Towards Greener Networks: A Statistical Approach to RAN Cell Control

Anonymous Authors

ABSTRACT

Mobile communication technologies, in their quest to deliver the highest possible data rates over the air interface, have nearly touched the Shannon limit [CITE]. This has been made possible through the implementation of Multi-user MIMO, beamforming, and the utilization of large bandwidths in the C-band and millimeter-wave spectrum [CITE]. 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. Although this technique has been extensively researched, we introduce a novel validation framework for evaluating its effectiveness and ensuring the Quality of Service (QoS) is maintained for end users. Our simulations, conducted over a rApp setup, demonstrate the effectiveness of the proposed technique, resulting in a significant reduction of [VALS] in power consumption.

ACM Reference Format:

Anonymous Authors. 2024. Towards Greener Networks: A Statistical Approach to RAN Cell Control. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

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. These include efficient channel coding techniques like Low Density Parity Checks (LDPC) to spatial re-use of channel as in Multi-user MIMO coupled with beamforming to direct data to specific users or set of users. In order to meet the ever-increasing service demands, more spectra has been harvested 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 in base-band. Achieving energy efficiency in mobile networks would be the largest problem 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. Recent initiatives such as Open RAN (O-RAN) [] have introduced the concept of RAN Intelligent Controllers (RICs) as flexible platforms for robust RAN control, platforms which we put into use. The disaggregation introduced in O-RAN paves way for the introduction of several network optimization techniques without disturbing core functionality. O-RAN control is enabled using applications called xApps (for Near-Real-Time RIC) and rApps (for Non-Real-Time RIC), with the choice of the implementation made depending on the time-frame of the control.

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 introduced by the O-RAN initiative, we have tested out the algorithm in a rApp compatible with any O-RAN compliant network. The main contributions of this paper are 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 solution that is readily deployable across all O-RAN compliant networks.

How is this different from all the other implementations? Is this just a simple implementation of a pre-existing idea in an O-RAN specification?

Usual heuristic heavy approaches to cell shutdown/bringup: 1. toggle between MIMO and SISO –> Machine Learning-Based MIMO Enabling Techniques for Energy Optimization in Cellular Networks (IEEE)

2. does what we do, except using GS-STN and RL –> Deep Reinforcement Learning With Spatio-Temporal Traffic Forecasting for Data-Driven Base Station Sleep Control (ACM)

3. more focused on TS xApp bit –> Energy Optimization in Ultra-Dense Radio Access Networks via Traffic-Aware Cell Switching (IEEE)

4. predict BBUs based on RRH traffic in a CRAN env. –> Traffic Prediction-Enabled Energy-Efficient Dynamic Computing Resource Allocation in CRAN Based on Deep Learning (IEEE)

5. similar, but QoS is guaranteed using spectral efficiency –> A New Heuristic Algorithm for Energy and Spectrum Efficient User Association in 5G Heterogeneous Networks (IEEE)

6. properly heuristic –> Energy Saving in 5G Cellular Networks using Machine Learning Based Cell Sleep Strategy (IEEE), An Efficient

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00

https://doi.org/XXXXXXX.XXXXXXX

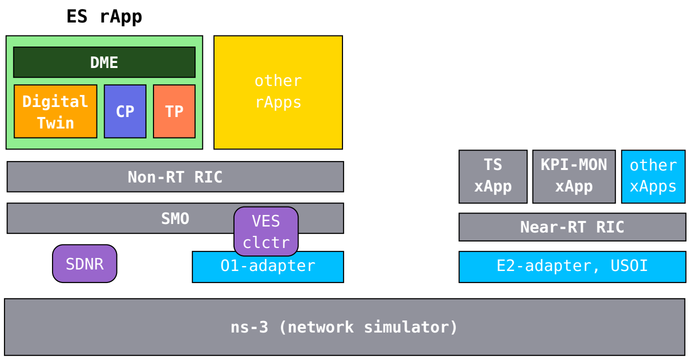

Figure 1: System Architecture Diagram

Energy-Saving Scheme Using Genetic Algorithm for 5G Heterogeneous Networks (IEEE)

The remainder of this paper is organized into six sections, as follows. Section 2 provides a discussion of current approaches to energy saving with the RAN stack, and how our approach has it's own merits. Section 3 contains an overall overview of our energy saving solution, while scxtion 4 provides an algorithm to arrive at 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.

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 [CITE] 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

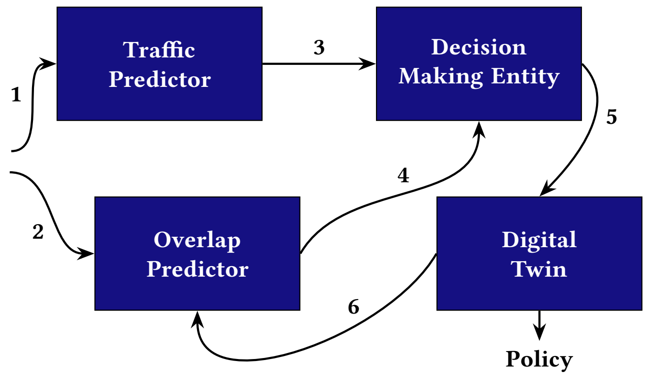

Figure 2: System architecture diagram

that we have considerable amount of current dissipated even when there are no users latched on the network. Hence, turning off the radios is an effective way of addressing this loss. However, this seemingly simple decision has multiple aspects to be looked at, the most important ones being:

- Which carriers should be switched off to achieve optimum power saving without impacting the QoS?
- What is the right time to switch off or on these carriers?

Although these are well studied problems, most of the implementations and studies are based on heuristics. CITE While heuristics can speed up the problem-solving process, they may not always provide the most accurate or optimal solution. This paper formulates the problem in a way that it can be solved with closed loop methods with finite time algorithms. We also define a metric based on statistical techniques to evaluate the "goodness" of any algorithm which attempts to implement energy saving through this or other techniques.

3 ENERGY SAVING SOLUTION

The solution to address the two aspects of the problem discussed in the previous section comprises of several components as shown in CITE. 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, typically the EMS, for execution. Otherwise, we reevaluate the decision. Each of the components are complex systems and described in the following subsections.

Within the O-RAN framework, the application manifests as a rApp hosted in Non Realtime RIC and the decision is fed to xApp 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.

The rApp is data driven in the sense that it does not incorporate a rules-based logic but 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.

In our setup, the rApp receives input data from the Radio Database, Traffic Predictor, and Coverage Predictor, and sends a shutdown/bringup policy (a declarative statement across the A1 interface) to the Near-RT RIC. The Shutdown and Bringup of nodes is handled by a Traffic Steering xApp. 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 is designed to be shared across multiple rApps and can import data from RF link simulators and drive tests through an external interface. A Dashboard for visualization of the Radio Mapping Database is also used as shown in the [CITE].

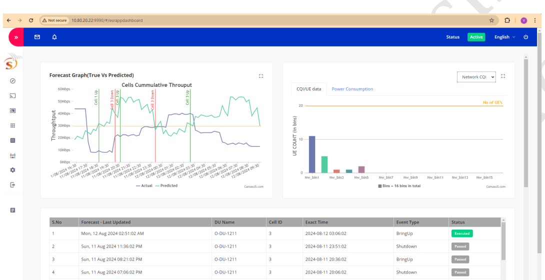

Figure 3: ES rApp GUI

4 DECISION ALGORITHM

The following subsections provide a detailed workflow of the solution discussed in [[CITE]], along with an in-depth explanation of the components involved in this decision-making process.

4.1 Decision Making Entity

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

Algorithm 1 `Energy Saving Entity Algorithm`,

procedure ENERGYSAVINGENTITY(τ, curr_tpt, cqi_curr, α_{th})
  tpt_pred ← TrafficPredictor(curr_tpt)
  if tpt_pred > τ **then**
    # Cell shutdown procedure
    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
    # Cell bring-up procedure
    for all nodes switched off in the system **do**
      Create policy to switch on 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

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 based on the network's predicted throughput values. 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 [[CITE]]

4.2 Traffic Prediction

The Traffic Predictor estimates the net traffic volume for each sector as a function of time. This prediction is based on historical data and previous measurements. The Traffic Predictor uses pre-trained LSTM to forecast these values for the near-future. The model is itself trained on initial system data (initial 300 entries from NS3 simulator) 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)

There is no existing technology that can model the traffic in a network with 100% accuracy. This is a known fact, which we can attribute to the stochastic nature of the traffic in the network. 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 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. Another reason, we do not use an online model is because traffic data varies irratically and not all the data we recieve is a scenario we want to model for. 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. Our experiments and subsequents observations are detailed in [CITE].

4.3 Digital Twin

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. CloudRF [CITE] is used to map out the area of service and simulate the power readings across it. The coverage area is represented as a 30 x 30 pixel grid, with power readings simulated for each individual pixel. 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 sectors which exceed the predefined threshold (P_th).

4.4 Coverage Prediction

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 all sectors contributing more than P_th. The system outputs a matrix, known as a Coverage Map, which represents the degree of overlap between these

Algorithm 2 `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 ← {p ∈ S_i | S_j is the strongest interferer for p}
                E_ij = Σ_{p∈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
```

sectors.P_th of the system is a variable set depending on the environemnt in which the system is running. It depends on the sensitivity desired by the user and is set by the user. The algorithm for the Coverage Predictor is detailed in Algorithm CITE E_{ij} is the number of sectors which have overlaps with other sectors.

[PRAMIT] - Could we represent this using a diagram of some sorts? Would be helpful.

4.5 Measuring QoS Gurantees

In the context of networking, Quality of Service (QoS) mainly involves guaranteeing a certain level of performance to a data flow. [https://dl.acm.org/doi/epdf/10.1145/173942.173943]
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 substantial 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 *Channel Quality Indicator (CQI)* values of 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 Quality of Service (QoS) is assured. While it's crucial to maintain a constant overall throughput, it's challenging to do so consistently, especially as network throughput tends to decrease when the network load is excessively high. Rather than focusing on individual metrics, we concentrate on measuring the system's Channel Quality Indicator (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.

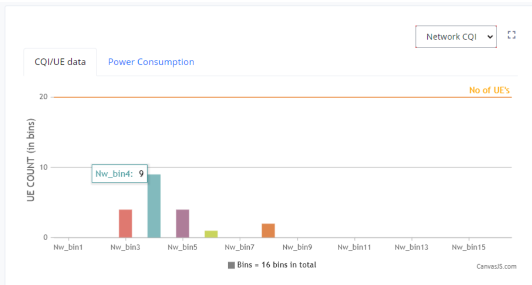

Figure 4: CQI Distribution

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 [CITE]. The KL divergence of two probability distributions P and Q is defined as [CITE]:

$$D_{KL}(P||Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)} \tag{1}$$

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. The KL Divergence Function cite is employed to compare these distributions. If the control policy does not induce a significant deviation from the baseline in the simulation, it is then relayed to the Near-RT RIC. The user sets the difference threshold, α_{th}, which varies based on the specific situation.

5 DESIGN RATIONALE

This section deals with explaining our reasoning and the experiments that lead to the methodology of our solution implementation. Initially, we discuss how the decision variables taken for our solution were chosen. In the later set of experiments, we elucidate our reasoning behind the choice of regression model used to represent the overall network throughput.

5.1 Decision Variables and Threshold Selection

The decision-making process in any solution is governed by a set of decision variables that determine the course of action to be taken. We proceed to evaluate the performance of our Energy Saving solution in terms of the three metrics mentioned below:

- **Network CQI Distribution:** As described in [CITE], we categorized the CQI values of the UEs based on the channel quality.
- **Network Throughput:** This metric represents the total throughput used by all the UEs connected to the system.
- **System Power Consumption:** This is the total power consumed by the system, measured in Watts (W).

The most important one to consider here is the throughput of the network, which is the primary metric used to decide whether a system needs a change in it's configiration or not. The rationale behind this is simple: our focus is on the aggregate network performance, and throughput serves as a reliable metric for this purpose. If the network state, specifically the channel quality and overall throughput, remains largely unchanged after the application of network-modifying policies, we deem the result acceptable.

The determination of thresholds for various decision variables is often crucial to the success of the algorithm. In this context, we aim to derive specific expressions to guide the selection of these variables. In our proposed solutions, we have three primary decision variables: τ, α_{th}, and P_{th}. - τ is the threshhold on the forecasted throughput, which is used to decide whether to shut down a cell or not. - α_{th} is the allowed divergence of the forecasted/likely CQI distribution from the current CQI distribution. It is used to decide whether a policy should be implemented or not. - P_{th} is the threshold on the power consumption of the cell. Only cells function above a certain energy-consumption threshold are to be considered for shutdown/bringup.

[PRAMIT]
Could you please provide a short write-up on how τ, α_{th} and P_{th} are selected? What are the factors we consider?

5.2 Datasets In Use

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, sourced from 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.

5.3 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 [CITE], 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 [CITE], with the graphs of our forecasts attached in the [CITE]. We observe that the LSTM model outperforms the other models, capturing the trend and seasonlity of the data the best, which are more prevalent in time-series data.

5.4 Model Verification

After arriving at using LSTMs as the model of choice for traffic forecasting, we proceed to verify that the model will be able to handle the simulated load using synthetic data. 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 model using the remaining datasets. The MSE values of all the trained models and the datasets in use is described in in Figures [CITE]. We observe that the LSTM trained on data with more seasonlity (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 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 will depend on the deployemnt environment's complexity, and in our specific setup we found 300 samples to suffice.

6 PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed Energy Saving solution in our software-defined O-RAN network. Our setup is lightweight, focusing primarily on evaluating the effectiveness of our proposed algorithm rather than the performance of the underlying simulations. We first discuss the simulation setup, followed by the threshold selection process of various decision variables. Our experiments to prove the effectiveness of the proposed algorithm are then presented, followed by 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.

6.1 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, 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 characteristics across the coverage area. The network is configured to operate in a 5G New Radio (NR) based CBRS network deployment.

The Digital Twin in use here is a different one as the one mentioned earlier in the paper in the Algo section [CITE]. 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. The Digital Twin, as previously described, supplements this setup by furnishing the rApp with requisite data. A more comprehensive explanation of the Digital Twin setup can be provided in the Appendix. [CITE]

6.2 Power Consumption Reduction

As described in [CITE], the proposed algorithm ensures that the QoS guarantees are maintained during the cell shutdown and bringup processes. We try to ensure the policy proposed does not lead to a significant drop in the overall CQI of the channels in use for transmission. We have conducted a series of experiments to evaluate the performance of our proposed solution especially considering the reduction in the power consumption over the given . First, we look at the performance figures on how the forecasts of the overall throughput help achieve cell shutdown and bringup.

[YOGESH]
What is the experiment we are performing? –> A simulation over a certain time-frame, which shows:
- Throughput increase or decrease leads to cell shutdown or bringup.
- How the power consumption of the system decreases over time.
- How the CQI/UE distribution is maintained for atleast a given timeframe
. I would like if the first two can be be compared vertically, with time stamps for cell shutdown/bringup maintained. The second graph can simply be a comparision of the CQI distribution at the start and the end of the experiment.

6.3 Maintaining QoS Guarantees

[YOGESH]
We can view the results in Fig 3. for Cell Shutdown and Bringup both. Fig 4. shows how our given solution leads to an eventual decrease in total power consumption of the system. Fig 5. (right now Fig. 2) shows how the CQI distribution is maintained over the given time-frame.

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

Dataset/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

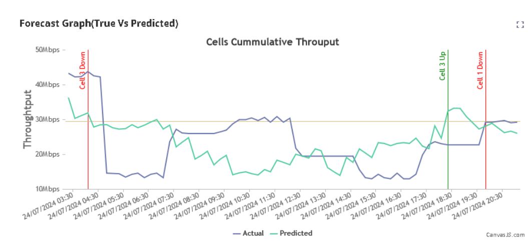

Figure 5: Throughput Forecasts and Decision Making

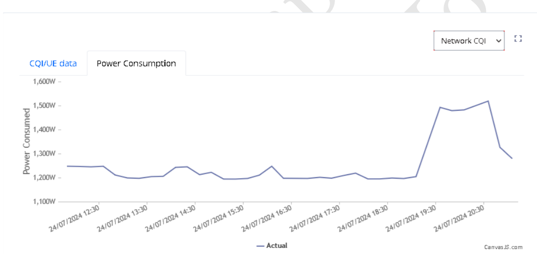

Figure 6: Example of Energy Saving Results

7 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 variosu componenst to implement our solution, from LSTM fopr trafffic prediction and various logical algorithms for other compoennts. The novel part of our proposed solution involved using a Digital Twin to validate the proposed solution in a simulated environment, to prevent causing iussues which cannot be revereseed. A LSTM neural network model was trained on various time-series datasets. [YOGESH] Will complete after results are ready.

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.

REFERENCES

A CELL BRINGUP