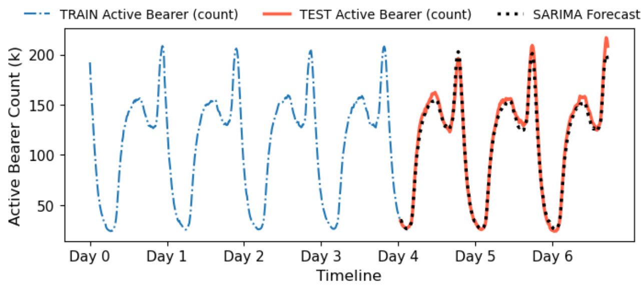

FIGURE 9. Forecasting the active bearer count of CP.

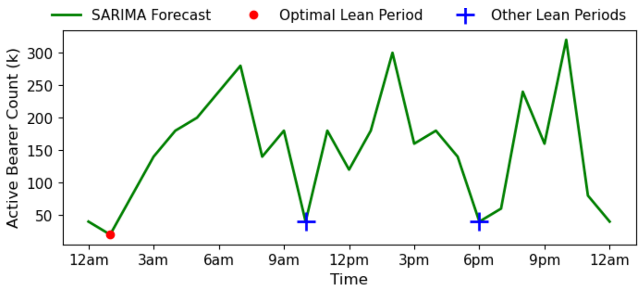

FIGURE 10. Active bearer count forecast of CP (Public Transport).

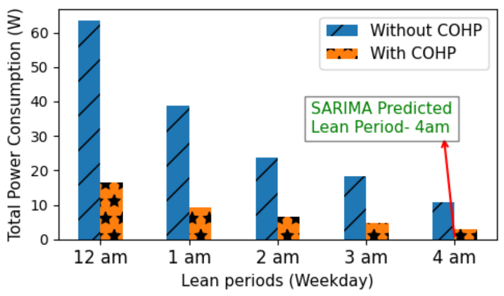

FIGURE 11. Total UE power consumption during CP upgrade on weekday.

network periodically reports the network statistics such as faults, performance, etc. Generally, the stability of software upgrade is measured as a factor of system stability and KPIs performance. System stability involves checking on computation resources of the CP, such as compute and RAM, across the software upgrade as shown in Figures 6 and 7. In these figures, the pre- and post-upgrade graphs are almost consistent except for the thirty minutes between 03:45 to 04:15 AM. This difference is attributed to software upgrade because of which graph shows a drop in CP resource usage.

We have considered call drop statistics KPI to validate the CP software upgrade. Call drop statistics are calculated for the successfully connected calls that are abnormally dropped due to gNodeB or mobility management function failure. Figure 8 shows that the call drop percentage increases when the software upgrade is triggered at 03:45 AM and eventually became normal around 04:15 to 04:30 AM, within the monitoring window. However erstwhile, the call drop event graph shows the same behavior across the software upgrade. This concludes a successful software upgrade.

V. PERFORMANCE EVALUATION

Our experimental setup is designed to operate within a Kubernetes cluster, leveraging containerized microservices for efficient and scalable processing. To store PM data, we have utilized Prometheus [20] as our time-series database, deployed on Kubernetes using Helm charts. Data from Prometheus is streamed to Apache Kafka [21], also deployed on the Kubernetes cluster to facilitate better scalability and flexibility. We have created custom Kubeflow Pipeline components [22] to handle data incoming from Kafka, preprocess the data, and train the SARIMA model using pmdarima 2.0.4 python library. The trained model is subsequently logged and stored in MLflow [23], which is also deployed within the Kubernetes environment for comprehensive model management.

Figure 9 shows the train and test data along with the SARIMA predictions of active bearer count. We have calculated the normalized Root Mean Squared Error (nRMSE) and R-Squared error (R2E) score [19]. This model has shown good accuracy of prediction as nRMSE is 0.025 (closer to 0) and R2E is 0.99 (closer to 1).

Our proposed AI/ML model is specific to each C_i as the bearer statistical data pattern varies for each C_i. The pattern depends on the area where base stations are connected to C_is. For example, Figure 10 shows the active bearer count forecast of C_i which has consolidated bearer data from the base stations deployed in the public transport sector. It shows that this C_i has clusters of lean periods in a day whereas, in Figure 9, there exists only one lean period for that C_i in a day.

Figures 11 and 12 show the variations of average total power consumed by UEs to restore the CP connection during weekday and holiday, respectively. This depicts the difference in power consumption made during manual upgrades and AI/ML-based automatic CP software upgrade. We have shown in our previous work [7] that the proposed COHP optimizes the power consumption by UEs to restore the CP connections. From the Figures, we can infer that 4 AM and 3 AM are the predicted optimal lean periods during weekday and holiday, respectively. We can also infer that by performing CP software upgrade during the predicted optimal lean period, we can save approximately 72-75% of total power consumption without deploying COHP (comparing 12 AM with 4 AM and 3 AM data during weekday and

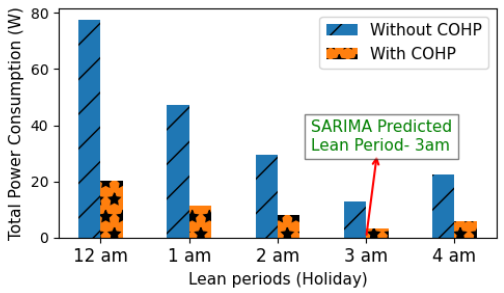

FIGURE 12. Total UE power consumption during CP upgrade on holiday.

holiday, respectively). With COHPv2 (i.e., COHP deployment along with optimal lean period prediction), we can save approximately 80-97% of total power consumption when CP software upgrade is performed at the predicted optimal lean period.

VI. CONCLUSION

In this study, we introduced COHPv2, an automated procedure designed to streamline the software upgrade process for CP functions within a NFV environment. By harnessing the power of AI/ML, COHPv2 analyzes bearer statistical traffic patterns to identify optimal lean periods for scheduling CP software upgrades, thereby minimizing disruptions to users. Through extensive experimentation, we demonstrated the effectiveness of COHPv2 in accurately predicting lean periods with a high level of precision, as evidenced by the SARIMA forecasting model's R-Squared score of approximately 0.99. Furthermore, we validated the forecasted lean periods using a dynamic Z-Score threshold, ensuring adaptability to seasonal variations in network traffic. Our proposed approach involves the generation of MDA reports containing lean period information, which are then utilized by the NFVO to trigger the CP software upgrade procedure based on operator feedback. The successful implementation of COHPv2 not only reduces the manual operational overhead associated with CP software upgrades but also achieves significant power savings for UEs, ranging from 80-97%. This work underscores the potential of automation and intelligence in revolutionizing cellular network management and orchestration, paving the way for more efficient and resilient telecommunication infrastructure. In the future, we plan to explore COHPv2 to ensure uninterrupted service for 5G Ultra Reliability and Low Latency (URLLC) services. Delay-critical Guaranteed Bit Rate (GBR) [6] is a specific type of bearer service designed in 5G systems to support this URLLC traffic. By incorporating Delay-critical GBR data into our predictive time series model, we can enhance COHPv2 to identify optimal CP software upgrade windows and avoid service disruption of 5G URLLC services.

REFERENCES

[1] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*. Cham, Switzerland: Springer, 2017, pp. 195–216, doi: 10.1007/978-3-319-67425-4_12.

[2] Cloud Native 5G Core. (2021). *Samsung 5G Core White Paper*. [Online]. Available: https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/cloud-native-5g-core/Cloud-Native-5G-Core-Samsung-5G-Core-Volume-2.pdf

[3] *HPE Technical White Paper, Container Platform for Telcos*. Accessed: May, 26, 2021. [Online]. Available: https://assets.ext.hpe.com/is/content/hpedam/a50003043enw

[4] Nokia. *Blog Containers and the Evolving 5G Cloud-Native Journey*. Accessed: May, 26, 2021. [Online]. Available: https://www.nokia.com/blog/containers-and-the-evolving-5g-cloud-native-journey

[5] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Upgrade, 2015–2020*, Cisco, San Jose, CA, USA, Feb. 2016.

[6] *System Architecture for the 5G System (5GS); (Stage 2)*, document 23.501, Release 19, 3GPP Tech. Specification, Jun. 2024.

[7] N. Kumar, K. Subramaniam, K. Narayanan, and N. Kommineni, "A novel power efficient CP block selection procedure for delay- tolerant services to handle CP outages in 5G and beyond," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2021, pp. 1–4.

[8] *Technical Specification Group Services and System Aspects; Management and Orchestration; Management Data Analytics (MDA)*, document 28.104, Version 0.3.0, Release 18, 3GPP Tech. Specification, Jun. 2023.

[9] Y. Zhang and A. Årvidsson, "Understanding the characteristics of cellular data traffic," in *Proc. ACM SIGCOMM Workshop Cellular Networks: Operations, Challenges, Future Design*, Aug. 2012, pp. 461–466.

[10] H. Wang, J. Ding, Y. Li, P. Hui, J. Yuan, and D. Jin, "Characterizing the spatio-temporal inhomogeneity of mobile traffic in large-scale cellular data networks," in *Proc. 7th Int. Workshop Hot Topics Planet-Scale Mobile Comput. Online Social Netw.*, Jun. 2015, pp. 19–24.

[11] R. Vila, V. Serra, M. N. Çankaya, and F. Quintino, "A general class of trimodal distributions: Properties and inference," *J. Appl. Statist.*, vol. 51, no. 8, pp. 1446–1469, Jun. 2024, doi: 10.1080/02664763.2023.2207785.

[12] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang, "Spatial modeling of the traffic density in cellular networks," *IEEE Wireless Commun.*, vol. 21, no. 1, pp. 80–88, Feb. 2014.

[13] L. Zhao, Y. Huang, Y. Wang, Y. Xu, Q. Feng, and E. Chen, "Base station traffic prediction based on feature selection and stacking ensemble learning," in *Proc. 4th Int. Conf. Comput., Netw. Internet Things*, May 2023, pp. 113–117, doi: 10.1145/3603781.3603800.

[14] L. Gonick, *The Cartoon Guide to Statistics*. New York, NY, USA: Harper Perennial Press, 1993.

[15] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1994, doi: 10.1515/9780691218632.

[16] NCSS Documentation. (2024). *NCSS Statistical Software Exponential Smoothing and ARIMA*. Accessed: Jul. 18, 2024. [Online]. Available: https://www.ncss.com/software/ncss/ncss-documentation/#TimeSeries

[17] G. R. A. Brito, A. R. Villaverde, A. L. Quan, and M. E. R. Pérez, "Comparison between SARIMA and holt–winters models for forecasting monthly streamflow in the western region of Cuba," *Social Netw. Appl. Sci.*, vol. 3, no. 6, Jun. 2021, Art. no. 671, doi: 10.1007/s42452-021-04667-5.

[18] MediaBrief. (2024). *Sony LIV Reports 50% User Increase, 64% More Views, 350% Rise in Live Views for UEFA EURO 2024*. [Online]. Available: https://mediabrief.com/sony-liv-reports-350-rise-in-live-views-for-uefa-euro24

[19] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," 2018, *arXiv:1809.03006*.

[20] M. Y. E. Saputra, N. Noprianto, S. N. Arief, V. N. Wijayaningrum, and Y. W. Syaifudin, "Real-time server monitoring and notification system with Prometheus, Grafana, and telegram integration," in *Proc. ASU Int. Conf. Emerg. Technol. Sustainability Intell. Syst. (ICETSIS)*, Jan. 2024, pp. 1808–1813, doi: 10.1109/icetsis61505.2024.10459488.

[21] B. R. Hiraman, M. C. Viresh, and C. K. Abhijeet, "A study of apache Kafka in big data stream processing," in *Proc. Int. Conf. Inf., Commun., Eng. Technol. (ICICET)*, Aug. 2018, pp. 1–3, doi: 10.1109/ICICET.2018.8533771.

[22] W.-K. Wong, W.-C. Lin, and Y.-B. Chen, "Research on kubeflow distributed machine learning," in *Proc. IET Int. Conf. Eng. Technol. Appl. (IET-ICETA)*, Oct. 2022, pp. 1–2, doi: 10.1109/IET-ICETA56553.2022.9971554.

[23] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, F. Xie, and C. Zumar, "Accelerating the machine learning lifecycle with MLflow," *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, Apr. 2018.

KARTHIKEYAN SUBRAMANIAM has been working in telecommunications systems for more than two decades. He is currently the Business Head of the Management SW Group, Samsung Research and Development Institute, Bengaluru, India, where he oversees more than 15 global operators. In addition, he contributes to improving operational efficiency for solution deployments, which are also published in 3GPP SA5 standards. Moreover, he participates in IEEE publications and has served as the session chair. His main focus has been in research and development of AI/ML-based solutions for managing next-generation telecom systems. His research interests include network management and orchestration, network slicing, cloud and container-native technologies, and software-defined networking.

NAVEEN KUMAR received the master's degree in computer science and engineering from Indian Institute of Technology Madras, India. He is currently a Chief Engineer with the Samsung Research and Development Institute, Bengaluru, India. His main research interests include device-to-device communication, mm-wave-based communication, and next-generation cellular networks with a focus on throughput maximization and QoS enhancement.

SUDHAKAR BALUSAMY received the bachelor's degree in computer science and engineering from the College of Engineering, Guindy, Chennai, India, in 2020. He is currently a Lead Engineer with the Samsung Research and Development Institute, Bengaluru, India. He has been part of the cloud platform/infrastructure development and operation team. His main research interests include cloud computing, machine learning, and software-defined networks.

CHITTARANJAN SAHOO received the Diploma and bachelor's degrees in electronic and telecommunication engineering and the master's degree in instrumentation from MIT, Anna University, Chennai, India, in 2001. Since 2001, he has been associated with multiple organizations, mainly dealing with telecommunications products. His research interests include telecommunication software development and maintenance for global operators.

GANESH CHANDRASEKARAN received the bachelor's degree in electronics and communication from the Anna University of Technology, Tamil Nadu, India, in 2011, the master's degree in telecommunications from University College London, U.K., in 2013, and the Ph.D. degree from the 5G Innovation Centre, University of Surrey, U.K., in 2016. Since 2017, he has been associated with the Samsung Electronics Research and Development Center, Suwon, South Korea, and Bengaluru, India. He is currently the Head of Service Management Orchestrator (SMO) and the Network Slice Manager (NSM) Part, Management S/W Group, Samsung Research and Development Institute, Bengaluru. His research interests include cutting-edge topics, including 5G/6G network design, end-to-end network slicing, network function virtualization, cloud and container-native technologies, software-defined networking, information-centric networking, delay tolerant network (DTN), and device-to-device (D2D) communication. Since 2018, he has been serving as an Active Reviewer for IEEE Access.