Machine Learning-Based MIMO Enabling Techniques for Energy Optimization in Cellular Networks

Mariam Aboelwafa[⋆], Mohamed Zaki[‡], Ayman Gaber[‡], Karim Seddik[⋆], Yasser Gadallah[⋆] and Ayman Elezabi[⋆]
[⋆]*The American University in Cairo,* [‡]*Vodafone Egypt*

***Abstract*—In this paper, we consider the problem of energy optimization in mobile networks by enabling the MIMO feature only when necessary. Enabling MIMO features at the base station increases energy consumption unnecessarily under many operating conditions. In this study, we employ machine learning-based approaches to decide on whether a SISO scheme can achieve the required Quality of Experience (QoE). If SISO can satisfy the target QoE, the base-station can decide to switch the MIMO feature off which can result in considerable energy savings. We consider two different machine learning approaches, namely, multi-layer perceptron (MLP) and recurrent neural networks (RNNs), to learn the SISO features from realistic mobile network data. The trained models are tested against the data obtained from MIMO cells in which the MIMO feature is disabled. Our results show the effectiveness of our proposed approach which presents a real-time, automated approach for MIMO enabling decisions.**

***Index Terms*—Energy optimization, mobile networks, SISO, MIMO, Machine Learning**

I. Introduction

Energy optimization has become a very important topic in dealing with practical communications and networking problems [1], [2]. This is because of the raised awareness all around the globe of the importance of having energy-conserving, green, and sustainable communication systems. The mobile communications technology, back in 2007, had a share of 2% in global carbon emissions and this is expected to double by 2020 [3]. This has drawn the researchers' attention to work on reducing the energy consumption of mobile networks. The component with the greatest share of energy consumption in mobile networks is the Base-station (or eNodeB in LTE standards) with a share greater than 50% [4]. Therefore, this work focuses on reducing energy consumption at the eNodeB in mobile networks.

In 4G networks, one of the most energy consuming features is the use of Multiple Input Multiple Output (MIMO) scheme. Unlike the Single Input Single Output (SISO) scheme, the main idea behind MIMO is to use multiple antennas at the transmitter and/or the receiver simultaneously. This allows for higher throughput and guarantees faster downloads and higher spectral efficiency [5].

Despite the fact that MIMO has many advantages, it consumes a large amount of energy that might not be always necessary. If a certain level of Quality of Experience (QoE) is achieved by SISO, a lot of energy can be saved by turning off the MIMO scheme at the eNodeB. Currently, mobile operators do this manually based on some user-defined schedules. An automated method to turn on/off the MIMO capability, based on the network performance, is certainly needed to reduce the amount of used energy.

To clarify how controlling MIMO can be effective to save energy, Vodafone Egypt provided information about the average energy consumption before and after turning off MIMO. For the MIMO sites, from our test dataset that will be presented later, turning off MIMO can cause energy saving that ranges from 2.5% to 8.6% of the total site energy (an average of 5.5% energy saving) resulting from only controlling the radio unit of the base-station. This variation of the energy savings is because of the fact that sites lie in different locations and are subject to different conditions, therefore, the total consumed energy is affected by other conditions other than the radio unit.

The main objective of this study is to use Machine Learning, specifically Neural Networks (NN), to learn some features of the network and decide whether SISO is sufficient to achieve a satisfactory level of QoE. Based on this, the mobile operator can decide whether to enable the MIMO capability.

NNs have been gaining a lot of interest recently in the wireless communications literature [6], [7]. NNs have the ability to learn the features of a certain system even if it has no model to represent it. By subjecting the NN to the data drawn from the system, it will be able to extract the common features and learn the performance of the system.

The main contribution of this work is to use two types of NNs: 1) Fully Connected NN (also called Multi-Layer Perceptron (MLP)) to learn the features of the SISO scheme 2) Recurrent NN to track any trends in the data history. Both machines are trained using historical data drawn from realistic SISO cells[1]. This data includes a number of network features recorded around the clock. When the training phase is complete, the machine is subjected to some cell features of a MIMO site and it emulates the performance of SISO scheme to decide whether SISO is able to achieve a satisfactory QoE. To make this decision, the average DownLink (DL) user throughput is predicted and monitored as a measure of QoE.

[1]All the data used in this paper are provided by one of the mobile operators in Egypt.

If the machine decides that SISO is enough, then MIMO can be turned off to save energy. Otherwise, the MIMO is kept on to achieve an acceptable QoE performance.

The remainder of the paper is organized as follows. A brief literature survey is presented in Section II. A background is presented in Section III. The proposed approach is explained in Section IV. Performance evaluation is presented in Section V. Finally, the study is concluded in Section VI.

II. Literature Review

Reducing the carbon footprint of mobile networks has been of interest in the literature recently. For example, [8] discusses the possibility of powering mobile networks with green energy. It presents an overview of the design and challenges of green energy enabled mobile networks.

The authors in [9], [10] and [11] discuss possible techniques to reduce the power consumption in base stations (BSs), since it is the entity with the highest power consumption in the whole mobile network [4]. They focus on optimizing air conditioning power consumption and minimizing feeder losses.

Exploitation of machine Learning (ML) techniques has been of interest in many research works in cellular networks. In [6], the authors investigate the possible use cases of machine learning in future cellular networks. They review the basic concepts of machine learning and propose their use in 5G networks, including cognitive radios, massive MIMOs, femto/small cells, heterogeneous networks, smart grid, energy harvesting and device-to-device communications. On the other hand, in [12], the authors are interested in cache content optimization in small base stations based on learning algorithms.

Using ML for energy saving in base stations has been investigated in the literature as well. In [13], the authors present a Reinforcement Learning (RL) approach for resource allocation in wireless networks. The presented algorithm learns the utility of performing various tasks over time and uses the application constraints for task management by optimizing energy usage and network lifetime. From another perspective, the authors in [14] adopt machine learning in energy harvesting. They propose a strategy learning algorithm that exploits the expected energy and adapts spectrum selection strategies to maximize the network's performance. The learning algorithm addresses how multiple users discover available channels and harvest energy over the network.

The novelty of our proposed approach lies in focusing on the energy optimization of the *Radio* part of the BS and not air conditioning or feeder losses. It targets the power amplifiers which are responsible for 65% of the total BS energy consumption [15]. Additionally, and to the best of our knowledge, this is the first work to address machine learning as means for the enabling decision of the MIMO scheme(s) on the basis of energy saving purposes. The current practice in mobile networks is to have some user-defined schedules or some other "experience-based" manual approach. These practices suffer from the fact that they are not real-time and that they are human-driven rather than data-driven. Our approach allows for real-time data-driven seamless operation (i.e. it learns features and their relationships, which is not the case in the human-driven approaches).

III. Technical Background

The Machine Learning (ML) mechanism utilized in this work is Neural Networks (NN). In particular, the Multi-Layer Perceptron (MLP) and the Recurrent NNs (RNNs) are adopted.

A. Multi-Layer Perceptron (MLP)

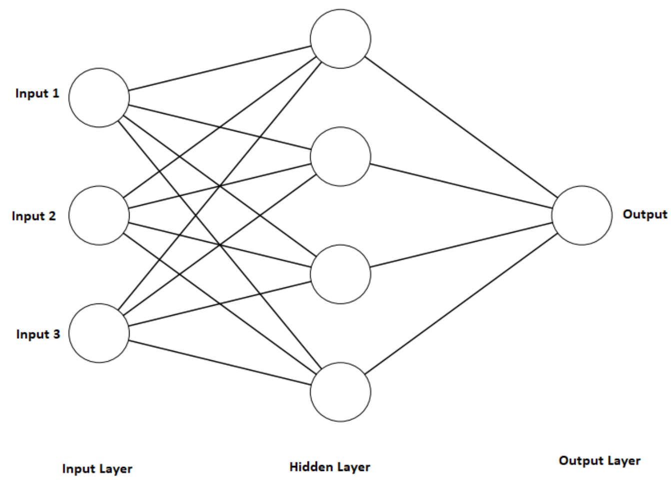

Fig. 1: An example of a simple multi-layer perceptron

MLP is a type of feed-forward artificial NN [16]. As shown in Fig. 1, an MLP consists of at least three layers of nodes: an input layer, a hidden layer(s) and an output layer. Each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training [17].

Generally, the output of the neural network can be represented as:

$$\mathbf{y} = h_\theta(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ is the input vector, h_θ is the full forward propagation function and θ is the set of weights and biases to be learned by the network during the training phase. The training phase is done by setting the output to be the target values provided by the training data. Therefore, for a certain h_θ, the only unknown in the equation is θ.

The steps of the learning process can be summarized as follows:

1) Initialize weights and biases (usually random).
2) Use the feed-forward direction (from input to output) to calculate the output.
3) Calculate the error function (a measure of the difference between the actual output and the target output). This error is a function of all the contributing errors from all connected neurons.
4) Use the error function, in the backward direction, to update the weights and biases using the equation:

$$\Delta\theta_i = -\alpha * \frac{\delta E}{\delta \theta_i}, \tag{2}$$

where $\Delta\theta_i$ is the update of the i^{th} weight/bias, α is the learning rate and E is the cumulative error function from all the previous contributing layers.

In this work, the error function that we use is the Mean Absolute Error (MAE) as will be justified later after presenting our system model and problem description.

B. Recurrent Neural Network (RNN)

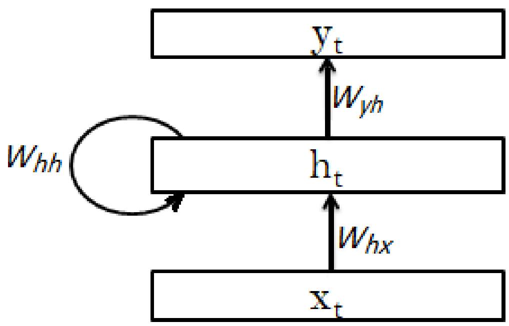

Fig. 2: An example of a simple Recurrent Neural Network

Recurrent Neural Networks (RNNs), as shown in Fig. 2, are distinguished from feed-forward networks by a feedback connection to their past decisions, taking their own outputs moment after moment as inputs. In other words, RNNs have memory. Adding memory to neural networks allows to track the information in the sequence itself, e.g. time correlations, trends, etc.

Generally, the output of an RNN can be given as:

$$y_t = f(W_{yh} \cdot h_t), \tag{3}$$

where y_t is the output at time t, f is the activation function of the output layer, W_{yh} is the set of weights and biases between the hidden layer and the output layer and h_t is the hidden layer output. This part is not different from the feed-forward NN. However the output of the hidden layer can be given as:

$$h_t = g(W_{hx} \cdot x_t + W_{hh} \cdot h_{t-1}), \tag{4}$$

where h_t is the output of the hidden layer at time t, g is the activation function of the hidden layer, W_{hx} is the set of weights and biases between the input layer and the hidden layer, x_t is the input at time t, W_{hh} is the set of weights in the feedback loop of the hidden layer and h_{t-1} is the output of the hidden layer at time $t-1$.

The training phase is done in a similar way to the feed-forward networks. However, the calculation of the cumulative error and its gradient is done in a different way. Details for the interested reader can be found in [18].

Next, the details of how an MLP and an RNN can be used to save energy in cellular networks are explained.

IV. Problem Description and Proposed Approach

As mentioned earlier, the main objective of this work is to exploit Machine Learning (ML) approaches to reduce energy consumption in the radio part of the base stations. This is achieved by turning off the MIMO feature *if and only if* SISO is able to achieve a minimum satisfactory Quality of Experience (QoE), which is measured in our work by a minimum average DownLink (DL) user throughput[2]. Our models will follow regression models to estimate the expected user DL average throughput for a SISO scheme based on some network parameters as will be explained later. It is worth mentioning that our algorithm does not provide any service guarantees for any specific UE. MIMO feature is turned on/off based on the expected average throughput per user.

A. Data

The data used for training, validation and testing are real data shared by Vodafone Egypt. It includes Key Performance Indicators (KPIs) from 145 **SISO** sites calculated at the cell level every hour. The KPIs are:

- DL Physical Resource Block (PRB) utilization.
- Average Channel Quality Indicator (CQI) per cell.
- DL traffic volume (in GBytes).
- Average number of User Equipment (UE).
- Maximum number of UEs.
- User DL average throughput.

The user DL average throughput is taken to be the output of the machine as it is the adopted measure of QoE (which is the current practice in mobile networks), while the remaining KPIs are taken as inputs. MLP takes only the current measurements to predict the DL user average throughput, if SISO is used. On the other hand, RNN tracks the historical trend of the data to predict the user DL average throughput.

The total number of data points in our data-set is 382,456. During our experiments, 70% of the available data was used for training, 20% was used for validation and 10% was used for testing.

B. ML Models Architectures

We now present the models that we use in the two NN methods that we use in this study as follows.

1) Multi-Layer Perceptron Architecture: The used MLP consists of:

- 1 input layer containing 12 neurons (ReLu activation function).
- 1 hidden layer containing 8 neurons (ReLu activation function).
- 1 output layer containing 1 neuron (DL average user throughput) (Linear activation function).

The used optimizer is ADAM, with learning rate of 0.001 and batch size of 50.

[2]Our approach presented in this paper can be readily generalized to any other measure of users' QoE.

It should be noted that number of hidden layers, neurons, batch size, optimizer and learning rate are all hyper-parameters that are selected after several trials.

A layout of the used network architecture can be found in Fig. 3

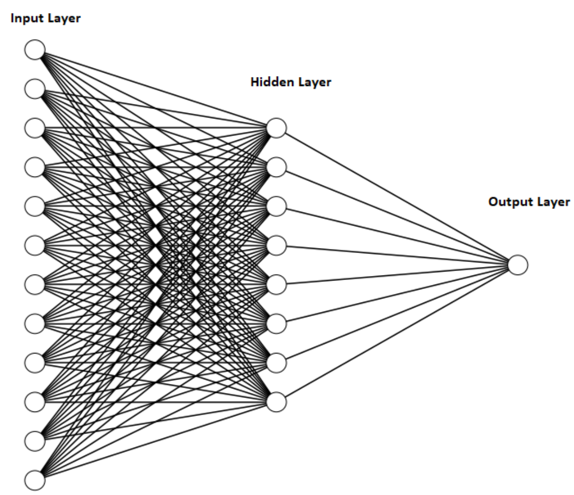

Fig. 3: Neural Network Architecture

2) Recurrent Neural Network: The used RNN consists of:

- 1 input layer containing 50 neurons (ReLu activation function).
- 1 hidden layer containing 100 neurons (ReLu activation function).
- 1 output layer containing 1 neuron (DL average user throughput) (Linear activation function).
- Feedback connection from the previous hidden layer output to the current hidden layer input to track the historical features of the data.

The used optimizer is ADAM, with learning rate of 0.001 and batch size of 72.

It should be noted that the available data are KPIs measures at every hour from different cells. To use these data to train the RNN, we assume that the historical trend of data does not differ from one site to another. Therefore, we separate the data from different cells, make sure they are arranged from oldest to newest and use these sequences to train the machine cell after cell. After each sequence of data, the trained machine is used as a starting point for the next sequence drawn from the next mobile network cell.

C. The Algorithm

The proposed scheme is a regression model that takes the available KPIs as inputs to the Neural Network (NN) and the DL user average throughput as the output. The main idea is to create an NN that is trained via the SISO data. This results in a network that has learnt the performance and behavior of SISO sites. Eventually, when the network is fully trained, it can take the KPIs of the MIMO site under consideration and use its KPIs to predict the DL average throughput using the SISO behavior learned by the network. If the predicted output is acceptable (i.e. achieves a satisfactory users' QoE), which means that SISO is good enough to handle the network traffic, then MIMO can be turned off (temporarily) to save energy. Otherwise, energy must be expended to keep the MIMO on to maintain a satisfactory users' QoE.

In our work, we define a satisfactory user QoE as achieving a user average DL throughput that is not less than 5 Mbps. Therefore, the output of the ML network is compared to a threshold of 5 Mbps to decide whether SISO can provide an acceptable QoE. Clearly, there is a trade-off between the threshold that represents the satisfactory QoE and the amount of saved energy.

Our proposed algorithm is described in Algorithm 1. This description applies to both the MLP and RNN architectures.

Algorithm 1 Using MLP and RNN to save energy in 4G cellular networks by optimizing MIMO usage: algorithm description

procedure PREPARE DATA SET
    Normalization of the data to the range from 0 to 1.
    Divide data set into: 70% training, 20% validation and 10% testing.
end procedure
procedure TRAINING
    Initialize: Random weights.
    Input: 5 KPIs provided by the training dataset.
    Output: DL average user throughput.
    back-propagation Learning Technique:
    while Validation error is decreasing **do**
        for Each training epoch **do**
            Monitor the output (DL average user throughput).
            Compute MAE based on current weights and current input.
            Update Weights based on the MAE and learning rate.
            Monitor validation error to avoid overfitting.
        end for
    end while
    Result: A machine that has learned the features (the historical trend in case of RNN) of SISO scheme.
end procedure
procedure TESTING
    Input: 5 KPIs from the testing data.
    Output: DL average user throughput.
    RUN NN in feed-forward direction.
    Compare predicted throughput to actual throughput.
    while Results are not satisfying **do**
        Change hyper parameters.
        Repeat training phase.
    end while
end procedure
procedure APPLICATION
    Use the fully trained machine to use the KPIs of a certain MIMO cell as inputs and apply regression to predict the output (DL average user throughput) if SISO scheme is used.
    If output is satisfying ($> 5Mbps$), MIMO is turned OFF to save energy. Otherwise, MIMO is used.
end procedure

V. Performance Evaluation

In this section, the performance evaluation results of the proposed schemes are presented.

A. Training Phase

The progress of the training and validation error during the training phase is shown in Fig. 4. It can be seen that the training error is slightly lower than the validation error. Both errors exhibit a decreasing trend. An early halt of the training process occurs when the validation error starts to increase while the training error is still decreasing (to avoid overfitting).

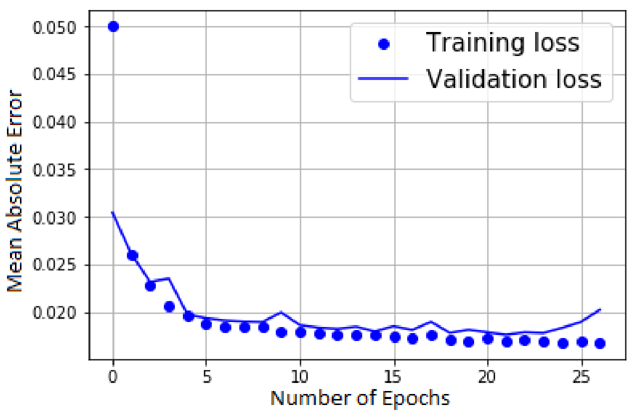

Fig. 4: Training and Validation errors (in Mbps) during the training process of MLP

At the end of the training phase, the performance of the machine is measured via the loss function. The Mean Absolute Error (MAE) is the chosen loss function during the training. Since our aim is to build a regression model to estimate the user DL average throughput, it is reasonable to consider MAE as our performance measure. The results are stated in Table I. The results show that our trained models are able to achieve very small MAE errors and that the RNN architecture was able to achieve smaller errors in general.

TABLE I: Mean Absolute Error for Data Normalized from 0 to 1

Machine	Training Phase	Validation Phase	Testing Phase
MLP	0.0962	0.0985	0.1702
RNN	0.0289	0.0304	0.0993

B. Testing Phase

When the machine completes the training phase, it is ready to be tested. In the testing phase, the machine uses new data (different from the training and validation data). MAE of the testing phase is recorded in Table I.

C. Testing using data from MIMO sites

To test the machine on MIMO sites, a group of MIMO sites are selected for the testing process by the service provider. MIMO is turned off in these sites and the DL average user throughput is recorded. The KPIs of these sites are fed to the proposed machines to predict the DL average user throughput and compared to the recorded actual throughput.

The MAE of this test is 0.31 for MLP and 0.26 for RNN. It should be noted that data were originally normalized to the range from 0 to 1. This is necessary when features have different ranges, like the case here. To have a look at the error function related to the original throughput value, normalization must be inverted. After denormalization, MAE reached the value of 2.75 Mbps for MLP and 1.38 Mbps for RNN.

It is worth mentioning that the sites used for training, validation and primary testing are SISO sites. On the other hand, the sites used in this second testing phase are MIMO sites which are different from the sites used in the previous processes.

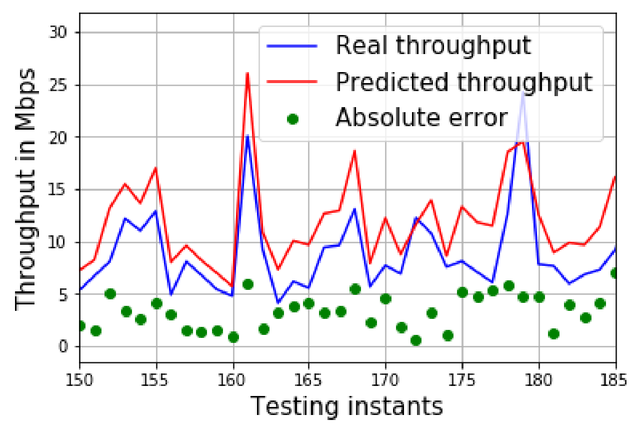

Fig. 5: Actual throughput, predicted throughput and the absolute error during a portion of the testing phase for MLP.

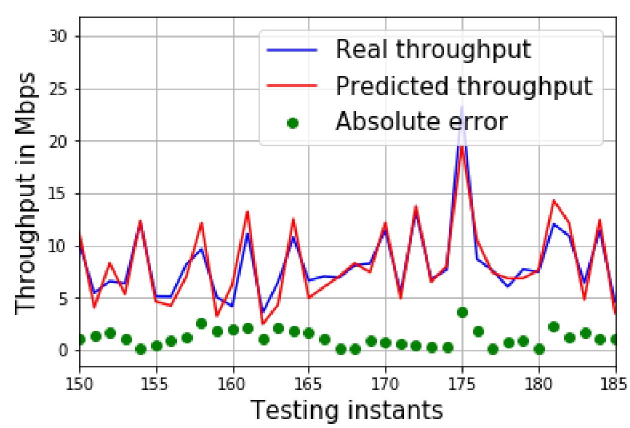

Fig. 6: Actual throughput, predicted throughput and the absolute error during a portion of the testing phase for RNN.

Fig. 5 and Fig. 6 show how close the predicted throughput (output of the machine) is to the actual throughput (provided by the data). They also show the absolute error at every point.

It should be noticed that the considered KPIs are not the only factors that affect the throughput. For example, the weather conditions, the location of the site, the interference and surrounding construction can all affect the achieved throughput. Considering more features is left for extending the current work.

As mentioned earlier, the threshold for acceptable QoE is chosen to be 5 Mbps. The percentage of correct decision (to turn MIMO on/off) for both the MLP and RNN architectures are presented in Table II. This result is important because the main objective of this paper is to take a correct decision to turn MIMO on/off not to predict the exact DL user average throughput.

TABLE II: Percentage of Correct and Erroneous Decisions

Decision	MLP	RNN
Correct Decision to turn MIMO on/off	88.21%	90.17%
Erroneous Decision to turn off MIMO	3.61%	2.55%
Erroneous Decision to keep MIMO on	8.18%	7.28%

VI. CONCLUSION

In this paper, we consider the use of ML-based approaches for energy saving in mobile networks. More specifically, these approaches help in deciding on the disabling MIMO schemes if the SISO mode can achieve a network user desired QoE. We propose two different architectures for ML networks to estimate the user DL average throughput under SISO base stations, namely, multi-layer perceptron (MLP) and recurrent neural network (RNN). We compare the estimated SISO user DL average throughput to a predefined threshold that represents the desired QoE. If SISO can achieve the target QoE, MIMO schemes are turned OFF, otherwise, MIMO schemes are enabled. We train our models based on real mobile network data. Results reveal the efficiency and effectiveness of proposed approaches as they allow for real-time, data-driven solutions. These are unique features of our proposed solutions as compared to the current practices in mobile networks.

REFERENCES

[1] Stefano Buzzi, I Chih-Lin, Thierry E Klein, H Vincent Poor, Chenyang Yang, and Alessio Zappone, "A survey of energy-efficient techniques for 5g networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 697–709, 2016.

[2] Junaid Ahmed Khan, Hassaan Khaliq Qureshi, and Adnan Iqbal, "Energy management in wireless sensor networks: A survey," *Computers & Electrical Engineering*, vol. 41, pp. 159–176, 2015.

[3] Albrecht Fehske, Gerhard Fettweis, Jens Malmodin, and Gergely Biczok, "The global footprint of mobile communications: The ecological and economic perspective," *IEEE Communications Magazine*, vol. 49, no. 8, pp. 55–62, 2011.

[4] Margot Deruyck, Willem Vereecken, Emmeric Tanghe, Wout Joseph, Mario Pickavet, Luc Martens, and Piet Demeester, "Power consumption in wireless access network," in *2010 European Wireless Conference (EW)*. IEEE, 2010, pp. 924–931.

[5] Thomas L Marzetta, "Massive mimo: an introduction," *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[6] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2016.

[7] Mingzhe Chen, Ursula Challita, Walid Saad, Changchuan Yin, and Mérouane Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[8] Tao Han and Nirwan Ansari, "Powering mobile networks with green energy," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 90–96, 2014.

[9] Antonio Spagnuolo, Antonio Petraglia, Carmela Vetromile, Roberto Formosi, and Carmine Lubritto, "Monitoring and optimization of energy consumption of base transceiver stations," *Energy*, vol. 81, pp. 286–293, 2015.

[10] Josip Lorincz, Tonko Garma, and Goran Petrovic, "Measurements and modelling of base station power consumption under real traffic loads," *Sensors*, vol. 12, no. 4, pp. 4281–4310, 2012.

[11] N Faruk, AA Ayeni, MY Muhammad, LA Olawoyin, A Abdulkarim, J Agbakoba, and MO Olufemi, "Techniques for minimizing power consumption of base transceiver station in mobile cellular systems," *International Journal of Sustainability*, vol. 2, no. 1, pp. 1–11, 2013.

[12] Pol Blasco and Deniz Gündüz, "Learning-based optimization of cache content in a small cell base station," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1897–1903.

[13] Kunal Shah and Mohan Kumar, "Distributed independent reinforcement learning (dirl) approach to resource management in wireless sensor networks," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 2007, pp. 1–9.

[14] Yi Liu, Yan Zhang, Rong Yu, and Shengli Xie, "Integrated energy and spectrum harvesting for 5g wireless communications," *IEEE Network*, vol. 29, no. 3, pp. 75–81, 2015.

[15] Tao Chen, Haesik Kim, and Yang Yang, "Energy efficiency metrics for green wireless communications," in *2010 International Conference on Wireless Communications & Signal Processing (WCSP)*. IEEE, 2010, pp. 1–6.

[16] Matt W Gardner and SR Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[17] Nevio Benvenuto and Francesco Piazza, "On the complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 967–969, 1992.

[18] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.