Towards Greener Networks: A Statistical Approach to RAN Cell Control

ABSTRACT

Mobile communication technologies, in their quest to deliver the highest possible data rates over the air interface, have nearly touched the Shannon limit. Despite the advancements in techniques to enhance service capabilities, the power usage by the radio and compute components at the base stations (known as eNodeB in LTE and gNodeB in NR) frequently constitutes a significant part of the operational costs (OPEX) for operators, a factor that is commonly disregarded. This paper delves into one of the most straightforward strategies for energy conservation in cellular networks: deactivating under-utilized cells. While there has been extensive research in this field, we present a novel, statistics-driven, and machine learning-assisted Energy Saving (ES) solution for Radio Access Network (RAN) cell shutdown. In addition, we introduce a novel validation framework for evaluating decision-making effectiveness and ensuring the Quality of Service (QoS) is maintained for end users. Our simulations, conducted over a O-RAN compliant rApp setup, demonstrate the effectiveness of the proposed technique, resulting in a significant reduction in power consumption.

ACM Reference Format:

. 2025. Towards Greener Networks: A Statistical Approach to RAN Cell Control. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the advancements in techniques to pack maximum data over the communication channel, the Information and Communication Technology (ICT) industry has devised technologies which help us reach the Shannon's limit on amount of data that can be reliably transferred over a channel. One of the most common methods to meet the ever-increasing service demands of the online public is harvesting more spectra for mobile communication networks, like the C-band which provides hundreds of MHz of bandwidth and millimeter wavelength spectrum which offers GHz of spectrum. What is often overlooked are the aspects of practical realization of these techniques which demand huge computation resources both at the radio units as well as for processing within the base-band. The energy consumption of mobile networks is a significant concern, with the ICT industry accounting for 10% of of the worldwide electricity consumption, a figure that is expected to double by 2025 [6]. Achieving energy efficiency in mobile networks, especially with the increasing deployment of 5G networks, is a significant issue facing the ICT industry to meet their goal of net zero emissions by 2050.

Within a mobile network, Radio Access Networks (RAN) have been found to be one the most significant users of a mobile network's total power supplied[9, 10].

This paper examines one of the simplest techniques for energy savings - powering off unused nodes. It aims to establish formalisms for analyzing this technique and its impact on the Quality of Service (QoS) as perceived by the end users. With the flexibility of implementation intrroduced by the O-RAN initiative, we have tested out the algorithim in a rApp compatible with any O-RAN compliant network. Our approach distinguishes itself from existing solutions by leveraging a statistically motivated framework rather than relying heavily on heuristics [4, 8].

Statistical approaches offer a more robust and data-driven foundation for decision-making compared to heuristic methods as is argued well in [2]. This approach allows for better adaptation to dependence on the inherent variability and unpredictability of network traffic patterns. While previous implementations often employ techniques such as toggling between MIMO and SISO [1], using reinforcement learning [11], or realloacting BBUs [5], our method focuses on a more fundamental statistical analysis of network behavior. This statistical foundation allows for a more robust and generalizable solution that can be easily adapted to various network configurations without the need for extensive training or parameter tuning. Unlike approaches that require specific network configurations or rely on proprietary systems, our method integrates seamlessly with the O-RAN architecture through the use of rApps. Figure 1 describes this interfacing with respect to traditional O-RAN architecture.

By combining statistical rigor with O-RAN compatibility, our approach offers a unique balance of effectiveness and practicality that sets it apart from the more heuristic-driven or computationally intensive methods prevalent in the current literature. The main contributions of this paper can be summarised as follows:

- Introduced a statistical metric to evaluate the effectiveness of implemented policies, offering a method to validate decisions prior to implementation.
- Developed a swift and efficient method for cell shutdown, eliminating the need for excessive KPI calls or extensive training time.
- Implemented and detailed a novel solution that is readily deployable across all O-RAN compliant networks.

The remainder of this paper is organized into six sections, as follows. Section 2 provides an overview of the problem at hand and outlines our proposed approach to address it. Section 3 details our energy saving solution, while Section 4 provides an algorithm to arrive at an optimal solution. The results obtained from the rApp evaluation using the software-defined NS-3 simulator are depicted and discussed in Section 5. Finally, Section 6 concludes the paper and suggests future research directions.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

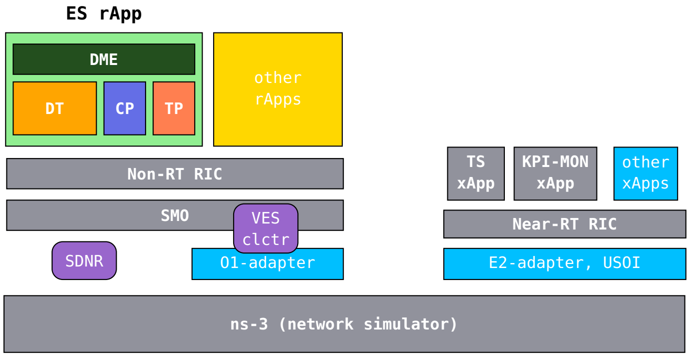

Figure 1: System Architecture Diagram

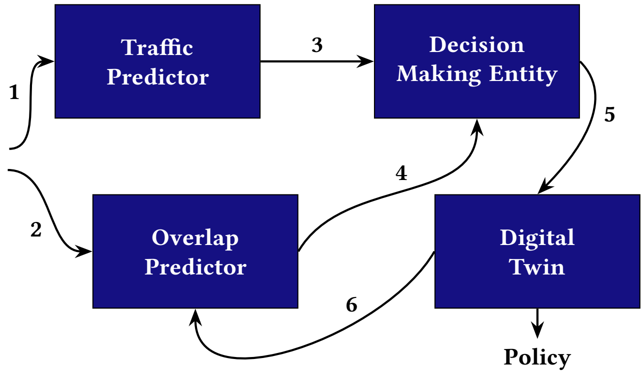

Figure 2: Energy Saving Solution Overview

2 PROBLEM STATEMENT

In case of dense urban deployments, where the greatest number of sites are commissioned, each coverage sector has multiple carriers deployed. This is done for several reasons, mainly to improve the capacity at the site or even to take advantage of different characteristics of different carriers (low-band for coverage, mid-band or high-band for capacity improvements). Additionally, coverage overlap is provided across the sites to improve call drop and handover failures. While complex techniques are used to balance the users and traffic across these carriers locally at the site, the aggregate traffic carried by a regional network is lesser than the total capacity of a properly planned network even during high load situations. During off-traffic hours, these networks tend to be further underutilized. The energy E consumed by each node is modeled as a sum of two parts, the baseline energy consumption and the energy consumed due to traffic:

$$E_{\text{total}} = E_{\text{quiescent}} + \gamma N_{\text{traffic}} \tag{1}$$

Here $E_{\text{quiescent}}$ represents the energy consumed by the system when it's in a quiescent or idle state. This is the baseline energy consumption that occurs regardless of the level of traffic and it depends on the operating point of the radio and server. γ represents the additional energy consumed per unit of traffic, represented as N_{traffic}.

In order to minimize the effects of distortion due to high Peak to Average Power Ratio (PAPR) of orthogonal frequency division access (OFDMA) waveforms, the operating point is so maintained that we have considerable amount of current dissipated even when there are no users latched on the network. Hence, turning off the cells is an effective way of addressing this loss. However, due to the constantly fluctuating nature of network traffic, maintaining such a state is not always effective. Therefore, we also need to activate the cells when the system requires it. The seemingly simple decision of cell control involves multiple *considerations* that must be carefully considered:

C1: What is the optimal timing for deactivation and reactivation of these cells?

C2: Which cells should be deactivated to maximize power savings while minimizing impact on the QoS?

C3: How can we ensure that this policy does not negatively affect overall system performance?

We answer these questions Throughout this paper, we will demonstrate how our designed components address each of these crucial challenges, providing a comprehensive solution to the cell shutdown optimization problem.

3 ENERGY-SAVING SOLUTION

3.1 Overview

The solution to address the three aspects of the problem discussed in the previous section comprises of several components. Figure 2 outlines a simplified version of how the components interact with each other to implement our ES solution. The data at each stage of the operational flow is indicated by numbers (1..6).

The system is fed with the key performance indicators (KPI) data captured from the network (1) along with topology and configuration information (2). The former is consumed by the traffic predictor to determine if in the upcoming hours, the volume of the traffic changes by an amount necessary to relook at the current state of all the nodes. The topology and configuration data (2) is static information usually gathered from the Element Management System (EMS). The decision algorithm relies on the predicted traffic estimates (3) and the overlap predictions (5) to determine the state of the network, i.e., which nodes should be turned on and which should be off.

However, at this stage, we need to make sure that such a decision is not detrimental to the performance of the network. Hence, it is run by the Digital Twin of the network to evaluate the "goodness" and if it passes a threshold, it is presented to the actual control unit, the EMS, for execution. Otherwise, we reevaluate the decision.

Each of the components are complex systems and described in the following sections. The rApp is data driven in the sense that it does not incorporate a rigid logic for cell shutdown, but instead determines the rules which meet the target objective based on the input data and network configuration. The non-RT RIC, in particular, is designed to handle tasks that do not require immediate response. This makes it ideal for applications focused on long-term optimization and strategic planning, such as energy control.

Algorithm 1 `Energy Saving Procedure`,

procedure EnergySavingEntity(τ, curr_tpt, cqi_curr, α_{th})
  tpt_pred ← TrafficPredictor(curr_tpt)
  if tpt_pred > τ **then**
    c_map ← CoveragePredictor()
    node ← max(c_map) ▷ Node with maximum E_{ij} for given sector
    Create policy for Shutdown.
    policy ← node
    cqi_future ← DigitalTwin(policy)
    α ← KL-Divergence(cqi_future, cqi_curr)
    if $\alpha < \alpha_{th}$ **then**
      Transmit Policy.
    else
      Reinvoke EnergySavingEntity(τ, curr_tpt, cqi_curr, α_{th}).
    end if
  else
    for all nodes switched off in the system **do**
      Create Policy to Bringup node n.
      cqi_n ← DigitalTwin(policy)
      α_n ← KL-Divergence(cqi_n, cqi_curr)
    end for
    Select finalPolicy with min $(\mathrm{mod}(\alpha_n - \alpha_{th}))$.
    Transmit finalPolicy.
  end if
end procedure

3.2 Base Station Control Algorithm

4 DECISION ALGORITHM

4.1 Overview

The following subsections provide a detailed workflow of the solution discussed in the previous secction, along with an in-depth explanation of the components involved in this decision-making process. This section provides answers to the challenges outlined in the Problem Statement. Each component and its corresponding challenge are discussed in detail.

Within the O-RAN framework, the application manifests as a rApp hosted in Non Realtime RIC and the decision is fed to the xApps and SDNR. The data collection and cleaning is done at the edge cloud to take advantage of the distributed processing and avoid pushing large amounts of data to regional data centers. Firstly, the E2 Nodes are configured by the Service Management and Orchestration (SMO) to report the data necessary via the O1 Interface. The functioning of the Non-RT RIC and SMO are tightly coupled, which enables the Non-RT RIC to retrieve the collected data through internal SMO communication.

In our setup, the rApp receives input data from the Radio Database, Traffic Predictor, and Coverage Predictor, each answering a question posed in 2. The rApp sends a shutdown/bringup policy as a declarative statement, across the A1 interface, to the Near-RT RIC. A Traffic Steering xApp assists in the "safe harboring" of users connected to an eNB before shutdown and bringup through the handover process. The decision is made periodically, with a 1-hour prediction window and 15-minute slots, i.e., four predictions are made every window. The rApp can import data from RF link simulators and drive tests through an external interface. A Dashboard for visualization of the system setup is also used with individual components represented in the Results.

4.2 Decision Making Entity (DME)

4.2.1 Cell Deactivation. The Decision Making Entity, the cornerstone of our Energy Saving Solution, serves two primary functions: determining whether a cell should be considered for shutdown or bringup, and assessing the consequences of energy-saving decisions prior to modifying the network configuration. The decision-making process leverages real-time data and incorporates historical predictions from both the Traffic Predictor and Coverage Predictor. The entity utilizes short-term throughput forecasts from the Traffic Predictor to determine whether a cell should be considered for shutdown/bringup. The entity considers shutting down a cell if the throughput exceeds a certain threshold, and conversely, contemplates activating a cell if the throughput falls below this threshold.

After finalizing the decision to either activate or shut down a cell, the Coverage Predictor aids the entity in identifying the network sectors that can be deactivated with minimal service disruption. Once the control decision and its target cell are both confirmed, the entity uses the Digital Twin's simulations to assess the potential impact of the energy-saving policy before configuring the network via the SMO. The overall functioning of this entity is defined in the procedure detailed in Algorithm 1.

4.3 Γ1: Traffic Prediction (TP)

The Traffic Predictor estimates the net traffic volume for each sector as a function of time, helping us decide when would be an potimum time for shutdown. There is no existing technology that can model the traffic in a network with 100% accuracy. This observation is well-established and can be ascribed to the inherent unpredictability of network traffic. Our approach emphasizes the use of a predictive model to accurately anticipate network traffic *fluctuations*. We establish a throughput threshold, beyond which network configurations require modification. The model is trained with the anticipation that it can guide us towards the appropriate direction of change, accounting for a certain degree of expected error. To prevent altering our system's configarition based on an erroneous prediction, we use the Digital Twin to simulate the effects of the change before implementing it in the real network. We intended to find a regression model that, above all, identified the trend and seasonality of traffic fluctuations.

We use an offline model for learning because using a pre-trained model with sufficient data does seem to suffice to predict traffic directions in our given setup. In further versions of the solution, we plan to use an online learning model to update the model with real-time data. Keeping in that in mind, we performed a few experiments with different regression models and found that the LSTM model was the best fit for our requirements. Although we conducted thorough experiments for model selection and verification, we could not include the details in this paper due to space constraints.

Algorithm 2 `Energy Saving Procedure`,

procedure ENERGYSAVINGENTITY(τ, curr_tpt, cqi_curr, α_{th})
    tpt_pred ⟵ TrafficPredictor(curr_tpt)
    if tpt_pred > τ **then**
        c_map ⟵ CoveragePredictor()
        node ⟵ max(c_map)   ▷ Node with maximum E_{ij} for given sector
        Create policy for Shutdown.
        policy ⟵ node
        cqi_future ⟵ DigitalTwin(policy)
        α ⟵ KL-Divergence(cqi_future, cqi_curr)
        if $\alpha < \alpha_{th}$ **then**
            Transmit Policy.
        else
            Reinvoke EnergySavingEntity(τ, curr_tpt, cqi_curr, α_{th}).
        end if
    else
        for all nodes switched off in the system **do**
            Create Policy to Bringup node *n*.
            cqi_n ⟵ DigitalTwin(policy)
            α_n ⟵ KL-Divergence(cqi_n, cqi_curr)
        end for
        Select finalPolicy with min (mod($\alpha_n - \alpha_{th}$)).
        Transmit finalPolicy.
    end if
end procedure

In our solution, this prediction is based on historical data and previous measurements. The Traffic Predictor uses a pre-trained LSTM with 64 cells to forecast these values for the near-future. The LSTM was trained on on initial system data (initial 300 entries from NS3 simulator), using a batch size of 32 and 100 training epochs. The inputs to the LSTM model are throughput, cell to which throughput belongs and the timestamp of the reading. Every 1 hr, the model makes four fresh predictions (+15, +30, +45, +60). Based on this, a decision on when cell control is to be implemented is made.

4.4 Γ2: Digital Twin (DT)

The Digital Twin is a powerful tool for network management and optimization, as it allows operators to test and predict the effects of changes of a policy in a risk-free virtual environment before implementing it in a real network. In the context of our solution, the Digital Twin is used to simulate a cellular network and is used to understand how the cell shutdown/bringup will affect the system overall. Our solution utilizes the same validation process to confirm the effectiveness of our policy decisions.

It is implemented using CloudRF [3]. The coverage area is represented as a 30 x 30 pixel grid, with power readings simulated for each individual pixel. CloudRF is used to map out the area of service and simulate the network characteristics across it. CloudRF generates predictions of the expected CQI distribution of the system using a Radio Link budget simulator. This system is initialized with network inventory and predicted RF power (downlink) for each pixel from all sectors in service.

4.5 Γ3: Coverage Prediction (CP)

The Coverage Predictor estimates the *coverage overlap*, the areas where signals from neighboring sectors intersect. It identifies sectors that, if shut down, would not impact the overall network coverage. Sectors exhibiting the highest degree of overlap are prioritized for shutdown, given that their discontinuation is less likely to impact coverage due to the compensatory capabilities of the remaining interconnected sectors.

The system takes as input the simulated received power level (sourced from CloudRF) for each pixel from the participating sectors. The system outputs a matrix, known as the Coverage Map, which represents the degree of overlap between these sectors. The algorithm for the Coverage Predictor is detailed in Algorithm 2. In the algorithm, E_{ij} represents the number of sectors which have overlaps with other sectors.

Algorithm 3 `Coverage Predictor Algorithm`,

function COVERAGEPREDICTION(Network)
Input: Network Configuration
Output: ***OverlapGraph*** representing the degree of overlap between sectors
    for each pixel in CloudRF p **do**
        Identify sector $S_{serving}$ with highest power at p.
        Mark others as interfering sectors.
    end for
    for each sector S_i **do**
        for each interfering sector S_j **do**
# Adjacency condition: There exists a pixel in S_i where S_j is the strongest interferer.
            if S_i and S_j are adjacent **then**

$$P_{ij} \leftarrow \{p \in S_i \mid S_j \text{ is the strongest interferer for } p\}$$

                $E_{ij} = \sum_{p \in P_{ij}} p$
            end if
        end for
    end for
    Generate ***OverlapGraph*** :
    - Each sector S_i is a vertex.
    - Each E_{ij} is a weighted edge between S_i and S_j.
end function

4.6 Measuring QoS Gurantees

In the realm of networking, QoS primarily entails ensuring a specific level of performance for a data flow. This is achieved by prioritizing certain network characteristics over others. In order to uphold QoS commitments in our system, it is essential to ensure that the implementation of energy-saving policies does not lead to a non-trivial decline in network performance. This can be achieved by ensuring the system operates optimally at the outset and maintaining its initial state throughout. This principle guides our approach to monitoring system functioning. Among these variables, the *CQI* values of the UEs connected to the active cells and the system's *total throughput* are of paramount importance.

Why don't we consider the throughput of each individual UE? Firstly, it's logistically impractical. UEs connect and disconnect from the network at a rapid pace, making it difficult and computationally intensive to track individual allotments. Secondly, as long as the total system performance remains consistent with its state prior to control application, our initial QoS is assured. Rather than focusing on individual metrics, we concentrate on measuring the system's CQI distribution. A low CQI value signifies poor channel quality, while a high CQI value signifies excellent channel quality. Our goal is to utilize high-quality channels. By consistently maintaining the use of such channels, we can ensure the preservation of the QoS to the users.

The overall system's CQI is measured by assigning each UE to a CQI-value 'bin', which is determined based on the channel quality measured by the core network. The distribution of UEs across CQI bins closely mirrors a discrete probability distribution of CQI values across the network. We employ the Kullback-Leibler (KL) divergence, a statistical measure from information theory, to ensure that the input or output distributions do not deviate significantly from a baseline distribution. The KL divergence of two probability distributions P and Q is defined as:

$$D_{KL}(P, Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)} \tag{2}$$

In this context, $P(i)$ and $Q(i)$ represent the probabilities of the ith CQI bin in the respective distributions P and Q.

We derive the initial CQI distribution from the NS-3 simulator. Subsequently, the rApp policy is applied to generate a new network configuration, which is simulated in the Digital Twin to obtain a subsequent CQI distribution. If the control policy doesn't cause a substantial divergence from the baseline in the simulation (quantified using the previously defined KL Divergence), it is subsequently forwarded to the Near-RT RIC. The user sets the difference threshold, α_{th}, which varies based on the specific environment the network is present in.

5 PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed Energy Saving solution in our software-defined O-RAN network. We first discuss the simulation setup, followed by the results of our experiment and a discussion on maintaining QoS guarantees. In the ensuing graphs, to illustrate the long-term impact of the energy-saving algorithm, we've adopted a time conversion convention where 10 seconds equate to 15 minutes.

5.1 Simulation Setup

Our solution is deployed as an independent rApp, interfacing with the Non-RT RIC and the A1 interface of the O-RAN stack. This rApp dispatches decision-making policies to the Near-RT RIC, which houses the TS xApp responsible for reallocating UEs during cell shutdown or bringup processes. The network, simulated as a Digital Twin using an NS-3 Simulator [7], comprises eight cells with 20 User Equipments (UEs) per cell, operating in a single-threaded mode. The rApp operates in a feedback loop with the Digital Twin, obtaining power readings, throughput, and other network characteristics across the coverage area. The network is configured to operate in a 5G New Radio (NR) based CBRS network deployment.

Two Digital Twins? The Digital Twin in use here is a different one as the one mentioned earlier in the paper. For simplicity, we refer to the Digital Twin implemented with NS-3 as the 'NS-3 Simulator', and the one implemented with CloudRF as the 'Digital Twin'. The Digital Twin integrated into the ES rApp, is a streamlined simulator that reports selected characteristics of the deployed system, implemented using CloudRF. The NS-3 simulator serves as the backbone for our network simulation, encompassing the core network, the gNBs, and the UEs.

5.2 Power Consumption Reduction

We have conducted a series of experiments to evaluate the performance of our proposed solution especially focusing on the reduction in the power consumption over the given time-frame. We look at the performance figures on how the forecasts of the overall throughput forecast helps achieve cell shutdown and bringup. Figure 3 depicts two scenarios of cell shutdown and activation: the upper scenario illustrates this process in a single-cell setup, while the lower scenario demonstrates it in an eight-cell configuration. The LSTM-based Traffic Predictor effectively identifies the optimal times for shutdown and activation, as clearly demonstrated in the single-cell setup. Furthermore, our system has the capability to operate across multiple cells simultaneously.

The most straightforward method to observe a decrease in energy consumption involves continuously tracking the total power usage of the network, specifically within the NS-3 simulator. In our implementation, we've incorporated monitoring hooks to track this value. The observed data is visualized in Figure 5. In our simulations, we observed a 32% reduction in power across short sample durations. We anticipate that the power savings could be significantly higher in longer and more complex setups.

5.3 Maintaining QoS Guarantees

In this section, we demonstrate how our setup maintains the QoS guarantees during system operation. By utilizing the CQI data collected from all connected UEs, we can illustrate the system's CQI distribution at any given moment. Figure 4 illustrates the CQI distribution of the system before (depicted in the upper graph) and after (shown in the lower graph) the implementation of a cell shutdown policy. The system's probability distribution experiences only minor changes, with a single CQI bin undergoing significant alterations. This further substantiates that our ES solution preserves the channel quality for all UEs connected to the active eNBs, thereby ensuring the QoS.

6 CONCLUSIONS

This work aimed to develop and evaluate an Energy Saving Solution as a rApp for the O-RAN architecture. The approach consisted of using using various components to implement our solution. The novel aspect of our approach lies in the use of a Digital Twin to validate the proposed solution using KL Divergence as a statistical measure to accurately quantify the changes in our system. We leveraged machine learning to identify patterns in network traffic, rather than attempting to construct a fully accurate model of the

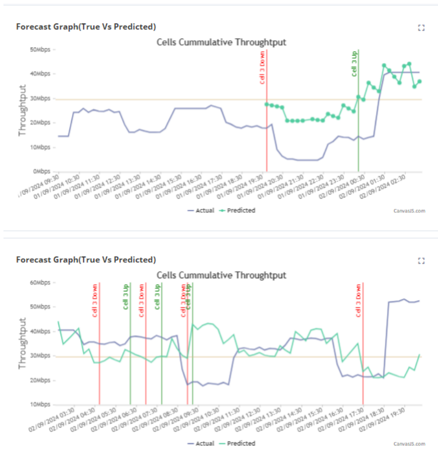

Figure 3: Functioning of our ES rApp over the NS-3 Simulator

Figure 4: Comparison of CQI distributions before and after the implementation of cell shutdown

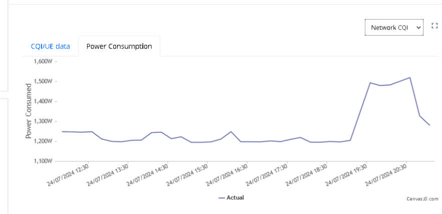

Figure 5: Power Consumption of the Setup over time

network traffic. The decision-making entity and coverage predictor operated based on cellular-informed logical rules.

The energy saving results via ML-enabled rApp control in the the simulated NS-3 environment are encouraging and provide a basis for further enhancement in the ML model as well as the decision-making entity to incorporate other decision variables as the future scope of the work. The results indicate that the proposed solution can be a viable option for operators to reduce their OPEX while maintaining the QoS for their subscribers. Further work can include impelmenting other prediction models to analyze different model performances in the end-to-end experimental deployment. Furthermore, the enhanced rApp version provides an overall energy-saving solution to be used for efficient RAN control/management, not only in experimental simulations but also in any real-world environment.

A COVERAGE ALGORITHM

Algorithm 4 `Coverage Predictor Algorithm`,

function CoveragePrediction(Network)
  Input: Network Configuration
  Output: ***OverlapGraph*** representing the degree of overlap between sectors
  for each pixel in CloudRF p **do**
    Identify sector $S_{serving}$ with highest power at p.
    Mark others as interfering sectors.
  end for
  for each sector S_i **do**
    for each interfering sector S_j **do**
  # Adjacency condition: There exists a pixel in S_i where S_j is the strongest interferer.
      if S_i and S_j are adjacent **then**

$$P_{ij} \leftarrow \{p \in S_i \mid S_j \text{ is the strongest interferer for } p\}$$

        $E_{ij} = \sum_{p \in P_{ij}} p$
      end if
    end for
  end for
  Generate ***OverlapGraph*** :
  - Each sector S_i is a vertex.
  - Each E_{ij} is a weighted edge between S_i and S_j.
end function

B DASHBOARD GUI

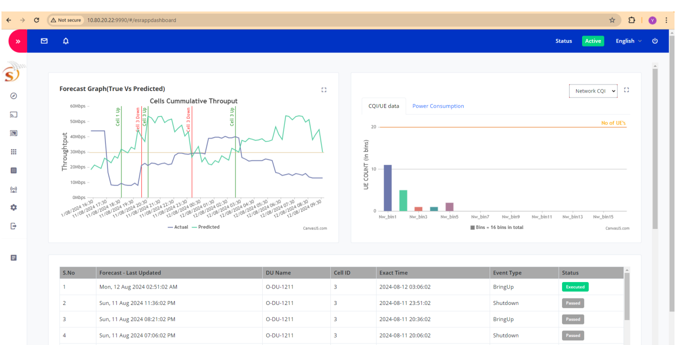

Figure 6: ES rApp GUI

C DESIGN RATIONALE

In this section, we elucidate our reasoning behind the choice of regression model used to represent the overall network throughput.

C.1 Datasets In Use

We intended to find a regression model that, above all, identified the trend and seasonlity of traffic fluctuations. The models underwent evaluation using a mix of four real-world and five synthetic time-series datasets, each exhibiting diverse trends and seasonal patterns:

- Dataset 1: COMED Dataset - This real-world dataset, released by the Commonwealth Edison Company, illustrates the temporal variations in power consumption across a specific group of households.
- Dataset 2: Microsoft Dataset - This dataset, obtained using a data scraper, encapsulates the temporal variations in Microsoft's stock price.
- Dataset 3: Temperature Dataset - This dataset, sourced from Kaggle, depicts the temporal progression of the Earth's surface temperature.
- Dataset 4: No Trend Dataset - This synthetic dataset, created using a blend of sinusoidal and random noise functions, exhibits no discernible trend or seasonality.
- Dataset 5: Upwards Trend Dataset - This synthetic dataset is similar to Dataset 4, but it exhibits a noticeable upward trend (without any seasonality).
- Dataset 6: Downwards Trend Dataset - This synthetic dataset is similar to Dataset 4, but it exhibits a noticeable downward trend (without any seasonality).
- Dataset 7: Upwards Trend Dataset with Seasonality - Dataset 5 with added seasonality.
- Dataset 8: Downwards Trend Dataset with Seasonality - Dataset 6 with added seasonality.
- Dataset 9: Simulator Dataset - A synthetic dataset generated using our ns-3 simulator, taken to ensure that these models perform with traffic data and not just randomized time-serieses.

C.2 Model Selection

The choice of regression model used for Traffic Prediction is crucial to the success of the solution. In this section, we compare the performance of three different regression models: Prophet, ARIMA, and LSTMs. We train our models on all our real-world datasets (COMED, Microsoft and Temperature) and evaluate their performance on a validation set of the same dataset. Our findings can be seen in Table 1. We observe that the LSTM model outperforms Prophet, capturing the trend and seasonlity of the data the best. The ARIMA model was found to be outright the worst performer, both taking the longest to train as well making forecasts completely ignoring the trend and seasonlity of the inputed data. Considering how promising the LSTM's performance seemed, we decided to further evaluate the same.

C.3 Model Verification

After arriving at using LSTMs as the model of choice for traffic forecasting, we had to ensure that the model would be able to handle the simulated load. We did so using synthetic data of various types, as outlined in our Dataset section. To verify the robustness of the model's forecasts, we trained the LSTM models using a diverse range of datasets, each exhibiting unique general trends. For each dataset, we trained a corresponding LSTM model. We used Mean Squared Error (MSE) as an evaluation index to evaluate the forecast accuracy of the models. Subsequently, we cross-validated each trained model with the remaining datasets. The MSE values of all the trained models and the datasets in use is described in in Table 2. We observe that the LSTM trained on data with more seasonlity

Table 1: Performance of Different Models on Various Synthetic Datasets

Dataset Type	Prophet Performance	LSTM Performance
No Trend, No Seasonality	Does not trend correctly, trends in the opposite direction	Follows trend but does not account for the noise well
Upwards Trend, No Seasonality	Steadily trends upwards but not according to the data (underfits)	Follows trend but predicts widely off values when accounting for noise
Downwards Trend, No Seasonality	Trends appropriately but underfits, does not recognize dataset intricacies	Recognizes trend and seasonality but produces very inaccurate values due to noise
Upwards Trend, With Seasonality	Recognizes trend but not seasonality	Recognizes trend and seasonality well, performs satisfactorily with test data
Downwards Trend, With Seasonality	Recognizes trend but not seasonality	Recognizes trend and seasonality but produces very inaccurate values due to noise

Table 2: Performance of Different LSTM Models on Various Datasets

Data/Model	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
Model 1	0.044	0.0054	0.2507	2.0302	0.6256	1.8354	1.186	1.1158
Model 2	0.3358	0.2262	0.2609	0.915	0.2613	0.855	0.486	0.4774
Model 3	0.5695	0.7270	0.4274	1.0624	0.3611	1.091	0.6027	0.593
Model 4	1.0267	0.5424	0.7875	0.5457	0.7711	1.3335	0.6144	0.9841
Model 5	0.5713	0.6027	0.4731	1.069	0.2857	1.011	0.5556	0.5418
Model 6	1.3595	1.1141	1.3407	1.2204	0.9841	1.2831	1.0261	1.0280
Model 7	0.5459	0.3503	0.2500	0.9837	0.2005	0.9108	0.4610	0.4590
Model 8	0.5806	0.2504	0.1865	0.9605	0.1640	0.8965	0.417	0.4134

(model 7,8 and 9) perform the best all around, with the lowest MSE values. This is expected, as the LSTM model is designed to capture the long-term dependencies in the data, which are more prevalent in datasets with seasonality.

Therefore, when training our model on real-world data, we should ensure that the data has a significant amount of seasonality to ensure the best performance. The amount of data used to train the LSTM model is crucial to the success of the solution. If we train the model on excessive data, the model may overfit to the training data and fail to generalize to unseen data. This would be especially catastrophic in our specific use case, as we This will depend on the deployemnt environment's complexity, and in our specific setup we found 300 samples to suffice.

REFERENCES

[1] Mariam Aboelwafa, Mohamed Zaki, Ayman Gaber, Karim Seddik, Yasser Gadallah, and Ayman Elezabi. 2020. Machine Learning-Based MIMO Enabling Techniques for Energy Optimization in Cellular Networks. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)* (Las Vegas, NV, USA). IEEE Press, 1–6. https://doi.org/10.1109/CCNC46108.2020.9045149

[2] Leo Breiman. 2001. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statist. Sci.* 16, 3 (2001), 199 – 231. https://doi.org/10.1214/ss/1009213726

[3] CloudRF. 2024. *CloudRF*. https://www.cloudrf.com/

[4] Hasna Fourati, Rihab Maaloul, Lamia Fourati, and Mohamed Jmaiel. 2023. An Efficient Energy-Saving Scheme Using Genetic Algorithm for 5G Heterogeneous Networks. *IEEE Systems Journal* 17, 1 (2023), 589–600. https://doi.org/10.1109/JSYST.2022.3166228

[5] Yongqin Fu and Xianbin Wang. 2022. Traffic Prediction-Enabled Energy-Efficient Dynamic Computing Resource Allocation in CRAN Based on Deep Learning. *IEEE Open Journal of the Communications Society* 3 (2022), 159–175. https://doi.org/10.1109/OJCOMS.2022.3146886

[6] Erol Gelenbe. 2023. Electricity Consumption by ICT: Facts, trends, and measurements. *Ubiquity* 2023, August, Article 1 (Aug. 2023), 15 pages. https://doi.org/10.1145/3613207

[7] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. 2008. Network simulations with the ns-3 simulator. *SIGCOMM demonstration* 14, 14 (2008), 527.

[8] Yin-Hsin Liu and Bo-Chen Kung. 2023. Energy Saving in 5G Cellular Networks using Machine Learning Based Cell Sleep Strategy. In *2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 154–159.

[9] Next Generation Mobile Networks (NGMN). 2024. *Sustainability Challenges and Initiatives in Mobile Networks*. Technical Report. NGMN Alliance. https://www.ngmn.org/publications/sustainability-challenges-and-initiatives-in-mobile-networks.html

[10] Prabhu Kaliyammal Thiruvasagam, Chandrasekar T, Vinay Venkataram, Vivek Raja Ilangovan, Maneesha Perapalla, Rajisha Payyanur, Senthilnathan M D, Vishal Kumar, and Kokila J. 2023. Open RAN: Evolution of Architecture, Deployment Aspects, and Future Directions. arXiv:2301.06713

[11] Qiong Wu, Xu Chen, Zhi Zhou, Liang Chen, and Junshan Zhang. 2021. Deep Reinforcement Learning With Spatio-Temporal Traffic Forecasting for Data-Driven Base Station Sleep Control. *IEEE/ACM Transactions on Networking* 29, 2 (2021), 935–948. https://doi.org/10.1109/TNET.2021.3053771