Machine Learning Models for Virtual Base Station Power Consumption Estimation

Elen Gomes, Lucas Rodrigues, Diego Bezerra, Djamel F. H. Sadok and Glauco Gonçalves

Abstract— **The growing demand for energy efficiency in mobile networks has driven the adoption of virtualized Base Stations running on general-purpose processors. This paper compares machine learning models for estimating power consumption using data from four processor architectures. Extreme Gradient Boosted Trees Regressions delivered the most accurate and robust predictions. Neural Networks exhibited unstable performance, particularly on specific platforms, whereas Linear Regressions demonstrated lower reliability in low-power scenarios. Results highlight the critical importance of aligning the processor architecture with the chosen model to ensure practical power estimation and energy optimization.**

Keywords— **Energy efficient 5G, vRAN, Power Modelling.**

I. Introduction

Along with the evolution of different generations of mobile networks, their underlying systems have increasingly demanded more energy due to the rising number of users, the emergence of more resource-demanding services, and growing data traffic volumes [1]. Whereas energy efficiency (EE) is a core design principle of 5G New Radio (NR) [2], the widespread adoption of mobile networks as the backbone to support smart cities, Industry 4.0, and autonomous vehicles imposes new challenges for ensuring network EE [1].

In this context, it is noteworthy that, on average, about 70% of a telecommunications operator's energy consumption comes from the Radio Access Network (RAN). Within the RAN, the Base Station (BS) is the most energy-consuming component, accounting for approximately half of the total consumption [3]. Therefore, actions on energy consumption optimization and reduction of carbon emissions [4] can help 5G networks to promote the United Nations Sustainable Development Goals (UN SDGs) [5] and overall environmental preservation.

Among the available technological approaches for improving EE at the BS level, Network Function Virtualization (NFV) stands out. Under this approach, BS functions are implemented in software as virtual Base Stations (vBSs), running on general-purpose Central Processing Units (CPU) or shared computing pools [6]. It is known that vBSs can typically consume significantly more power than those running on dedicated hardware [7]. However, a vBS can be fine-tuned and rescheduled in real-time by software to meet current service and user demands. This flexibility allows for freeing up CPU time for other network functions, contributing to improved overall efficiency.

Elen Gomes, Lucas Rodrigues, and Glauco Gonçalves are part of LASSE, UFPA, Belém-PA. Diego Bezerra and Djamel F. H. Sadok are part of GPRT, UFPE, Recife-PE. E-mails: {elen.cristina, lucas.lima.rodrigues} @itec.ufpa.br, glaucogoncalves@ufpa.br, {diego.bezerra, jamel} @gprt.ufpe.br.

Implementing energy-saving strategies at the software level impacts operational costs and the network's environmental performance [8], underscoring the importance of accurate energy prediction and optimization tools.

Estimating how much power a vBS consumes at a given time is imperative in energy consumption. When working with a BS, such a value can be measured directly from the device. However, power must be estimated for a vBS network function since the vBS shares a CPU with other network functions. Regarding modeling the energy consumption of BS, various approaches have been used across the existing literature. Linear models have been commonly used for estimating power consumption, as discussed by [9]. However, a strong tendency exists toward adopting Machine Learning (ML) models [6].

In this direction, this work conducts a comparative analysis of the performance of Neural Networks (NN) and Extreme Gradient Boosted Trees Regression (XGB) models against the Linear Regression (LR) model, used as a baseline in prior studies [6], [9], [10]. These experiments aim to identify the most effective approach for estimating power consumption, providing insights applicable to energy usage modeling in vBS environments. Model evaluation is conducted in a controlled experimental setup using the dataset provided by [11].

From a practical standpoint, models like these can be integrated into network management systems, allowing dynamic adjustments in the configuration of vBSs, such as modifying sleep duration based on energy consumption. This approach can significantly contribute to the network's EE while reducing operational costs and environmental impact.

This paper is organized as follows: Section II discusses the related work; Section III describes how the experiments were designed; Section IV presents and discusses the results; and finally, Section V provides the findings and future research directions.

II. Related Work

One of the earlier approaches to power consumption modeling in BSs is presented in [12], where Quadratic and LR models estimate the energy consumption of 2G and 3G BSs as a function of voice and data traffic. The quadratic regression model shows better performance. The authors also note that the BS power to traffic pattern relationship changes for each BS and period, which indicates that a one-size-fits-all model would fail to capture all the variability of the RAN, and, in turn, it opens space for developing BS-specific ML models.

Building upon this foundation, [10] employs a combination of XGB, CatBoost, and NN models through a weighted average approach to enhance the accuracy of energy consumption predictions. The results demonstrate the effectiveness of this approach, achieving a Mean Absolute Percentage Error (MAPE) of 3.5620%, Root Mean Squared Error (RMSE) of 1.1524, and Mean Absolute Error (MAE) of 1.0245. The authors further highlight the significant impact of BS configuration on energy consumption, emphasizing factors such as Remote Radio Units (RU) type, number of antennas, and other operational parameters.

Further extending the scope of ML-based energy modeling, [9] introduces an Energy Analytics Dashboard supporting Energy Scoring and a RU EE Analysis. This work used ML techniques to identify inefficient RUs regarding energy consumption. The MAPE was measured for XGB (10.83%), Random Forest Regressor (10.95%), Decision Tree Regressor (14.22%), and Huber Regressor (29.41%) algorithms, with XGB achieving the best performance.

In a vBS-focused context, authors in [6] present vBS power consumption regressors using NNs trained with data that includes features such as airtime, Signal-to-Noise Ratio (SNR), and Modulation and Coding Scheme (MCS). The NN models were compared against a custom regression model with domain knowledge from [13]. To train and test the models, authors leveraged data from the dataset in [11] where each model was labeled according to the vBS's underlying CPUs, namely, NUC1, NUC2, SERVER1, and SERVER2. The results show that, in general, the NN model performance is similar to that of the regression model, RMSE is below 0.20 for NUC1, between 0.20 and 0.40 for NUC2 and SERVER1, and between 1.20 and 1.40 for SERVER2 for both prediction techniques, but without requiring deep domain knowledge or making problematic assumptions on the way variables are related.

Despite the growing interest in energy-efficient modeling, most prior works focus on physical BSs or RUs. In contrast, vBSs, a trend in the evolution of RAN, remain underexplored. Existing studies show that ML models must be tailored to the hardware and operational profile of each BS, and this is particularly true for vBSs, where energy consumption is closely tied to the characteristics of the compute resources. While ensemble tree-based models (e.g., XGB, Random Forest) have shown superior performance in energy prediction tasks, none of the cited studies has investigated or compared such models specifically in the context of vBSs.

III. Material and Methods

Based on prior literature, this work compares three distinct models: LR, NN, and XGB. LR is a baseline due to its simplicity, low computational cost, and frequent use in earlier studies [9]. NN was selected for its ability to model nonlinear patterns without domain-specific assumptions [6], while XGB, a tree-based ensemble, was included for its proven accuracy in energy prediction [10]. Together, these models represent traditional, neural, and tree-based approaches, supporting a broad evaluation of predictive performance across CPUs.

To evaluate the proposed models, this paper adopts a methodological framework illustrated in Figure 1. All Python code used in the experiments is publicly available in a GitHub repository to support transparency and reproducibility. This repository includes scripts for generating feature density plots, performing model selection with log files in Text File (TXT) format, and conducting train-test experiments with scatter plots and output in Comma-Separated Values (CSV) files[1].

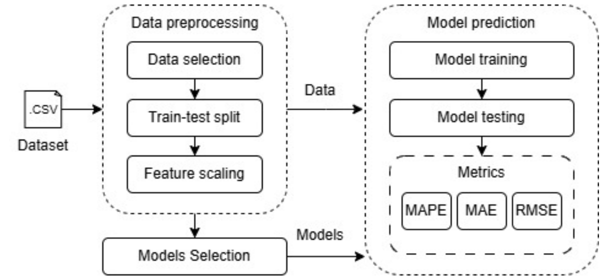

Fig. 1. Evaluation scheme of power consumption prediction models.

The experiments are based on the dataset made available by [11], specifically the `dataset_ul.csv` file, which contains Key Performance Indicators (KPIs) from the uplink channel and vBS power consumption metrics. The experimental setup includes a vBS, user equipment (UE), and a digital power meter. Both the UE and the vBS are emulated using an Ettus Universal Software Radio Peripheral (USRP) B210. The Baseband Unit (BBU) runs on two Intel Next Unit of Computing (NUCs) and two servers. The Remote Radio Head (RRH) is powered via USB from the BBU. The UE utilizes a general-purpose computer connected to the vBS via SMA cables with attenuators. All devices run Ubuntu 18.04 and a custom version of Software Radio Systems LTE (`srsLTE`) 19.12, which enables dynamic MCS and airtime control, as well as the retrieval of real-time performance metrics. Power consumption is measured by software using Intel's Running Average Power Limit (RAPL) interface with the `Turbostat` utility, capturing CPU power consumption only. Several experiments were conducted, with experimental factors including bandwidth, transmission mode, uplink load, UE transmission gain, MCS, and airtime. Each configuration runs for one minute, during which the average and variance of power usage are collected. The uplink traffic load is generated using the Multi-Generator Network Test Tool (`MGEN`). Further implementation details and additional experimental results can be found in the original publications by [11], [13].

Hyperparameters were tuned using Random Search, except for the LR, which used a simple Grid Search due to having only two configuration options (with/without intercept). The search spaces comprised 11,741,760 combinations for XGB and 1,818,720 for NN. A 1% sample of each was used to identify optimal configurations, selected based on the lowest MAPE score. For reproducibility, all models were instantiated with fixed seeds. Table I summarizes the hyperparameters and their tested values, which were chosen in an equiprobable way.

Following the methodology presented by [6], airtime, SNR and MCS variables were selected as features, with

[1] Available at: `https://github.com/lasseufpa/ml-for-vbs-power.git`. Accessed on May 19, 2025.

TABLE I
HYPERPARAMETERS AND VALUE RANGES USED IN MODEL TUNING.

Model	Hyperparameter	Value Range
XGB	n_estimators	randint(50, 201)
	learning_rate	[0.01, 0.05, 0.1, 0.2, 0.3]
	max_depth	randint(3, 11)
	min_child_weight	randint(1, 7)
	gamma	[0, 0.1, 0.3, 1]
	subsample	[0.7, 0.8, 1.0]
	colsample_bytree	[0.7, 0.8, 1.0]
	reg_alpha	[0, 0.01, 0.1]
	reg_lambda	[0.1, 1, 10]
NN	hidden_layers	1-3 layers (5-100 units with step 5)
	activation	["relu", "tanh", "logistic"]
	solver	["adam", "sgd"]
	learning_rate_init	[0.0005, 0.001, 0.01]
	alpha	[0.0001, 0.001, 0.01]
	beta_1	[0.9, 0.95]
	beta_2	[0.99, 0.999]

the respective column names `airtime`, `mean_snr`, and `mean_used_mcs`, while the power consumption measured by software was used as the target variable (`rapl_power` column). To ensure the consistency and quality of the dataset, only records that met the following selection criteria were considered:

1) The MCS variable can take on different values up to a maximum limit defined by the platform (column `fixed_mcs_flag=0`);
2) No experimental failures (column `failed_experiment=0`);
3) A channel bandwidth of 10 MHz (column `BW=50`).

The evaluation considered the four CPUs in the dataset, using aliases from [6], as well as a complete data split of 80% for training and 20% for testing, which results in the sample distribution shown in Table II.

TABLE II
SAMPLE DISTRIBUTION PER CPU AND ALIASES.

Alias	CPU	TDP[W]	Train	Test
NUC1	i7-8559U@2.70GHz	28	479	119
NUC2	i7-8650U@1.90GHz	15	128	32
Server1	i7-6700@3.40GHz	65	86	21
Server2	i7-9700@3.00GHz	65	86	21

Table II presents the Thermal Design Power (TDP) of the processors used in the Server and NUC environments. TDP represents the average power, in watts, that a processor dissipates when operating at its Base Frequency with all cores active under an Intel-defined, high-complexity workload. As expected, Server processors exhibit higher TDP values, reflecting their greater computational and thermal requirements[2].

[2]Available at: `https://www.intel.com/content/www/us/en/products/compare.html?productIds=191792,88196,124968,137979`. Accessed on May 19, 2025.

Additionally, the models were generated separately for each CPU since each one exhibits a different power consumption range for the same experiment. At the same time, the input characteristics (airtime, SNR, and MCS) do not vary significantly, as illustrated in Figure 2, which shows the probability density functions of both the input features and the target variable (Power Consumption).

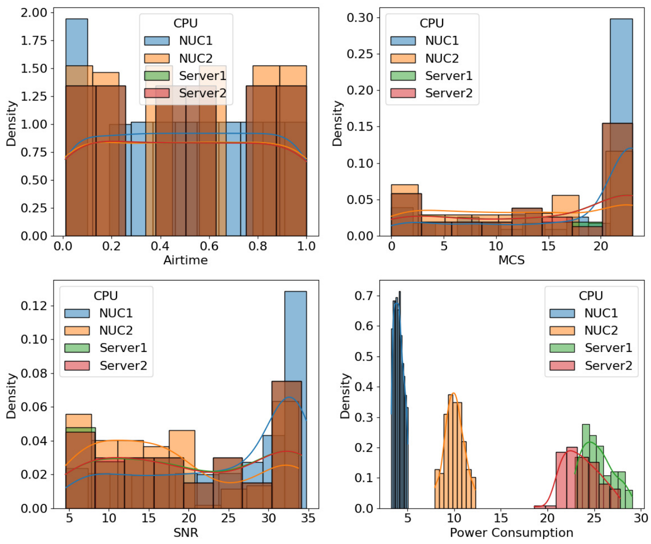

Fig. 2. Statistical distribution of the features and the target for each CPU.

The features were standardized using the `MinMaxScaler`, which adjusts the values to the range $[0, 1]$, as shown in Equation 1. The standardized values can then be rescaled to a new range $[a, b]$, using Equation 2; however, for this study, the original $[0, 1]$ scale was used.

$$X_{std} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

$$X_{scaled} = X_{std} \cdot (b - a) + a \tag{2}$$

Finally, the models were trained using the training data and evaluated based on the estimates generated on the test set. The comparison is based on the metrics MAE, RMSE, and MAPE, in line with the approach adopted in [10]. The MAE, given in Eq. (3), measures the average of the absolute differences between the actual values y and the predicted values $\hat{y}$, where n is the number of samples. It is simple to interpret and less sensitive to outliers [14]. The MAPE, in Eq. (4), expresses the error as a percentage relative to the actual value, facilitating comparisons across different scales, although it is sensitive to minimal values ($y_i \approx 0$). The RMSE, defined in Eq. (5), computes the square root of the mean squared error, heavily penalizing significant errors and therefore being more sensitive to outliers than MAE.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} = \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

IV. Results and Discussion

The estimative performance of all evaluated models across different CPUs is summarized in Table III. Among all evaluated approaches, the XGB algorithm demonstrated the most consistent and accurate performance across all processors, achieving the best MAE and MAPE values in every scenario. However, it did not always produce the lowest RMSE, which is known to be more sensitive to outliers.

TABLE III

Metric results for each model.

CPU	Model	MAE [W]	RMSE [W]	MAPE [%]
NUC1	LR	0.079	0.105	2.022
	XGB	**0.024**	**0.031**	**0.611**
	NN	**0.024**	**0.031**	0.619
NUC2	LR	0.286	0.358	2.895
	XGB	**0.246**	0.371	**2.537**
	NN	0.269	**0.334**	2.800
Server1	LR	0.332	0.425	1.306
	XGB	**0.256**	**0.370**	**1.018**
	NN	0.281	0.378	1.080
Server2	LR	0.845	**1.401**	3.762
	XGB	**0.665**	1.537	**2.970**
	NN	1.469	1.713	6.431

For the NUC1 device, the XGB and NN models yielded nearly identical results, with both models sharing the lowest MAE and RMSE values. The NN exhibited a marginally higher MAPE, differing only at the third decimal place, indicating that both models are practically interchangeable.

On NUC2, XGB again achieved the best MAE and MAPE, while NN slightly outperformed XGB regarding RMSE, likely due to better handling of outliers in that specific scenario.

On Server1, XGB produced the lowest errors across all three metrics, reaffirming its robustness. For Server2, a similar trend to NUC2 was observed: XGB achieved the best MAE and MAPE, whereas the LR model obtained the lowest RMSE.

When evaluating average performance across all processors, LR achieved a mean MAE of 0.385, RMSE of 0.572, and MAPE of 2.496%. XGB achieved a mean MAE of 0.298, RMSE of 0.577, and MAPE of 1.784%, while NN yielded a mean MAE of 0.511, RMSE of 0.614, and MAPE of 2.732%. Therefore, XGB had the lowest average MAE and MAPE, whereas LR achieved the lowest average RMSE.

It is important to emphasize that, in absolute terms, minor prediction errors (measured in watts) are more critical in low-power devices. While server-grade processors (Server1 and Server2) have a TDP of 65 W, the NUC1 and NUC2 devices operate at 28 W and 15 W, respectively. That means an error of 1.469 W represents only 2.26% of the TDP for servers but corresponds to 5.25% of NUC1's TDP and 9.79% of NUC2's TDP. Consequently, models must be precise when applied to resource-constrained devices such as NUCs.

The RMSE values obtained by the best-performing models for each CPU were closely aligned with those reported in [6]. Although the author did not disclose the exact performance metrics, the models were evaluated using the same dataset, and the reported RMSE ranges are consistent with the results of this study. This alignment reinforces the validity of the adopted approaches in nearly all cases, except for the LR model on Server1 and the models evaluated on Server2.

Such influence of outliers can be visually confirmed in the scatter plots shown in Figures 3, 4, 5, and 6. In these plots, the comparison between the model predictions and the identity line (representing ideal predictions) reveals that the points are more tightly clustered around the line in NUC1, suggesting lower variability and greater ease in estimating power consumption. For NUC1, it is also evident that the LR model generated predictions that deviated significantly from the identity line compared to the other models, a behavior not observed with the same intensity on the different CPUs.

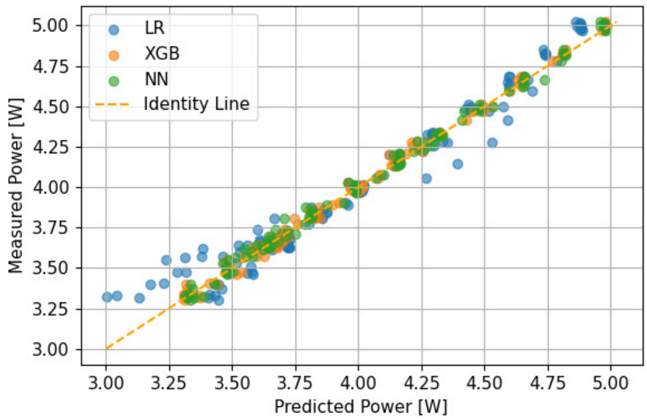

Fig. 3. Scatter plot comparing model predictions on NUC1.

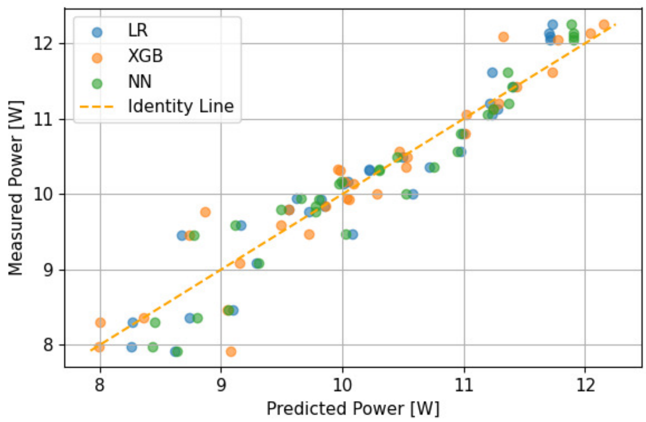

Fig. 4. Scatter plot comparing model predictions on NUC2.

In contrast, the Server2 CPU exhibited a wider dispersion in prediction errors, with several instances of substantial underestimation and overestimation. In extreme cases, power consumption values around $21W$ were predicted to be as high as $27.5W$, highlighting the model's difficulty in achieving reliable generalization in this context.

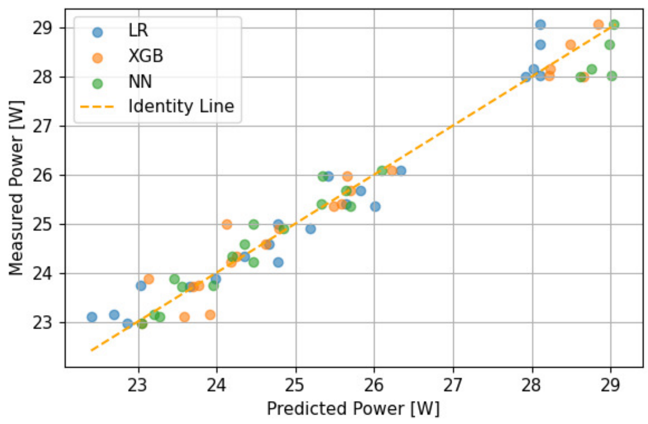

Fig. 5. Scatter plot comparing model predictions on Server1.

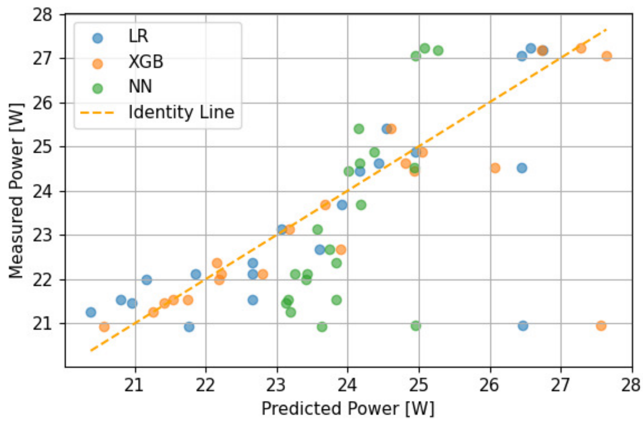

Fig. 6. Scatter plot comparing model predictions on Server2.

A notable behavior observed in the neural network model on Server2 was its tendency to restrict predictions to a narrow range between $23W$ and $26W$ rather than leveraging the full spectrum of observed values. This limitation likely contributed to this model's significantly higher error metrics than its performance on other processors.

V. Conclusion

This work evaluated LR, NN, and XGB models for estimating power consumption in vBS scenarios across different CPU architectures. XGB consistently outperformed the others in terms of MAE and MAPE, while LR achieved the lowest RMSE; however, it was less reliable in low-power scenarios due to its sensitivity to outliers. NN exhibited unstable behavior, particularly on Server2, with significantly higher prediction errors. Despite the limited diversity of CPUs and reduced number of samples, especially for server platforms, the findings align with prior research and indicate that power estimation must consider both the hardware architecture and the chosen model. XGB proved to be the most accurate and robust approach for heterogeneous vBS environments.

As future work, the adoption of online learning techniques is proposed to enable continuous adaptation to workload and hardware variations. Additionally, the developed models are intended for integration into an xApp to support real-time power estimation and dynamic energy optimization in vBS environments.

Aknowledgements

This work was supported by National Council for Scientific and Technological Development (CNPq) (process: 408326/2023-9), the Rede Nacional de Ensino e Pesquisa (RNP), the Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD), and the OpenRAN@Brasil program (process MCTI No: A01245.014203/2021-14).

References

[1] J. Lorincz, Z. Klarin, and D. Begusic, "Analyses of User Density Impact on Energy-efficiency Metrics in 5G Networks," in *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2020, pp. 1–6. DOI: 10.23919/SoftCOM50211.2020.9238172.

[2] 3GPP, "Study on scenarios and requirements for next generation access technologies," *3rd Generation Partnership Project (3GPP), Technical report (TR) 36.331*, 2017.

[3] L. Kundu, X. Lin, and R. Gadiyar, "Toward Energy Efficient RAN: From Industry Standards to Trending Practice," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 36–43, 2025. DOI: 10.1109/MWC.010.2400061.

[4] Z. A. Rahim and M. Saqib Iqbal, "Analysing 5G Patented Green Network Technologies Using TRIZ Patent Literature Review," in *2024 IEEE 6th Symposium on Computers & Informatics (ISCI)*, 2024, pp. 293–298. DOI: 10.1109/ISCI62787.2024.10668036.

[5] U. G. Assembly, "Transforming our world: the 2030 Agenda for Sustainable Development," 2015.

[6] M. Dzaferagic, J. A. Ayala-Romero, and M. Ruffini, "ML Approach for Power Consumption Prediction in Virtualized Base Stations," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 986–991. DOI: 10.1109/GCWkshps56602.2022.10008643.

[7] G. N. Katsaros, R. Tafazolli, and K. Nikitopoulos, "On the Power Consumption of Massive-MIMO, 5G New Radio with Software-Based PHY Processing," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 765–770. DOI: 10.1109/GCWkshps56602.2022.10008564.

[8] Y.-N. R. Li, M. Chen, J. Xu, L. Tian, and K. Huang, "Power Saving Techniques for 5G and Beyond," *IEEE Access*, vol. 8, pp. 108 675–108 690, 2020. DOI: 10.1109/ACCESS.2020.3001180.

[9] N. Pardhasaradhi, J. Bose, A. Vikram, A. Verma, and M. Jain, "Identification of Inefficient Radios for Efficient Energy Consumption in a Mobile Network," in *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2024, pp. 608–612. DOI: 10.1109/COMSNETS59351.2024.10426851.

[10] M. Ilyasse, A. Nafidi, R. Gutiérrez-Sánchez, and S. Aourik, "A machine learning approach for energy consumption in 5G networks," *ITU Journal on Future and Evolving Technologies*, vol. 5, pp. 447–457, Dec. 2024. DOI: 10.52953/IJXZ5881.

[11] J. X. Salvat Lozano, J. A. Ayala-Romero, L. Zanzi, A. Garcia-Saavedra, and X. Costa-Perez. "O-RAN experimental evaluation datasets." (2022).

[12] J. Nasreddine, K. Fakih, B. Haidar, D. Serhal, and S. Haidar, "Estimating Base Station Power Consumption Using Regression," in *2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, 2019, pp. 1–4. DOI: 10.1109/BIOSMART.2019.8734253.

[13] J. A. Ayala-Romero, I. Khalid, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Experimental Evaluation of Power Consumption in Virtualized Base Stations," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500323.

[14] J. Tohka and M. van Gils, "Evaluation of machine learning algorithms for health and wellness applications: A tutorial," *Computers in Biology and Medicine*, vol. 132, p. 104 324, 2021, ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2021.104324.