IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 June 2024, accepted 24 June 2024, date of publication 1 July 2024, date of current version 16 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3421633

RESEARCH ARTICLE

Exploring Potential of ML-Aided Mobile Traffic Prediction for Energy-Efficient Optimization of Network Resources Using Real World Dataset

ANETA KOLACKOVA[1], SALIH SEVGICAN[2], MUHAMMET FATIH ULU[2], JALE SADREDDINI[3], PAVEL MASEK[1], (Member, IEEE), JIRI HOSEK[1,4], (Senior Member, IEEE), JAN JERABEK[1], AND TUNA TUGCU[2]

[1]Department of Telecommunications, Brno University of Technology, 601 90 Brno, Czech Republic
[2]Department of Computer Engineering, NETLAB, Boğaziçi University, 34342 Istanbul, Turkiye
[3]Department of Computer Engineering, Istanbul Arel University, 34537 Istanbul, Turkiye
[4]Unit of Electrical Engineering, Tampere University, 33100 Tampere, Finland

Corresponding author: Aneta Kolackova (aneta.kolackova@vut.cz)

The research described in this paper was financed by the Technology Agency of the Czech Republic under the grant FW10010014.

ABSTRACT To meet the extremely stringent but diverse requirements of Beyond Fifth-Generation (B5G) networks, traffic-aware adaptive utilization of network resources is becoming essential. To cope with that, a detailed traffic data analysis enables opportunities for mobile network operators to improve the Quality of Service (QoS) in the next-generation mobile communication systems. This paper presents a comprehensive analysis of the real world data collected from an operator's 4G+ and 5G infrastructure during a seven-month campaign. Efficient Machine Learning (ML) based network traffic predictions are presented together with a statistical model to develop optimal resource allocation strategies by using the data gathered during the pandemic, an era when the data volume, as well as the bandwidth requirements and the end users' expectations, were significantly elevated in terms of QoS, given the huge shift to the online world. Data analysis confirmed the assumption that there are traffic changes during the day and the whole week, which helped us to find new research directions regarding resource allocation optimization of next-generation mobile networks. Furthermore, we introduce the Predictive Energy Saver for Baseband Units (PESBiU) algorithm, which utilizes traffic prediction and power consumption analysis to manage the power states (sleep or active) of BBUs in a network. The PESBiU algorithm utilizes the results from ML predictions to effectively balance energy efficiency and network performance, demonstrating its potential for practical deployment in future mobile communication networks by transitioning BBUs to sleep mode during low-traffic periods, thereby achieving significant power savings.

INDEX TERMS Beyond 5G networks, data analysis, machine learning, mobile traffic, resource allocation, BBU energy saving.

I. INTRODUCTION

The rapid growth in cellular technologies, driven by the ever-increasing mobile data volume, is expected to spawn new communication services and play a major role in applications over the air, such as tactile Internet, holographic telepresence, and collective virtual reality (VR), which can be unified under the extended reality (XR) vision. To enable the operation of all devices and to handle their traffic requirements, telco operators have to deploy new solutions and expand their infrastructure by innovating and implementing the mechanisms of Beyond fifth-generation (B5G) such as Cloud Radio Access Network (C-RAN) [1] and Edge Computing [2].

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

With more equipment and technology all around us, carbon emissions are also increasing. Companies from various industries are diligently working on methods to reduce their carbon footprint. Within this widespread effort, the telecommunications industry is deeply involved, striving to achieve "Net Zero" emissions by 2050 through persistent and sustainable actions [3].

Mobile operators aim to reduce their Operating Expenditure/Capital Expenditure (OPEX/CAPEX) while maintaining the Quality of Service (QoS) at the required level. Due to the significance of QoS-aware networks in mobile communication services for both operators and customers, many researchers have conducted studies on their ideal method of QoS implementation [1].

Considering the forthcoming requirements of the B5G networks, it is also necessary to establish QoS-driven resource management schemes and identify appropriate Key Performance Indicators (KPIs) for the Radio Access Network (RAN) as well as the Core network. With the increasing dynamicity that comes with B5G networks due to new emerging applications and services, it is necessary to address this change. To develop such dynamically adopting architectures and to increase the efficiency of network resource allocation while contributing to the minimization of energy consumption, a comprehensive understanding of the network traffic patterns and users' behavior is required. To enable this, emerging Machine Learning (ML) technologies are among promising solutions [4].

The aim of this work is to demonstrate an exploratory ML-aided analysis of a real world mobile dataset, which opens up the path for novel ideas regarding the orchestration of B5G networks in return for higher efficiency in operating costs, lower energy consumption, and increased QoS. To this end, we gathered a real world dataset of network traffic for a period of seven months (2020-08-15 to 2021-02-28) in an average-sized city from a 4G+ and 5G network. The data explained in Section III-A consists of measurements from different mobile sites, where each site is located in a different area in the city and includes several evolved NodeBs (eNodeBs). Considering that the measurement campaign was conducted during the COVID-19 pandemic, we also want to study the impact of such a specific period on the user's behavior and cellular traffic data.

We investigate and discuss this real world dataset. These data served as a source for network traffic prediction results, which we have achieved with the help of Recurrent Neural Networks (RNNs), utilizing gating mechanisms of Long Short-Term Memory (LSTM) unit and Gated Recurrent Unit (GRU). We compare these results with another ML-aided tool, called Facebook's Prophet (FB's Prophet) [5], which uses regression techniques based on the statistical and temporal features of the data.

Furthermore, based on analysis and prediction results, we created Predictive Energy Saver for Baseband Units (PESBiU) Algorithm which can help to improve utilization of resource allocation, and energy efficiency in B5G networks. The emphasis on this topic is rooted in the considerable share of energy consumption in the total expenses of mobile networks. The research referenced here [6] highlights that energy costs account for 23% of all network-related expenditures. Additionally, it points out that the energy usage of the RAN constitutes 73% of the total energy consumption.

The contributions of this paper are summarized as follows:

- A comprehensive analysis of recent real world data from 4G+ and 5G systems to extract and interpret the real cellular data traffic patterns.
- Comparison of ML-aided efficient network data prediction techniques and statistical models to develop optimal resource allocation strategies and minimize energy consumption using the data gathered during the seven-month measurement campaign in a live cellular network.
- Creation of the PESBiU algorithm, which leverages ML predictions to manage the power states of BBUs by transitioning them to sleep mode during low-traffic periods, thus optimizing energy efficiency while maintaining network performance.
- Discussion of proposed methodologies and algorithms as solutions for the major challenges that need to be significantly improved in the B5G networks, such as radio resource optimization and green networking.

II. RELATED WORKS

Much research has been conducted concerning network optimization, resource allocation, and traffic prediction. Feng et al. [7], employ deep learning (DL) techniques for mobile traffic prediction and manage to forecast the 3G/4G traffic with only 6-8% error. Predicting the cellular traffic with high accuracy allows for shutting down underutilized BBUs. However, the dataset used has a few drawbacks. It is from 2014, and the data does not contain separate information about downlink and uplink traffic volumes, which prevents training the traffic volume types and optimizing BBU usage based on downlink and uplink separately.

In [8], the authors focus on in-network pooling framework in B5G era and the attention-based deep reinforcement learning algorithm. In [9], Wu et al. focus on energy-efficient base station switching techniques, while in [10], the authors propose strategies for switching off base stations in heterogeneous networks. However, most of the propositions and calculations come from theoretical assumptions and do not consider the use of real world data for 4G+ and 5G systems and ML techniques. Similarly, in [11], the authors address the dearth of research on per-user traffic analysis in cellular networks and advocate for a shift from traditional cell-aggregated traffic management to a more energy- and cost-efficient user-centric approach, but they artificially generate their dataset.

In [12], the real world data was used. The authors manage to cluster the traffic on BBUs in the network alongside the traffic forecast in order to reduce the costs of C-RAN deployments. Their framework effectively increases

the average capacity utility to 83.4% and 76.7% in two datasets and reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture. However, the authors do not consider the minimization of energy consumption, and their model is trained on a dataset from 2013. For these reasons, there is an important area that requires further investigation and improvement, and that is serving as our motivation. Further, none of the related works utilize recent real world datasets or even data during the COVID-19 pandemic.

On the other hand, the literature is rich in studies using synthetic data concerning power consumption generated via simulators. In [13], the authors select a Remote Radio Head (RRH) based on the traffic density using Efficient Local Search Algorithm (ELSA). Lee et al., in [14] minimize power consumption in both BBU and RRH by threshold-based RRH switching and BBU aggregation. The authors in [15] propose reducing the number of active BBUs and power consumption by switching the BBUs on/off, based on the traffic load, using combined Best Fit Decreasing and Modified Best Fit Decreasing algorithms.

Having in mind the outcomes from the studies discussed above, intelligent data-oriented approaches remain untouched. When it comes to studies including data-oriented approaches, ML can be seen as one of the core approaches for reducing power consumption, optimal resource allocation in the cloud management systems, optimizing network performance and QoS with Software-defined Networking (SDN) and many other studies in the next generation mobile networks.

In [16], the DL methodology called LSTM is proven to work best compared to other linear and non-linear ML algorithms for predicting network traffic patterns. A more recent study presented in [17] runs the LSTM DL algorithm on the Long Term Evolution (LTE) dataset and shows that LSTM is a promising prediction algorithm for mobile traffic. As we can see from the previous works, the main subjects we are interested in are more comprehensive and, therefore, more promising in predicting the cellular traffic pattern.

Consequently, our main subjects are to predict the cellular traffic pattern using ML-based tools with a real world dataset and to minimize energy usage with emphasis on the energy efficiency of the BBU, while maintaining a considerable level of QoS.

III. RESEARCH METHODOLOGY

Let us focus on the measured dataset, which is thoroughly described and explained. Following that, in subsection III-B, implemented ML methodologies are explained.

A. DATASET

The dataset has overall records of the network quality information from 12 different eNodeBs (RRH+BBU) scattered into four cell sites. For 200 days, 4800 log records consisting of hourly observations of mobile data usage have been recorded for each eNodeB. In the coverage of sites, a total of 212,000 users are served with 1.5 terabytes of traffic volume daily on average. Table 1 summarizes the cell site's important physical communication parameters.

TABLE 1. Physical communication parameters.

Parameters	Values
City size	300 km^2
Population density	0.5 mil
Cell type	Macro
Number of cell sites	4 (Neighboring cell sites)
Number of eNodeBs (RRH+BBU)	12
Each eNodeB	6 sectors - 12 cells
eNodeBs of one cell site	700 MHz, 850 MHz, 2100 MHz
Observed time	7 months/every hour
Office area	site A and B - distance 1 km
Residential area	site C and D - distance 3 km

The data is collected from four macro cell sites. Two sites, A and B, are located in the office area with specific antenna deployment on tall buildings' roofs. The other two, sites C and D, are located in the residential area in cell site shelters between residential buildings. Site C is in a densely populated area, than site D, which is more on the edge of the city. The distance between office cell site B and residential cell site C is approximately 3 km, see Fig. 1. The specific locations of the base stations are left unspecified intentionally to protect the commercial benefits of the operator.

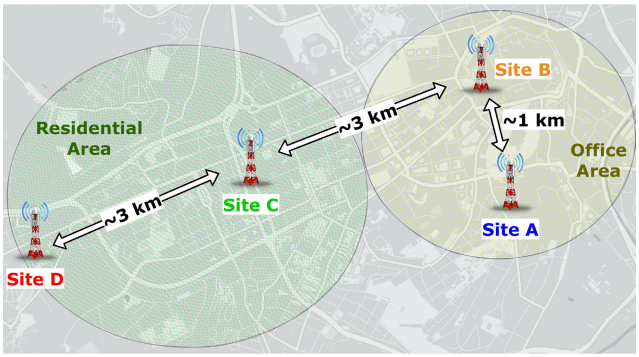


FIGURE 1. Location of macro cell sites - Bird's-eye view.

The data has been recorded through the analytic platform, which monitors the performance of cell sites in the network and other information regarding the eNodeBs (cell site status, number of drops, blocks, power consumption etc.). Each eNodeB sends KPIs to this platform database hourly in the form of raw counters. The collected data contains information about the data volumes in the downlink and uplink directions, the number of connected UEs and packet loss, even for specific QoS Class Identifier (QCI) etc. And especially the power consumption data for each BBU and RRH measured in Watt hours.

B. MACHINE LEARNING ALGORITHMS

The recently introduced and standardized features of the next-generation wireless communications enable operators to gather more data from their mobile networks [18]. In [19], a component called Network Data Analytics Function (NWDAF), which is first introduced with 5G

for ML purposes, is studied, and a proof-of-concept of ML applications for a 5G network with synthetic data is shown. Thanks to real world data, NWDAF can better optimize a B5G network by predicting near-future network traffic, hence utilizing the network resources better by means of reduced power consumption as well as increased QoS via reduced latency over the SDN network. Not only does the NWDAF component contribute to next-generation networking, but it also creates opportunities for the management of B5G networks. For example, an operator can reduce its OPEX and CAPEX by analyzing the data and making future predictions. In [20], recent research about the usage of ML in next-generation networks is discussed, and a detailed inspection of the results that address network efficiency, resource optimization, power allocation, and virtualization is provided. In [12], a comparison of ML methodologies between Artificial Neural Networks (ANNs) and the statistical prediction called Autoregressive Integrated Moving Average (ARIMA) is discussed, showing that ANN outperforms in terms of network prediction purposes. In [16], a comparison among ANN algorithm types shows that LSTM has the minimum Mean Squared Error (MSE) of 0.042 while Feed Forward Neural Network (FNN) has the highest MSE score of 0.091. In [21], the authors show that for network traffic prediction, the accuracy of LSTM is around 90%, which is approximately 10% higher than other DL algorithms. In [16], although LSTM shows more accurate results than GRU in predicting GÉANT backbone network traffic, its running time and computational cost are higher than GRU. Considering the mentioned efficiency of GRU and LSTM, both mechanisms are comparable. Therefore, besides LSTM, a more recent gating mechanism, GRU, is proposed for this work to show how both methods perform in network traffic with our dataset. As the studies discuss, ML is one of the great solutions for the network scalability problems in the future of wireless networking. Hence, working with recent real-life data is crucial to making ML solutions more realistic and keeping them as up-to-date as possible, as network patterns may change.

This study uses two types of RNN, LSTM and GRU, to achieve the best network traffic prediction results based on their proven superior performance among other DL mechanisms, as explained above. The technical reasons for LSTM and GRU's superiority are that both are good at predicting while utilizing historical data. On the other hand, as Cho et al. suggest in the GRU's proposition paper [22], GRU's optimal performance can be achieved in small datasets. Therefore, LSTM tends to perform better than GRU in large datasets. This conditional performance difference can be better understood if we examine the implementation of the LSTM and GRU algorithms. Before moving into implementation details, we should define some fundamental terms, such as gates and identity connection. Gates in Gated Neural Networks (GNNs) are decision-makers for using identity connections or regular stacked layers. The significance of identity connections is that not only do they use the output of consecutive layers, but also they can use the output of much lower layers. Thus, this gate mechanism allows GNNs to learn incrementally, making neural networks safe from long-term dependencies. Two algorithms, LSTM and GRU, are types of RNNs and use a gating mechanism. Besides RNNs, Prophet is used, which is a powerful tool that works for linear or non-linear growth and can help generate accurate and reliable forecasts for time series data with multiple seasonality.

A brief explanation of RNNs, the implementation details of LSTM and GRU, and a description of FB's Prophet are as follows:

1) RNN

RNNs use sequential data, which can be time-dependent, and are a type of ANN. A significant feature of RNNs is that they can utilize previous input to update the current output. The significance of the data at the very beginning can be preserved. RNN usage areas include Natural Language Processing (NLP), image captioning, speech recognition, and language translation. Therefore, using RNN mechanics to analyze sequential mobile network traffic data is appropriate. However, one drawback of Vanilla RNN is that the model's learning can slow down or even stop. Gradients become increasingly small as they are back-propagated to earlier layers during training. This phenomenon can make it difficult for the model to learn long-range dependencies, as the gradients may become so small that they vanish, preventing the network from effectively updating the weights of earlier layers.

One of the main causes of the vanishing gradient problem in RNNs is the repeated multiplication of gradients through the weight matrices during Back-Propagation Through Time (BPTT). If these weight matrices have eigenvalues close to zero, the gradients can quickly diminish as they are multiplied across multiple time steps.

To mitigate the vanishing gradient problem, various techniques have been proposed, including using activation functions that alleviate the saturation of gradients (such as ReLU), using gradient clipping to prevent gradients from becoming too large or too small, and using architectures like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) which are designed to better capture long-range dependencies by controlling the flow of information.

2) LSTM

LSTM is proposed by S. Hochreiter and J. Schmidhuber in [23], the first attempt to solve the vanishing gradient problem by designing an advanced activation function. Additionally, LSTMs are trained using the BPTT method, a variant of backpropagation that allows gradients to flow through the entire sequence, enabling the model to learn long-term dependencies. The representation of a single unit of LSTM is in Fig. 2.

There are three types of gates in an LSTM unit: a forget gate, an input gate, and an output gate. The previous hidden

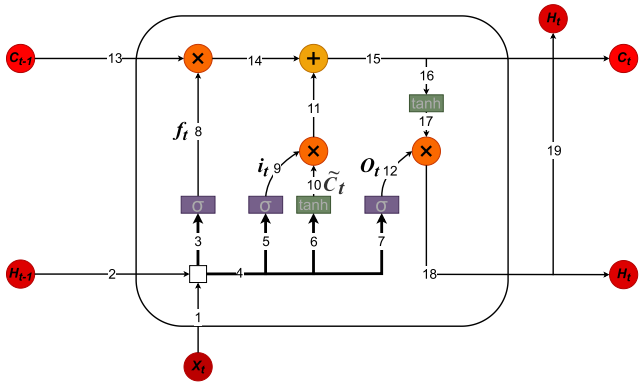


FIGURE 2. A Representation of an LSTM Unit.

state (known as short-term memory) vector (Path 2 in Fig. 2) and the input vector (Path 1 in Fig. 2) concatenate (say *the input&hidden concatenation* for the semantic convenience for this section), and all gates use the input&hidden concatenation.

The forget gate (Path 8 in Fig. 2, abbreviated as f_t) is the result of the sigmoid function (defined as $\sigma(x) = e^x/(1+e^x)$ applied to the summation of the dot product of the forget gate weight matrix and the input&hidden concatenation (Path 3 in Fig. 2) and forget gate bias vector, i.e.,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$

where $\sigma(arg)$ is the sigmoid function that takes argument *arg* and maps the argument between 0 and 1, W_f is the current forget gate weight matrix, $[h_{t-1}, x_t]$ is the interpolation of the previous hidden state vector and input vector, and b_f is the current forget gate bias vector. The semantic responsibility of the forget vector is to decide what and how much to discard or to decide what and how much to remember. Eventually, the previous cell state vector (Path 13 in Fig. 2) is multiplied by the forget vector element-wise (and results as Path 14 in Fig. 2).

Applying the sigmoid function to the summation of the dot product of the input gate weight matrix and the input&hidden concatenation and input gate bias vector results as the input gate (Path 9 in Fig. 2, abbreviated as i_t), which will be then multiplied by the new candidate values vector (Path 10 in Fig. 2, abbreviated as $\tilde{C}_t$).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{2}$$

A similar approach can find the new candidate values vector, $\tilde{C}_t$. (Only the activation function is different, and it is *tanh* (defined as $\tanh(x) = (e^{2x}-1)/(e^{2x}+1)$ now).

$$\tilde{C}_t = \tanh(W_{\tilde{C}} \cdot [h_{t-1}, x_t] + b_{\tilde{C}}). \tag{3}$$

The semantic responsibility of the input gate is to decide what to learn from the new input matrix, X_t. In the input gate-related calculations, the sigmoid function and the *tanh* function are used. *tanh* maps input value *arg* between -1 and 1. Multiplication of i_t with $\tilde{C}_t$ maps the values between -1 and 1 (Path 11 in Fig. 2).

Without calculations of the output gate, the cell state (known as long-term memory) can be calculated (Path 15 in Fig. 2).

$$C_t = C_{t-1} \odot f_t + \tilde{C}_t \odot i_t, \tag{4}$$

where $\odot$ symbolizes the Hadamard product. This step opens the possibility that the boundaries of some cell state vector variables are not limited to -1 or 1. Therefore, the *tanh* function will shrink the cell state in the output gate-related calculations.

Till now, the cell state vector has been updated. What remains is an update of the hidden state vector. The output gate-related calculations update the short-term memory. Similar to the previous calculations, the output gate vector (Path 12 in Fig 2) is:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_O). \tag{5}$$

Connecting this equation with the cell state vector, we have:

$$H_t = O_t \odot \tanh(C_t). \tag{6}$$

Thus, the working principle of all three gates' and candidate values vectors is explained, and selecting LSTM for our case is justified.

3) GRU

The second type of RNN used in this work is GRU, proposed by Cho et al. in [22]. GRU is similar to LSTM in methodology to solve the vanishing gradient problem. However, the key differences make one preferable depending on the machine learning applications, dataset length, etc.

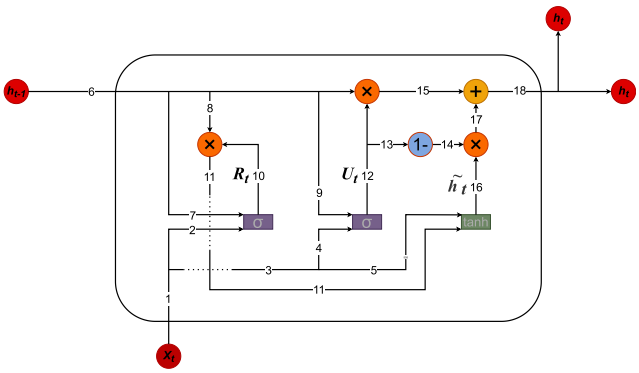


FIGURE 3. A Representation of a GRU.

A single GRU has two types of gates: a reset gate and an update gate. The reset gate (known as short-term memory) decides the amount of negligible past information and ignores it, and it works as the unit's short-term memory. The calculation of the GRU gates has more parameters than any gate calculation in an LSTM unit. Therefore, let us divide these calculations into steps and start with the reset gate. The first step is to find the impact of the input vector (say this impact vector I_i) on the reset gate:

$$I_i = W_r \cdot X_t + b_r, \tag{7}$$

where W_r is the weight matrix of the reset gate for processing the input vector (X_t, there is another weight matrix for the previous hidden state vector, making this calculation a bit longer), and b_r is the bias vector of the reset gate for processing the input vector (there is another bias vector too). The second step is to find the impact of the previous hidden state vector:

$$I_h = \omega_r \cdot h_{t-1} + \beta_r, \tag{8}$$

where ω_r is the weight matrix of the reset gate for processing the previous hidden state vector (h_{t-1}), and β_r is the bias vector of the reset gate for processing the previous hidden state vector. Then, the forget gate vector (Path 10 in Fig. 3) can be calculated as:

$$R_t = \sigma(I_i + I_h). \tag{9}$$

The update gate decides the amount of previous information that is important and provides this information survive. It behaves as long-term memory. The calculation of the update gate is similar to the reset gate. Impact of the input vector (say this impact vector Φ_i) on the update gate:

$$\Phi_i = W_u \cdot X_t + b_u, \tag{10}$$

where W_u is the weight matrix of the update gate for processing the input vector, and b_u is the bias vector of the update gate. The impact of the previous hidden state vector is:

$$\Phi_h = \omega_u \cdot h_{t-1} + \beta_u, \tag{11}$$

where ω_u is the weight matrix of the update gate, and β_u is the bias vector of the update gate. Both are for processing the hidden state vector. Then, the update gate vector (Path 12 in Fig. 3) can be calculated as:

$$U_t = \sigma(\Phi_i + \Phi_h). \tag{12}$$

The first step to calculate the candidate hidden state vector (Path 16 in Fig. 3) is to find the impact of the input vector (say this impact vector τ_i) on the candidate hidden state vector:

$$\tau_i = W_{\tilde{h}} \cdot X_t + b_{\tilde{h}}, \tag{13}$$

where $w_{\tilde{h}}$ is the weight matrix and $b_{\tilde{h}}$ is the bias vector of the candidate hidden state. Both are for processing the input vector. The second step is to find the impact of the reset gate, calculated in the Eq. 9:

$$\tau_r = \omega_{\tilde{h}} \cdot (h_{t-1} \odot R_t) + \beta_{\tilde{h}}, \tag{14}$$

where $\omega_{\tilde{h}}$ is the weight matrix and $\beta_{\tilde{h}}$ is the bias vector of the candidate state. Then, the candidate hidden state vector (Path 16 in Fig. 3) can be calculated as:

$$\tilde{h}_t = \tanh(\tau_i + \tau_r). \tag{15}$$

Lastly, we can calculate the current hidden state vector as follows:

$$h_t = h_{t-1} \odot U_t + \tilde{h}_t \odot (1 - U_t). \tag{16}$$

4) PROPHET

Prophet allows forecasting sequential data depending on time. In the procedure, considerations of daily, weekly, yearly, seasonality, etc. effects are present. Thus, non-linear trends fit these considerations. It is a statistics-based method that works best when the effects are robust.

To compare RNN algorithms with the statistical techniques, we used FB's Prophet prediction. We aim to show that LSTM and GRU outperform statistical techniques when evaluated using real world, up-to-date data and demonstrate a comparison between LSTM and GRU performance-wise.

C. ENERGY CONSUMPTION IN B5G

To accommodate the increasing traffic demands associated with B5G networks, it's imperative to expand capacity, which requires the deployment of more eNodeBs within the RAN segment. However, eNodeBs are not only expensive but also highly energy-intensive. According to survey [6], eNodeBs equipment is responsible for 50% of the total energy consumption in RAN, with air conditioning systems following closely at 40%. These cooling systems are essential to prevent eNodeBs from overheating. Therefore, reducing the energy consumption of eNodeBs and enhancing their energy efficiency emerges as a primary strategy for lowering overall energy usage.

1) ENERGY CONSUMPTION OF BASEBAND UNIT (BBU)

As was mentioned in III-A, eNodeB has two parts - BBU + RRH. The BBU handles the baseband processing tasks, while the RRH is responsible for converting BBU signals into radio signals (and vice versa) to facilitate transmission or reception through antennas. Typically, a single BBU collaborates with several RRHs to manage the processing of radio signals. This implies that the power consumption of a BBU escalates with the number of connected RRHs and the volume of signaling required, which varies according to traffic demands [3].

From a hardware viewpoint, the energy consumption of a BBU varies based on the design of its components. The types of processors and modems incorporated into a BBU play a significant role in the overall energy efficiency. Given that BBU equipment predominantly consists of digital components, the specific architecture or design chosen for a BBU can result in varying levels of energy usage. Additionally, certain BBU models offer advanced power-saving capabilities, which are influenced by their physical form factor and the design of their chipsets.

Considering it from software perspective in B5G networks, energy saving strategies for BBUs have centered around techniques like Dynamic Scaling, Sleep Modes and Cloud-Based and Virtualized BBUs. Dynamic Scaling involves adjusting the BBU's processing capacity to match the current traffic load, potentially powering down unneeded resources during low usage periods. Sleep Modes, another tactic, entail putting BBUs into a low-power state during off-peak times, which can drastically cut energy use, provided it's done without negatively impacting network

service. Transitioning BBUs to cloud-based or virtualized settings enables more efficient resource use network-wide, minimizing the necessity for physical infrastructure and thus conserving energy. These virtualized BBUs can be adjusted dynamically to match demand, enhancing energy efficiency. Additionally, employing Artificial Intelligence (AI) and ML offers a forward-thinking method by predicting traffic flows and adjusting BBU operations in real-time for optimal energy savings. This article highlights the use of a combined approach to improve energy efficiency in B5G networks. It illustrates a strategy to reduce energy consumption in BBUs, showcasing innovative solutions for sustainable network management [3].

In real-world commercial networks, traffic generation fluctuates between full and empty loads, influenced by factors such as time, user count, and coverage area. Leveraging this dataset, we can correlate the actual traffic load with energy consumption, enabling us to develop an algorithm aimed at enhancing the energy efficiency of BBUs.

2) ENODEB'S FORMULATION OF ENERGY CONSUMPTION

The ratio of energy consumption between BBUs and RRHs depends on used technology, cell load, weather conditions, temperature etc. In [3], SK Telecom reports that the amount of power consumed by BBU is 22% whereas that by RRH is 78%, which is almost 1:4. According to our measurements, from the data set, the ratio is 1:5. Generally, RRHs contribute more significantly to the overall energy consumption within a network than BBUs. On the other hand, BBUs are generally underutilized because of the dynamic behavior of network traffic, as will be shown later in this paper. As a result, it is critical to address these issues by leveraging the complex characteristic of traffic for improving the energy efficiency of BBUs at low traffic hours.

To compute the weighted average of power/energy consumption, weighting factors are applied according to a daily (24-hour) traffic load distribution profile, which includes three load levels: low load (low), medium load (med), and high load (high) load. This profile is characterized by the duration, in hours, of each load level, denoted as t_{low}, t_{med}, and t_{high}, respectively. Operators have the flexibility to establish a load distribution profile that accurately mirrors the network's conditions and require that this specific profile be utilized in calculations. The average power consumption [W] of central and remote parts defined as [24]:

$$P_{eqp} = P_{BBU} + P_{RRH}, \tag{17}$$

in which P_{BBU} and P_{RRH} [W] are average power consumption of central and remote parts defined as:

$$P_{BBU} = \frac{(P_{high,BBU} \cdot t_{high} + P_{med,BBU} \cdot t_{med} + P_{low,BBU} \cdot t_{low})}{(t_{high} + t_{med} + t_{low})} \tag{18}$$

$$P_{RRH} = \frac{(P_{high,RRH} \cdot t_{high} + P_{med,RRH} \cdot t_{med} + P_{low,RRH} \cdot t_{low})}{(t_{high} + t_{med} + t_{low})} \tag{19}$$

IV. ANALYSIS AND EXPERIMENTAL RESULTS

ML algorithms require that the raw data is cleaned and processed in such a way that the algorithm would perform with minimal error during training and testing. In our study, we utilized Python for data cleaning, preprocessing, and the implementation of machine learning models.

A. PRE-PROCESSING OF DATA

The dataset used in this study and explained in Section III-A consists of traffic volume, throughput, and packet loss in the mobile network where these parameters are KPIs of B5G networks [25]. Parameters that are out of the scope of this study, such as call duration, are discarded. The data samples are taken every hour from eNodeBs. However, due to the occasional maintenance periods in the network, it was not possible to record some data points on eNodeBs. To prevent bias and misinterpretation, the daily observation periods when the eNodeBs were under maintenance are removed, which corresponds to 0.80% of all records.

Secondly, the data has been split into training and testing datasets. The training part is approximately 75% of the overall dataset, while the testing dataset is about 25%. The testing dataset is required to confirm the robustness of the trained algorithm. The ML models explained in Section III-B have been trained using training data, and the results are given by calculating the difference between the actual testing data and the predictions done by the trained ML algorithm. The training and testing datasets have been scaled using a standard, common algorithm before training a DL algorithm.

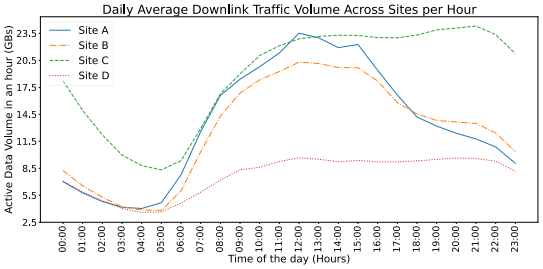
After pre-processing, we train the algorithms explained in Section III-B and discuss the results in the following section.

B. DATA ANALYSIS

The operation of a mobile network has a dynamic nature. Over time, the patterns can vary significantly, which makes them hard to predict, especially in a short-term perspective. Therefore, the data gathered in a mobile network can exhibit temporal and spatial variations.

In our study, we have gathered and categorized the data under four different cell sites, namely sites A, B, C, and D, as explained in Section III-A. Sites A and B correspond to the region where offices and business centers are located. Sites C and D correspond to the residential areas (see Fig. 1). Data analysis shows that sites A and B have similar characteristics through time, and the same goes for sites C and D. Due to the similarities among the sites, we explain the figures of sites A and C as representatives of the business and the residential districts and discuss them together with sites B and D.

Fig. 4 shows a comparison of uplink and downlink traffic volumes across all site cells. As shown, the processed data indicate diverse characteristics. In the case of site C (green line), the traffic volume pattern surpasses the other sites by 179%, which results in a 7.5 GB difference on the average per day and leads to higher utilization of site C. Another indication is that site C (green line) and site D (red line) have



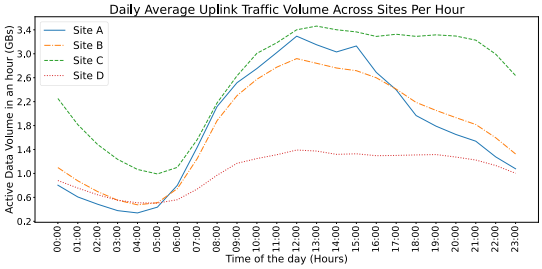


FIGURE 4. Daily avg. DL and UL traffic volume.

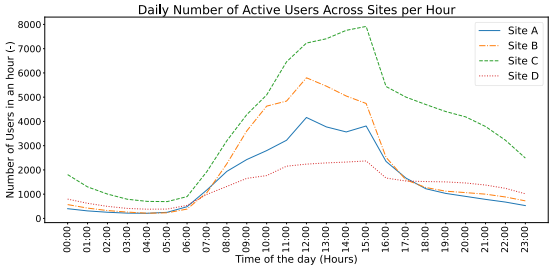


FIGURE 5. Daily number of active users.

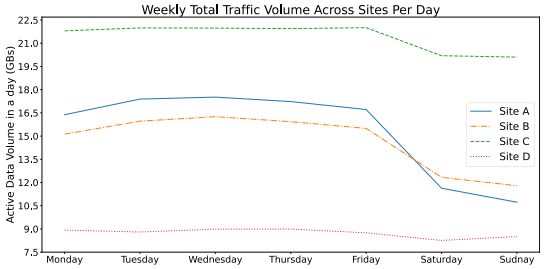


FIGURE 7. Average traffic volume on sites through a week.

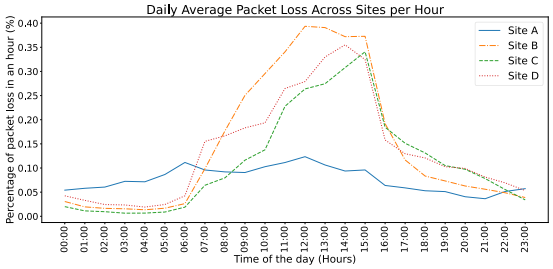


FIGURE 6. Daily average packet loss in the network.

different traffic volume trends than site A (blue line) and site B (orange line) after about 2:00 p.m. Comparing the traffic volume by the end of the day, after 4:00 p.m., the sites located in the office area (A and B) have a 52 % traffic volume drop. Meanwhile, the sites in the residential areas (C and D) have a 10 % drop in traffic volume.

In Fig. 5, the average number of active users per site is shown. This figure has similarities with the daily traffic volume in Fig. 4. This similarity is expected since a high number of users registered on the cell site could generate higher traffic volume in the area. One data characteristic in sites C and D is unique to residential areas: The number of active users towards the evening does not show a steep decline as in sites A and B, but the number goes down until 5:00 a.m of the following morning. Despite a shallow declination in the number of active users, the traffic volume of sites C and D does not show a similar pattern as the number of users, which shows that at this time period of the day, after 4:00 p.m., the amount of data traffic per device is much higher than average. Therefore, we can conclude that the traffic volume stays almost the same for the rest of the work shift hours.

We also look at the packet loss parameter's behavior to understand the cell sites' reliability and ease of connectivity. In our investigation, we average the packet loss values over time and remove the effects of packet loss spikes, such as data anomalies (values beyond the possible boundaries, i.e., values taken on extreme fluctuations). However, information about network reliability can be obtained by examining the packet loss inspection figure, which comprehends the whole time period. In Fig. 6, we observe the packet loss parameter follows a similar pattern as traffic volume data. However, if we consider cell sites individually, there are some conclusions to make. When sites C and D are compared, despite the low number of active users in site D, packet loss is higher than in site C, which is a cell site from a residential area as well. Site B's packet loss reaches much higher values at the beginning of the work shift around 8:00 a.m. and continues to have higher values than the other sites when compared. Aside from daily statistics, weekly changes in traffic volume are also investigated. In Fig. 7, we compare the total traffic volume across all sites. The typical behavior is that the total traffic volume shows a decreasing trend on weekends. Site cells in residential districts have gentle decline patterns compared to the site cells located in business districts.

C. RESULTS OF ML-BASED NETWORK TRAFFIC PREDICTION

With the insights we obtained from Section IV-B, we applied three different algorithms for network traffic volume prediction. Implementing three different algorithms allows us

to compare them. The comparison may enable the selection of the optimum method under similar circumstances. First, considering the temporal characteristics of the data, we apply FB's Prophet algorithm that makes predictions by using temporal features such as seasonality, trend, and holiday effects. The second approach is implementing a neural network consisting of multiple layers of LSTM neurons. We have a neural network structure consisting of multiple GRU neuron layers as the third and last approach. Further, we discuss the implementation insights and pairwise comparisons of the three different algorithms.

1) FB'S PROPHET

As discussed in Section III-B, the authors in [5] created a library to generalize processing implementations and predict time-series data. This library is popular because it helps researchers make predictions based on the data's statistical features by decomposing the time series into its seasonality and trends while taking the holidays and similar time-specific features into account. The method predicts the value of a function at time t using a trend component and an error term. A trend component can also be modeled as a linear function. Moreover, Prophet uses the Fourier series to model different types of modeling, allowing flexibility in this area. Prophet predicts the posterior distribution of parameters, where Bayesian inference is used. Also, the Markov Chain Monte Carlo (MCMC) method is utilized in this part of the algorithm.

2) LSTM

In Section II, we have discussed the studies that proved LSTM is one of the best practices when it comes to making predictions on time series data. With this motivation, we apply an LSTM neural network consisting of seven layers together with input and output layers. Layers are added sequentially with different dimensionalities of output spaces. Linear and Exponential Linear Unit activation functions are used to construct the layers, and three of five hidden layers use linear activation functions. The remaining two layers use exponential linear unit activation functions. To initialize weights with statistical function, our choice as a kernel initializer is the normal distribution for every layer. We choose the MAE loss function and Adaptive Moment Estimation Algorithm (ADAM) optimizer for the LSTM network to minimize prediction errors. The parameters and configuration we have chosen for the LSTM algorithm constitute an initial prediction model, which does not heavily lean on the problem's complexity (in our case, the traffic volume prediction). An example of LSTM predicting the traffic volume on January 26, 2021, can be seen in Fig. 10.

3) GRU

We have explained the GRU's mechanism and referred to some papers that show GRU's significance in predicting sequential data. Therefore, we construct a GRU neural network consisting of six hidden layers. Six hidden layers are experimentally selected as optimum. Having more layers did not bring any considerable accuracy improvement in predicting the test inputs but increased overall memory consumption. Having fewer layers decreased accuracy, and despite that, it lessened overall memory consumption; losing accuracy is not preferred. Further, a batch normalization layer, an input, and an output layer. Our implementation mainly has two differences between LSTM-aided neural network and GRU-aided neural network: the first difference is the choice of GRU, and the second one is that we add a batch normalization layer. The remaining parts of the implementation are the same.

TABLE 2. Network traffic volume predictions for downlink.

Model	Metric	Site A	Site B	Site C	Site D
3*FB's Prophet	MAPE (%)	29.32	22.85	**13.46**	21.76
	MAE (MBs)	3687.24	2162.10	2447.59	**1485.08**
	Accuracy (%)	70.68	77.15	**86.54**	78.24
3*LSTM	MAPE (%)	7.01	10.04	**6.11**	6.57
	MAE (MBs)	844.31	869.26	1107.10	**449.28**
	Accuracy (%)	92.99	89.96	**93.89**	93.43
3*GRU	MAPE (%)	10.03	11.07	7.98	**7.79**
	MAE (MBs)	1197.33	1044.07	1328.82	**546.60**
	Accuracy (%)	89.97	88.93	92.02	**92.21**

TABLE 3. Network traffic volume predictions for uplink.

Model	Metric	Site A	Site B	Site C	Site D
3*FB's Prophet	MAPE (%)	62.49	31.78	**12.11**	32.54
	MAE (MBs)	844.49	448.39	**293.09**	374.35
	Accuracy (%)	37.51	68.22	**87.89**	67.46
3*LSTM	MAPE (%)	9.90	11.81	**7.10**	9.66
	MAE (MBs)	164.42	163.59	178.99	**107.28**
	Accuracy (%)	90.10	88.19	**92.90**	90.34
3*GRU	MAPE (%)	12.25	21.83	**8.40**	16.23
	MAE (MBs)	194.50	248.77	216.14	**167.68**
	Accuracy (%)	87.75	78.17	**91.60**	83.77

Prediction results of FB's Prophet, LSTM, and GRU are given in Tables 2 and 3. The Mean Absolute Percentage Error (MAPE)[1] and Mean Absolute Error (MAE)[2] results of FB's Prophet prediction in Tables 2 and 3 show that the error for downlink is lower than that for uplink. This observation follows the fact that the downlink traffic is more predictable by using statistics. Therefore, uplink traffic has a higher standard deviation and is less predictable.

The results are provided by taking the average of ten trials (i.e., using different numbered epochs and/or different neural network architecture) since the performance of AI prediction changes based on those parameters. The comparison between pairs of FB's Prophet, LSTM, and GRU is made by taking the lower one as the reference point. The calculation is made by $100 \times (A_i - A_j)/A_j$, where A stands for ith or jth accuracy, provided that A_i is greater than A_j. Further, A_is can be found with $100 - \text{MAPE}_i$. For instance, by checking Table 2, the accuracy of LSTM for site A is $100 - 7.01 = 92.99$ (%), whereas the accuracy of FB's Prophet is $100 - 29.32 = 70.68$ (%). Then, using the calculation formula,

[1]MAPE represents how much the prediction deviates from the actual data in terms of percentage.

[2]MAE calculates the difference between the prediction and the actual data using the sum of absolute differences in terms of megabytes.

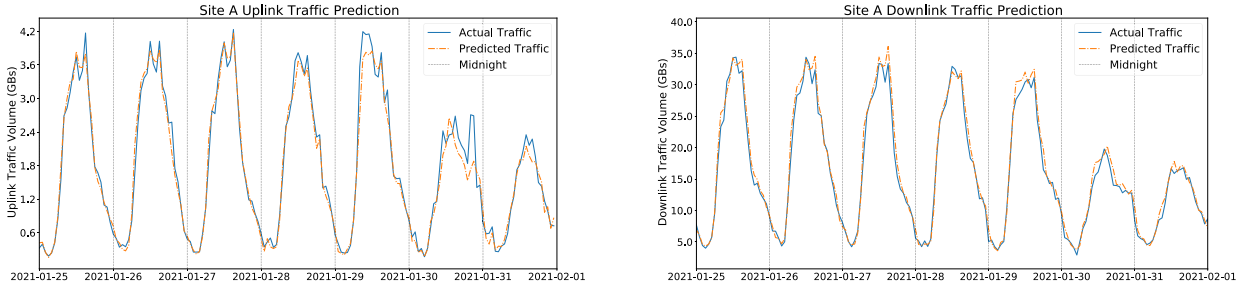


FIGURE 8. A sample week of uplink and downlink traffic prediction for site A using LSTM.[3]

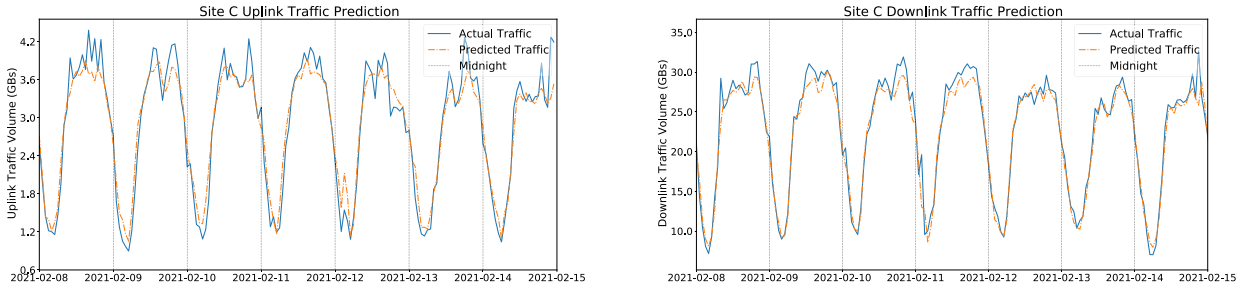


FIGURE 9. A sample week of uplink and downlink traffic prediction for site C using LSTM.

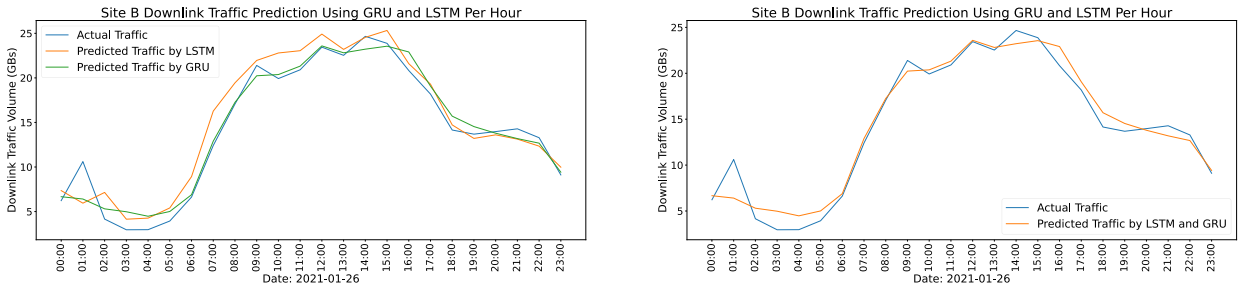


FIGURE 10. a: LSTM's and GRU's daily mobile traffic prediction for site B. b: LSTM and GRU implemented together.[3]

we have $100 \times (92.99 - 70.68)/70.68 = 31.56$ (%), meaning that LSTM is 31.56% more accurate than FB's prophet at predicting network traffic volume for downlinks. Further accuracy calculations are made depending on this evaluation.

As apparent, there is no perfect correlation between LSTM, GRU, and FB's Prophet. The most predictable place for all methods is site C, which is in a residential area. One reason for better prediction is the fact that site C shows a low packet loss. Having low packet loss increases predictability for LSTM, GRU, and Prophet algorithms. Some other reasons can be thought of as due to the nature of machine learning. Therefore, AI methods, namely LSTM and GRU, show better performance. Yet, FB's Prophet benefits the most from having low packet loss. Site A uplink traffic is predicted with less than 50% accuracy by FB's Prophet, which shows the deficiency of the statistical methods. Further, Prophet is considerably accurate where the dataset is robust. There may be data anomalies that could not be detected by our own method. LSTM and GRU perform with about 10% and 12% MAPE error rates in network traffic volume predictions for uplink, respectively. It is important to note GRU's behavior for site B and for the uplink case. If we take the reference point as site A, GRU's error rate increases by 78.2% for site B. Also, from site B to site C, GRU's error rate decreases by 61.5%. One particular reason for this situation can be that learning from datasets of early days or months is useful for later cases, and with GRU, the importance of early processed data does exist, but not as much as LSTM. Further, this is consistent with the features of GRU and LSTM. It is important to note that even though DL algorithms are performed multiple times and their

[3]Result data and other materials available at: https://vutbr-my.sharepoint.com/:f:/g/personal/xkolac15_vutbr_cz/EosEzyevAUtImgN-0Pbg0-8BLtNgcmpltKmjfKqCJqAikQ?e=FzpZyh.

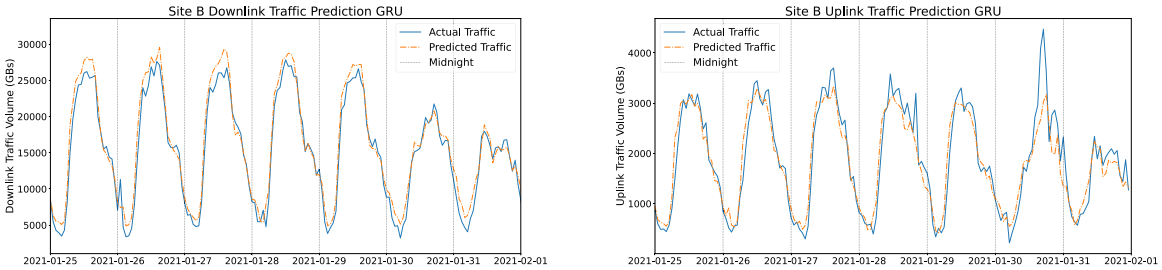


FIGURE 11. A sample week of downlink and uplink traffic prediction for site B using GRU.

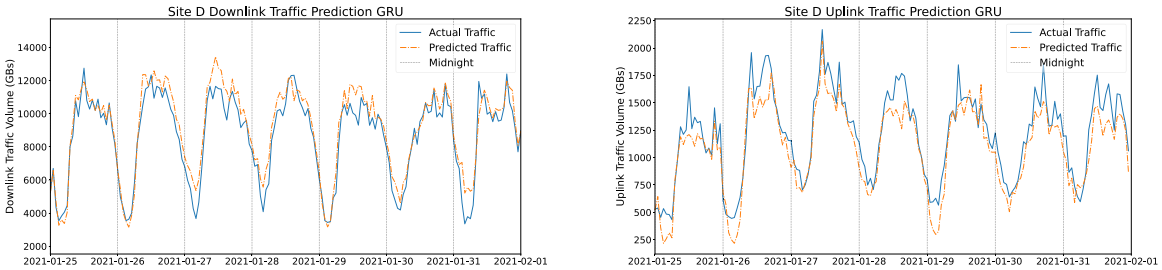


FIGURE 12. A sample week of downlink and uplink traffic prediction for site D using GRU.

average is recorded, the deterministic results are still not guaranteed.

On average, LSTM is 19.02% for downlinks and 52.27% for uplinks more accurate than GRU at predicting the traffic volume in the network. It can be seen that LSTM outperforms FB's Prophet algorithm since LSTM does not follow a basic methodology such as just taking statistical features into account. Evaluating LSTM performance across the sites, site B has the lowest prediction performance. This result is expected as discussed in Section IV-B; the higher packet loss percentage and the possibility of the presence of highly mobile users in the area introduce difficulty in providing accurate predictions. Algorithms that predicted sites C and D have relatively better performance for downlink traffic volume due to their more stable daily average traffic, as seen in Fig. 4. However, we do not observe similar performance for the downlink for sites C, D, and other cell sites. The uplink traffic has a higher standard deviation than downlink and is more prone to anomalies, which is challenging even for a complex ML algorithm. This deviation and prediction performance of LSTM is apparent when one compares the uplink figures Fig. 8 and Fig. 9, respectively. Also, GRU prediction results for a sample week of downlink and uplink are visualized in Fig. 11 and Fig. 12. Thus, one can see that except for spikes in actual mobile traffic, GRU performs well in predicting traffic.

GRU performs better in the comparison of GRU and FB's Prophet. GRU is 16.69% more accurate than FB's Prophet in estimating downlink network traffic volume, whereas, for uplink, GRU is 44.23% more accurate. This is an expected result since the GRU structure is also based on more complex operations than FB's Prophet, rather than considering only statistical properties. GRU's weakest prediction occurs in site B, with a ratio of 11.07% MAPE error. This is similar to the LSTM case: packet losses weaken the prediction accuracies. In comparing LSTM and GRU, LSTM shows slightly better performance than GRU. LSTM is 1.97% more accurate in predicting downlink network traffic volume. For the uplink, this accuracy difference increases to 6.19%.

Overall, LSTM performs better for uplink and downlink network traffic than traditional statistical models and GRU. LSTM yields promising results for its utilization in next-generation wireless networking. The future ideas that this study can contribute to are discussed in Section VI.

Despite promising results with these prediction methodologies, algorithms still have room for improvement. Prediction performance is the critical indicator of progress. As more insights and different types of data are gathered from mobile networks, prediction performance can be improved if the data properly explains the variations in the traffic volume. Another perspective for improvement is to increase the complexity and depth of the trained ML model. However, one should be aware of the risks of an over-trained model, which is beyond the scope of this paper.

Additionally, in Fig. 10b, LSTM and GRU are implemented together to compare the combination of LSTM and GRU, only LSTM, and only GRU. The results are similar to those discussed separately in Tables 2.

V. PREDICTIVE ENERGY SAVING FOR BASEBAND UNITS USING TRAFFIC FORECASTING

In comparison to the existing strategies detailed in Section III-C, the PESBiU algorithm introduces a method to predict traffic volume for BBUs. By forecasting when traffic will reach specific thresholds, PESBiU can proactively transition BBUs into sleep mode before they would do

so autonomously. This approach significantly enhances the optimization of BBU energy management.

Our traffic prediction methodology is based on real-world data, unlike other studies that rely on synthetic datasets. This, combined with energy consumption data, allows us to validate the correlation between traffic volume and energy usage.

Through analyzing network traffic patterns, we observe that data transfer volumes tend to decrease at predictable times, such as between 3:00 am and 5:00 am and in business districts during weekends. During these periods, we can turn off some BBUs to save energy. As detailed in Section V-A, this strategy effectively reduces power consumption. The process involves redistributing the load from BBUs operating under low load conditions to others with available capacity, allowing some units to be shut down.

To demonstrate the efficacy of our PESBiU algorithm, we conducted a case study on site B, to illustrate traffic patterns and to quantify the energy savings achievable with PESBiU. We opted for the LSTM model for traffic prediction due to its superior performance. The decision was based on comparative analysis: LSTM achieved a MAPE of 8.52 %, outperforming both the GRU model, which had a MAPE of 11.95 %, and FB's Prophet, with a much higher MAPE of 28.28 %. Furthermore, LSTM has slight advantage over GRU, with a 3.9 % better ratio, solidified its selection for achieving more accurate traffic predictions.

A. POWER CONSUMPTION BEHAVIOR

This section presents a detailed analysis of power consumption behavior at site B, located in the city center. Like all macro cell sites examined in this study, site B comprises three BBUs. This site was selected to illustrate the relationship between Average Power Consumption [W] and Consumed Energy [Wh] with Traffic Volume [GB] in both downlink and uplink, as derived from Key Performance Indicator (KPI) values in the dataset.

By accurately predicting traffic per BBU, we can estimate the corresponding energy consumption due to the established correlation between traffic volume and power usage. This relationship underpins the development of the PESBiU.

According to the technical specifications of a BBU, the maximum theoretical throughput is 1200 Mbps for downlink (DL) and 600 Mbps for uplink [26]. Which corresponds to 540 DL / 270 UL GB of traffic volume per hour. These values do not correspond to the achieved traffic volume on the selected cell sites. However, as mentioned in III-C such discrepancies are common in real-world environments, where BBU utilization can vary significantly. The PESBiU algorithm accounts for these variations by adjusting thresholds based on the specific load of each cell site.

It is important to note that while all BBUs utilize the same technology, variations such as the number of connected RRHs, microchip differences, or connection types can affect power consumption. This variability is evident at site B, where despite BBU_1 having the highest power consumption, it handles the lowest traffic volume. Nevertheless, traffic flow and power consumption remain correlated, as shown in Figures 13 and 14. The summary of observed values can be seen in the Table 4, it summarizes total traffic volume, average traffic volume, average power consumption, and consumed energy for the BBUs.

The specific power consumption characteristics of each BBU, determined by setup and hardware, ensure that BBU_1's higher power usage despite lower traffic volume does not impact the PESBiU algorithm's decision-making process. The algorithm prioritizes traffic volume predictions using the LSTM model, which inherently aligns with power consumption. However, these specific power consumption characteristics can be utilized in future enhancements of the PESBiU algorithm for even more precise energy optimization.

TABLE 4. One selected week of power behavior for each BBUs.

	Total traffic volume [GB]	AVG traffic volume [GB]	AVG Power Consumption [W]	Consumed energy [Wh]
BBU_1	1690	10.06	**136.77**	**5747.75**
BBU_2	**2530**	**15.05**	114.51	4812
BBU_3	1701	10.12	117.70	4951.5

Fig. 13 and 14 illustrate the average power consumption and consumed energy, respectively, in relation to traffic volume over a week. The graphs display data from Monday to Sunday, showing the correlation between traffic volume and power usage for each BBU. Fig. 15 focuses on a single day, Tuesday, to demonstrate the algorithm's application in a daily context, later in this paper.

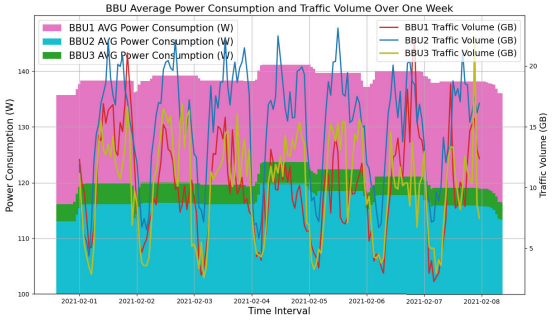


FIGURE 13. BBU's Avg Power Cons. with Traffic Volume - 1 Week, siteB.

For clarity, Fig. 15 presents a detailed view of the data for Tuesday, highlighting the specific power consumption and traffic volume correlation for that day and it will be referenced later in subsection V-C.

B. PREDICTIVE ENERGY SAVING FOR BASEBAND UNITS (PESBIU) ALGORITHM

In this section, we describe our created algorithm called Predictive Energy Saver for Baseband Units (PESBiU). Where we combine our finding regarding the traffic prediction and

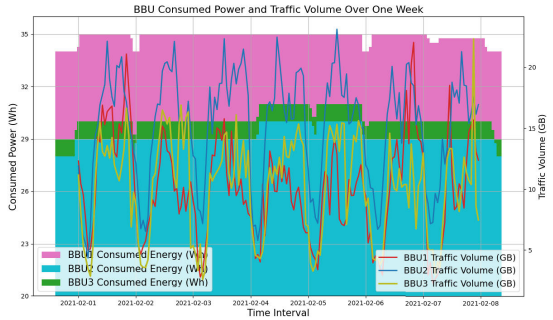


FIGURE 14. BBU's Consumed Energy with Traffic Volume - 1 Week, site B.

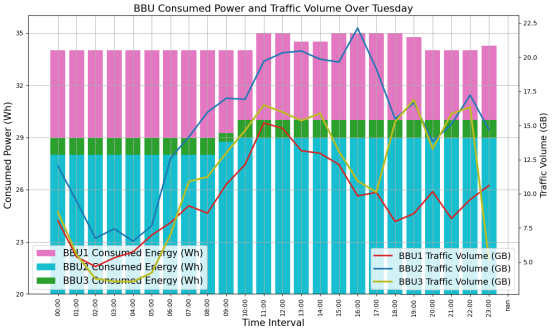


FIGURE 15. BBU's Consumed Energy with Traffic Volume - 1 Day, site B.

power consumption. The PESBiU algorithm is designed to manage the power states (sleep or active) of BBUs in a network based on predicted traffic to save energy while ensuring network performance.

Before describing the PESBiU algorithm, it's essential to understand the key notations and definitions used. These mathematical notations explain the parameters and conditions upon which the algorithm operates.

1) NOTATIONS AND DEFINITIONS

- $BBUs$: Set of BBUs $\{BBU_1, BBU_2, BBU_3\}$
- C_i: Capacity of BBU_i based on analysis
- T_{total}: Total traffic capacity, $T_{\text{total}} = \sum_{i=1}^{3} C_i$
- T_{pred}: Predicted total traffic
- T_{pred_i}: Predicted traffic for BBU_i
- T_{actual}: Actual total traffic
- T_i: Current traffic of BBU_i
- Status$_i$: Status of BBU_i, can be "active" or "sleep"

2) RULES FOR TRAFFIC LEVELS

The algorithm uses the predicted total traffic and individual BBU traffic predictions to classify the traffic levels, which influences the decision on which BBUs can be put to sleep.

- Traffic Level for Cell Site - the overall predicted traffic level, which are derived from [24] (can also be changed, if necessary, on the basis of the provider's request), is categorized as follows:

$$\begin{cases} \text{Low Traffic} & \text{if } T_{\text{pred}} < 0.3 \times T_{\text{total}} \\ \text{Medium Traffic} & \text{if } 0.3 \times T_{\text{total}} \leq T_{\text{pred}} \leq 0.7 \times T_{\text{total}} \\ \text{High Traffic} & \text{if } T_{\text{pred}} > 0.7 \times T_{\text{total}} \end{cases}$$

- Traffic Level for Each BBU - each BBU's predicted traffic level is classified as:

$$\begin{cases} \text{BBU.Low Traffic} & \text{if } T_{pred_i} < 0.3 \times C_i \\ \text{BBU.Medium Traffic} & \text{if } 0.3 \times C_i \leq T_{pred_i} \leq 0.7 \times C_i \\ \text{BBU.High Traffic} & \text{if } T_{pred_i} > 0.7 \times C_i \end{cases}$$

3) SAFE SWITCH MECHANISM

- Traffic Difference Monitoring - to ensure stability and avoid unpredicted changes, the algorithm monitors the difference between actual and predicted traffic:

$$\Delta T = \left| \frac{T_{\text{actual}} - T_{\text{pred}}}{T_{\text{pred}}} \right|$$

- Threshold for Safe Switch - if the relative traffic difference ΔT exceeds 10%, all BBUs remain operational:

$$\Delta T > 0.1 \implies \text{All BBUs remain operational}$$

4) INTEGRATION WITH PESBIU ALGORITHM

The PESBiU algorithm, see Alg. 1, effectively utilizes the defined notations and traffic level classifications to manage the power states of BBUs for optimized energy usage. It starts by predicting the total and individual BBU traffic (T_{pred} and T_{pred_i}), and classifies the overall traffic into Low, Medium, or High levels based on defined thresholds. Depending on the traffic level, it selects BBUs with the lowest predicted traffic for potential sleep mode, setting wake-up thresholds at 30% of their capacities. The algorithm ensures that other active BBUs can cover the traffic of those selected for sleep without exceeding their wake-up thresholds, thanks to the implemented redistribute function. If the difference between actual and predicted traffic ΔT exceeds 10%, all BBUs remain operational to prevent unpredicted changes in network. Finally, the algorithm wakes up any BBUs if their traffic exceeds the wake-up thresholds, ensuring a balance between energy efficiency and network performance.

C. EVALUATION OF PESBIU ALGORITHM

To validate the PESBiU algorithm, we implemented it on a specific day within the selected week to evaluate potential energy savings. Tuesday was chosen for this analysis, as illustrated in Fig. 15, which utilizes values obtained through a standard function. The predicted data for the simulation were generated using LSTM recurrent neural networks. By forecasting future network traffic load and applying the PESBiU algorithm, we were able to estimate the necessary number of BBUs required to operate the network efficiently without underutilizing available BBUs.

Our observations included instances where one BBU was offline, and the remaining two BBUs had to compensate by covering its data traffic. This scenario enabled us to determine a coefficient representing the increase in energy consumption due to traffic redistribution, set at 0.05 for every 1 percent increase in covered traffic. This coefficient was incorporated into our simulation to adjust the calculated energy consumption accurately.

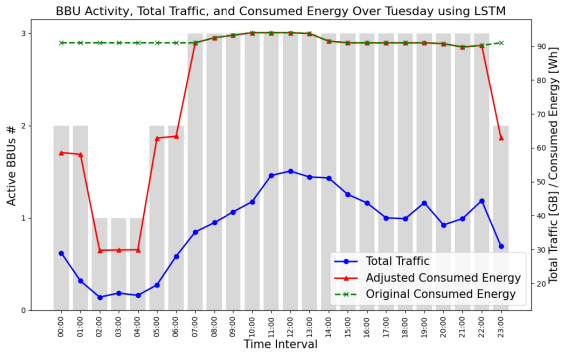


FIGURE 16. Required number of BBUs prediction by LSTM and PESBiU.

The graphical results in Fig. 16 demonstrate that between 2:00 AM and 5:00 AM, it was possible to put two BBUs into sleep mode, reducing energy consumption from 273 Wh to 90 Wh during that 3 hours period. For the subsequent five hours, the network operated with only two BBUs. Overall, the application of the PESBiU algorithm resulted in an energy saving of 332.35 Wh for 24 hours, which corresponds to a 15.12 % reduction for the entire day. This estimation is based on an assumed BBU capacity of 25 GB per hour, derived from weekly observations.

VI. OPEN ISSUES AND FUTURE RESEARCH PLAN

As discussed in Section I,machine learning (ML) offers significant advantages for next-generation mobile communication networks. Our analysis of recently gathered real-world data demonstrates promising results in predicting short-term network behavior, which has various applications for enhancing network efficiency.

This paper also explores additional potential research directions. Some of these are partially integrated into our PESBiU algorithm, particularly in the context of BBU load balancing during the validation check for covering BBUs. In this process, traffic is redistributed to avoid reaching critical thresholds.

1) C-RAN BBU LOAD BALANCING

In a C-RAN architecture where BBUs are in the same BBU pool, load balancing can be investigated in terms of assigning appropriate BBU resources within a pool, based on data volume prediction. Load balancing of BBU pools is crucial to determine the QoS of the network. Arranging utilization of virtual BBUs in C-RAN does affect the waiting time of users while performing an action that will use mobile traffic, like latency and jitter in the network, and many other QoS parameters [1]. Since the developed ML algorithm is able to predict data volumes for each BBU, one can even consider BBU pools with different BBU capacities, which results in better utilization. Considering the traffic load prediction results in this study, the BBU pool capacity can be dynamically adjusted based on the actual needs so the network resources and QoS can be maintained more efficiently.

Algorithm 1 PESBiU algorithm

Input: T_{total}, T_{pred}, C_i for $i = 1, 2, 3$
Output: Updated statuses of BBUs ('sleep' or 'active')
Detecting Opportunity:
$T_{\text{pred}} = \text{PredictTotalTraffic}()$
$T_{\text{pred}_i} = \text{PredictTrafficForBBU}(i)$
$$\text{TrafficLevel} = \begin{cases} \text{L} & \text{if } T_{\text{pred}} < 0.3 \times T_{\text{total}} \\ \text{M} & \text{if } 0.3 \times T_{\text{total}} \leq T_{\text{pred}} \leq 0.7 \times T_{\text{total}} \\ \text{H} & \text{if } T_{\text{pred}} > 0.7 \times T_{\text{total}} \end{cases}$$
if TrafficLevel = L **then**
    Identify BBUs i_1, i_2, i_3 such that $T_{\text{pred}_{i_1}} \leq T_{\text{pred}_{i_2}} \leq T_{\text{pred}_{i_3}}$
    $\text{BBUsForSleep} \leftarrow \{i_1, i_2\}$
else if TrafficLevel = M **then**
    Identify BBUs i_1, i_2, i_3 such that $T_{\text{pred}_{i_1}} \leq T_{\text{pred}_{i_2}} \leq T_{\text{pred}_{i_3}}$
    $\text{BBUsForSleep} \leftarrow \{i_1\}$
else
    $\text{BBUsForSleep} \leftarrow \emptyset$
end if
for i in BBUsForSleep **do**
    $\text{WakeUpThreshold}_i \leftarrow 0.3 \times C_i$
end for
Validation Check of Covering BBUs:
for i in BBUsForSleep **do**
    for j in BBUs $\setminus \{i\}$ **do**
        if $\text{Status}_j \neq$ sleep **and** $T_{pred_j} + T_{pred_i} < \text{WakeUpThreshold}_j$ **and** $\text{WakeUpThreshold}_j \geq 0.7 \times C_j$ **then**
            $\text{CoveringBBUs} \leftarrow \text{CoveringBBUs} \cup \{j\}$
        end if
    end for
end for
Entering Sleep Mode:
for i in BBUsForSleep **do**
    Transfer T_{pred_i} to one of CoveringBBUs
    $\text{Status}_i \leftarrow$ sleep
end for
Safe Switch Mechanism:
if $\Delta T > 0.1$ **then**
    All BBUs remain operational
end if
Wake-Up BBUs:
for j in BBUs $\setminus$ BBUsForSleep **do**
    if $T_{pred_j} + T_{pred_i} > \text{WakeUpThreshold}_j$ for any i in BBUsForSleep **then**
        $\text{Status}_i \leftarrow$ active
    end if
end for
for i in BBUsForSleep **do**
    if $T_{pred_i} > \text{WakeUpThreshold}_i$ **then**
        $\text{Status}_i \leftarrow$ active
    end if
end for=0

In Fig. 10, downlink traffic volume prediction (in MBs) using GRU for site B is given. Note that the predicted traffic volume is generally higher than actual traffic, which is a good sign since if the algorithm makes BBUs sleep just considering this GRU prediction, the number of BBUs will still be higher than needed, so clients will face a service quality degradation. On the other hand, if the actual downlink traffic is greater than the predicted downlink traffic at a time, customers may have poor service quality because our algorithm would make some BBUs sleep although they should have been active. Since we cannot guarantee that the predicted traffic will be higher than the actual traffic, we may need to adjust a scale factor for the predicted traffic to ensure that the predicted traffic is slightly higher than the actual traffic. Then, we can arrange hardware (i.e., BBUs), and software equipment accordingly. Therefore, all clients can have satisfied service quality. Also, the same methodology can be applied to LSTM prediction as well, i.e., to Fig. 10.

2) RRH CLUSTERING FOR C-RAN

According to user data rate requirements, RRH clustering could be utilized efficiently based on our network prediction. A one-to-one logical mapping exists between RRHs and BBUs in conventional RAN architectures. However, in C-RAN, we can establish a one-to-many relationship, where we assign one BBU to many RRHs. The RRHs with low traffic loads and user populations can be merged and managed by one BBU, whereas the RRHs with high traffic loads and user rates can be assigned to multiple BBUs. The decision of which RRHs will be attached to which BBU can be dynamically decided based on our downlink and uplink volume prediction. Since MAPE is only 8% on average for the downlink and 10.23% on average for the uplink, we can create load-aware RRH clusters more efficiently.

3) CENTRALIZED ALLOCATION OF NETWORK RESOURCES FOR TRANSPORT PART OF THE MOBILE NETWORK

All data volume that is sent from the cell site to the core passes through a transport network that consists of L2/L3 switches or routers. These switches/routers in 5G networks could be centrally controlled, to have an overview of entire network paths. Using data volume prediction with an accuracy of up to 93.72% for downlink and up to 92.23% for uplink will allow operators to dynamically adjust traffic in the transport network. Though, the predictive machine learning algorithms, and advanced modeling can be used to dynamically re-route messages in real-time via the lowest delay paths, paths with the lowest packet loss or lower throughput, based on actual users' needs. In 5G mobile networks, network slicing is a related concept. Each slice requires a different parameter set. For example, Ultra-Reliable Low Latency Communications (URLLC) traffic is highly sensitive to delay, whereas, for Massive Machine-Type Communications (mMTC), it is possible to consider packet loss as a critical parameter. With the help of QCI knowledge to identify the network slices in our dataset, one can predict more efficient routes using centralized management that has an overview of the entire network and their individual paths. So, it will be possible to dynamically re-route traffic through other paths according to the prediction, which decreases energy consumption and enhances user satisfaction.

VII. CONCLUSION

In this paper, a comprehensive real-world mobile data analysis is presented. We state our findings based on the traffic prediction for resource allocation in B5G networks. The characteristics and the features of a real world dataset are obtained from 4G+ and 5G systems to extract the real operator's data traffic patterns for the periods after COVID-19 struck our daily lives. Even though the developed traffic prediction models were trained with the 4G+ and 5G data, they can also be applied in B5G systems. Moreover, since the dataset was collected during the pandemic, it is expected to have a certain impact on network traffic data in the future, such as higher bandwidth usage in residential areas during certain time periods and higher usage of multimedia applications that create increased downlink traffic volume.

We have applied three different algorithms for network traffic volume prediction, namely FB's Prophet, Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU). The mean of value MAPE for LSTM is 8.52%, and 11.95% for GRU, whereas FB's Prophet is 28.28% showing that RNN algorithms, LSTM, and GRU, are better predictors than the statistical method FB's Prophet. Additionally, LSTM shows slightly better performance in the ratio of 3.9% in this dataset. This performance analysis shows how ML yields promising results in network data prediction.

Furthermore, we introduced the PESBiU algorithm, which combines traffic prediction and power consumption analysis to manage the power states (sleep or active) of BBUs in a network. The PESBiU algorithm prioritizes traffic volume to ensure that the network performance is maintained while optimizing energy usage. It successfully identifies low-traffic periods and transitions BBUs to sleep mode, resulting in significant power savings. When applied on one specific day in site B, it resulted in an energy saving of 332.35 Wh, which corresponds to a 15.12 % reduction for the entire day. This algorithm effectively balances energy efficiency and network performance, demonstrating its potential for practical deployment in future mobile communication networks.

Alongside the experiment results, different future directions and a network resources and QoS optimization techniques are proposed, which helps to increase efficiency and decrease power consumption in the next-generation wireless networks.

REFERENCES

[1] R. T. Rodoshi, T. Kim, and W. Choi, "Resource management in cloud radio access network: Conventional and new approaches," *Sensors*, vol. 20, no. 9, p. 2708, May 2020.

[2] S. Math, L. Zhang, S. Kim, and I. Ryoo, "An intelligent real-time traffic control based on mobile edge computing for individual private environment," *Secur. Commun. Netw.*, vol. 2020, pp. 1–11, Oct. 2020.
[3] *Green Mobile Network: Energy Saving Efforts by SK Telecom and NTT DOCOMO*. Accessed: Mar. 2024. [Online]. Available: https://www.docomo.ne.jp/english/corporate/technology/rd/docomo6g/
[4] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies," *IEEE Access*, vol. 9, pp. 67512–67547, 2021.
[5] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, 2018.
[6] *5G Energy Efficiencies: Green is New Black*. Accessed: Mar. 2024. [Online]. Available: https://data.gsmaintelligence.com/research/research/research-2020/5g-energy-efficiencies-green-is-the-new-black
[7] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "DeepTP: An end-to-end neural network for mobile cellular traffic prediction," *IEEE Netw.*, vol. 32, no. 6, pp. 108–115, Nov. 2018.
[8] Z. Di, T. Luo, C. Qiu, C. Zhang, Z. Liu, X. Wang, and J. Jiang, "In-network pooling: Contribution-aware allocation optimization for computing power network in B5G/6G era," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1190–1202, May 2023.
[9] J. Wu, Y. Zhang, M. Zukerman, and E. K. Yung, "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 803–826, 2nd Quart., 2015.
[10] F. Han, S. Zhao, L. Zhang, and J. Wu, "Survey of strategies for switching off base stations in heterogeneous networks for greener 5G systems," *IEEE Access*, vol. 4, pp. 4959–4973, 2016.
[11] A. Azari, F. Salehi, P. Papapetrou, and C. Cavdar, "Energy and resource efficiency by user traffic prediction and classification in cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 1082–1095, Jun. 2022.
[12] L. Chen, D. Yang, D. Zhang, C. Wang, J. Li, and T.-M.-T. Nguyen, "Deep mobile traffic forecast and complementary base station clustering for C-RAN optimization," *J. Netw. Comput. Appl.*, vol. 121, pp. 59–69, Nov. 2018.
[13] W. Zhao and S. Wang, "Traffic density-based RRH selection for power saving in C-RAN," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3157–3167, Dec. 2016.
[14] Y. Lee, K. Miyanabe, H. Nishiyama, N. Kato, and T. Yamada, "Threshold-based RRH switching scheme considering baseband unit aggregation for power saving in a cloud radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2676–2687, Sep. 2019.
[15] S. R. Aldaeabool and M. F. Abbod, "Reducing power consumption by dynamic BBUs-RRHs allocation in C-RAN," in *Proc. 25th Telecommun. Forum (TELFOR)*, 2017, pp. 1–4.
[16] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 2353–2358.
[17] S. Jaffry and S. F. Hasan, "Cellular traffic prediction using recurrent neural networks," in *Proc. IEEE 5th Int. Symp. Telecommun. Technol. (ISTT)*, Nov. 2020, pp. 94–98.
[18] I. Rahman, S. M. Razavi, O. Liberg, C. Hoymann, H. Wiemann, C. Tidestav, P. Schliwa-Bertling, P. Persson, and D. Gerstenberger, "5G evolution toward 5G advanced: An overview of 3GPP releases 17 and 18," *Ericsson Technol. Rev.*, vol. 2021, no. 14, pp. 2–12, Oct. 2021.
[19] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, "Intelligent network data analytics function in 5G cellular networks using machine learning," *J. Commun. Netw.*, vol. 22, no. 3, pp. 269–280, Jun. 2020.
[20] A. Mughees, M. Tahir, M. A. Sheikh, and A. Ahad, "Towards energy efficient 5G networks using machine learning: Taxonomy, research challenges, and future research directions," *IEEE Access*, vol. 8, pp. 187498–187522, 2020.
[21] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Netw.*, vol. 32, no. 6, pp. 42–49, Nov. 2018.
[22] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," in *Proc. SSST-8, 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, pp. 103–111.
[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[24] *Environmental Engineering (EE); Metrics and Measurement Method for Energy Efficiency of Wireless Access Network Equipment; Part 1: Power Consumption—Static Measurement Method*, Standard ETSI 202 706-1, 2022.
[25] T. Norp, "5G requirements and key performance indicators," *J. ICT Standardization*, vol. 6, no. 1, pp. 15–30, 2018.
[26] *Product Description for Ericsson 5216*. Accessed: Apr. 2024. [Online]. Available: https://www.tempestns.com/products/ericsson-baseband-5216-kdu137925-31/

ANETA KOLACKOVA received the master's degree in telecommunications and information technology from Brno University of Technology, in 2019, where she is currently pursuing the Ph.D. degree in 4G and 5G mobile technologies. During her studies, she also studied communication technologies with the Instituto Superior de Engenharia de Lisboa, in 2016 and 2017. In her research and development activities, she has been involved in projects dealing with mobile network virtualization, transport and backbone performance optimization of wireless systems, and cloud technologies. During her internship at a foreign operator, she gained hands-on experience with equipment from Ericsson and Cisco, where she is also further involved in building 5G networks.

SALIH SEVGICAN received the B.Sc. and M.Sc. degrees in computer engineering from Bogazici University, Turkiye, in June 2018 and January 2022, respectively. He was a Research Assistant during his exchange term with the University of Oulu, Finland. He is currently the Head of the Mobile Software Development Team and AI Team, Harpeon, Istanbul. His research interests include next-generation cellular networks, artificial intelligence, and cloud computing.

MUHAMMET FATIH ULU received the bachelor's degree in physics from Bilkent University, in 2020. He is currently pursuing the master's degree in computer engineering with Boğaziçi University. He is part of the Software Engineering Team, Mavinci, where he focuses on developing research and development applications powered by machine learning technologies.

JALE SADREDDINI received the Ph.D. degree in computer engineering from Karadeniz Technical University, in 2018. She currently holds the position of a 5G BB System Developer with Ericsson Canada Inc., focusing on carrier aggregation implementation in both downlink and uplink. Prior to her industry role, she was an Assistant Professor with Istanbul Arel University, also coordinating the Erasmus+ Program, from 2018 to 2020. In Summer 2019, she collaborated with Prof. Halim Yanikomeroglu with Carleton University. He has a history of international research, including a Visiting Researcher with Brno University of Technology, Czech Republic, in 2016, where she worked on a 3GPP long-term evolution testbed. Recognized for her teaching excellence, she was awarded "Best Teaching Assistant" from the Computer Engineering Department (2015–2016).

PAVEL MASEK (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering and Communication, Brno University of Technology (BUT), Czech Republic, in 2013 and 2017, respectively. He is currently a Researcher with the Department of Telecommunications, BUT. He is also co-supervising the WISLAB Research Group, where his current interests include various aspects in the area of heterogeneous communication networks. He has co-authored more than 120 research works on a variety of networking-related topics in internationally recognized venues, including those published in *IEEE Communications Magazine* and several technology products.

JIRI HOSEK (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from Brno University of Technology (BUT), in 2007 and 2011, respectively. He is currently an Associate Professor and the Deputy Vice-Head of Research and Development (R&D) and International Relations with the Department of Telecommunications, BUT. Since 2012, he has been coordinating the WISLAB Research Group, focusing on modern wireless technologies and applications. He has co-authored over 150 research works in networking, mobile communications, quality of service, quality of experience, and the IoT services.

JAN JERABEK was born in Bruntal, Czech Republic, in 1982. He received the B.Sc. and M.Sc. degrees and the Ph.D. degree in electrical engineering from Brno University of Technology, Czech Republic, in 2005, 2007, and 2011, respectively. He is currently an Associate Professor with the Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology. His research interests include analogue signal processing including circuit design, analyses, and measurements.

TUNA TUGCU received the B.S. degree in computer engineering from Boğaziçi University, the M.S. degree in computer and information science from New Jersey Institute of Technology, and the Ph.D. degree in computer engineering from Boğaziçi University. He was a Postdoctoral Fellow and a Visiting Professor with Georgia Institute of Technology. He is currently a Professor with the Computer Networks Research Laboratory (NETLAB) and Nanonetworking Research Group (NRG), Boğaziçi University. His research interests include nanonetworking, molecular communications, and wireless networks. He has served in the NATO Science and Technology Organization IST104-RTG050 Group (Cognitive Radio in NATO II) and IST-ET-074 Group (Network Aspects of Cognitive Radio). He also serves for the IEEE P1906.1 Nanoscale and Molecular Communications Working Group, contributing to the IEEE 1906.1-2015 Recommended Practice for Nanoscale and Molecular Communication Framework and IEEE Standard Data Model for Nanoscale Communication System standard specifications. He is currently an Associate Editor of IEEE TRANSACTIONS ON MOLECULAR BIOLOGICAL AND MULTI-SCALE COMMUNICATIONS. He is a devoted and active member of the #WeDoNotAcceptWeDoNotGiveUp movement for free, autonomous, and democratic academia.