Deep Neural Networks approach for Power Head-Room Predictions in 5G Networks and Beyond

Nazih Salhab, Rana Rahim, Rami Langar, Raouf Boutaba

Deep Neural Networks approach for Power Head-Room Predictions in 5G Networks and Beyond

Nazih Salhab*‡, Rana Rahim‡, Rami Langar*, Raouf Boutaba§
* University Gustave Eiffel, LIGM-CNRS UMR 8049, F-77454, Marne-la-Vallée, France
‡ LTRM, Faculty of Science, Lebanese University, Tripoli, Lebanon
§ David R. Cheriton School, University of Waterloo, Waterloo, ON, Canada
E-mails: nazih.salhab@univ-eiffel.fr; rana.rahim@ul.edu.lb; rami.langar@univ-eiffel.fr, rboutaba@uwaterloo.ca

***Abstract*—The enhanced Interference Mitigation and Traffic Adaptation (eIMTA) mechanism is a key enabler for 5G networks and beyond. Knowing that a User Equipment (UE) cannot transmit more than the maximum power allowed by its power class. The level of available Transmission Power (TP) in each UE is an essential input for the Uplink (UL) scheduler of the next Generation Node-B (gNB). Scheduling higher data rate than what is supported by the available TP is a waste of resources. In the Downlink (DL), the power level is known by the gNB that manages the power amplifier and the DL-scheduler. Conversely, in the UL, the available power is estimated by the UE and sent to the gNB as a key input for eIMTA, known as Power Head-Room (PHR). In this context, we propose in this paper, a Deep Neural Network (DNN) based model to predict the PHR and reduce dependency on reported measures. We evaluate the effectiveness of our proposal in a 5G experimental prototype, based on Open Air Interface (OAI). Obtained results show that using DNN outperforms state-of-the-art machine-learning approaches in terms of prediction accuracy, computation complexity and throughput.**

***Index Terms*—Deep Neural Networks, Machine Learning, Enhanced Interference Mitigation and Traffic Adaptation eIMTA, 5G, OpenAirInterface OAI, Power Head-Room prediction.**

I. Introduction

In Long Term Evolution-Advanced (LTE-A) and fifth Generation of mobile communication (5G) New Radio (NR), Power Head-Room (PHR) is a type of Medium Access Layer (MAC) Control Element (CE). It reports the headroom between the current User Equipment (UE) Transmission Power (TP) and the nominal power. The Evolved Node-B (eNB) in LTE-A and next Generation Node-B (gNB) in 5G use this reported value to estimate the Uplink (UL) bandwidth that a UE can use for a specific subframe [1]. The more resource blocks the UE uses, the higher UE TP gets, without exceeding the maximum power defined in the UE class specification. Consequently, a UE cannot use allocated resource blocks from assigned bandwidth if it does not have enough PHR.

The "enhanced Interference Mitigation and Traffic Adaptation" (eIMTA) allows for very dynamic adaptation of the Time Division Duplexing (TDD) pattern in response to capacity requirements in the UL and Downlink (DL). eIMTA was standardized in Third Generation Partnership Project (3GPP) Technical specification for LTE-A Release 12 [2]. "eIMTA-like" functionality is considered to be one of the key enablers for 5G technologies and beyond [3]. Even though the PHR is reported per subframe, the eIMTA time division duplexing has two sets of UL power control subframes configured. Therefore, PHR can also be very different between the two subframe sets [4]. According to [5, 6], a power headroom report can only be sent in sub-frames in which a UE has an UL transmission grant. Furthermore, the reported value corresponds to the subframe in which it is sent. Therefore, it is well-suited to predict the PHR value. Indeed, the UE cannot directly measure its actual transmission power headroom for the subframe in which the report is to be transmitted. In this context, we propose in this article to use deep neural networks (DNNs) in order to predict future PHR values in 5G. These DNNs are formed by the accumulated wealth of data in the gNB from previous measurements.

Our contributions can be summarized as follows:

- We provide an analytical PHR model to identify the predictors for a feature-based prediction.
- We formulate different Machine Learning (ML) approaches to predict the PHR values and evaluate them.
- We show the effectiveness of our proposal based on real measurement of PHR values using our 5G experimental prototype based on OpenAirInterface™ (OAI) [7].

The remainder of this paper is organized as follows. In section II, we present the related works. Section III describes our system model followed by our Problem Formulation and complexity Analysis in Section IV. In section V, we present the performance evaluation and we discuss obtained results. We finally conclude this paper in section VI.

II. Related Work

In this section, we discuss a selection of relevant papers that investigated PHR reporting for 5G networks and beyond.

Authors in [3] analyzed the main characteristics of eIMTA and illustrated its behavior using system-level simulations. The center for Advanced Technology in Telecommunications (CATT) along to multiple telecommunications suppliers [4] discussed the possibility of getting PHR for a set of two subframes. The goal is to use the current PHR mechanism in an eIMTA context. They concluded by emphasizing on the importance of having the eNB in control of individual PHR for each set.

Authors in [8] proposed a power-efficient resource allocation scheme using power headroom reports. They elaborated an adaptive Open-loop power control scheme based on the Signal-To-Interference-Ratio (SINR) and the UL interference in an

ISBN 978-3-903176-28-7

aim to improve cell capacity. They also detailed a fast closed-loop power control based on received SINR.

Authors in [9] relied on PHR to design autonomous cell-centered self-optimizing-network. All of these papers exploited the PHR and did not address aforementioned eIMTA challenges using a proactive PHR computation approach.

Authors in [10] proposed an inter-cell radio frame coordination scheme based on sliding codebook for fully dynamic TDD 5G networks. They formulated a two-objective optimization problem aimed at minimizing the average Inter-cell cross link interference while maximizing the achievable UL/DL capacity leveraging eIMTA for flexible TDD adaptation.

Authors in [11] proposed an optimization of the Physical Uplink Shared Channel (PUSCH) closed loop power control algorithm based on PHR. They developed an algorithm that calculates the power of each resource block once PHR measurement is received by the eNB/gNB.

Authors in [12] investigated how to leverage PHR to detect outages in LTE network in an aim to minimize costly drive-test missions for mobile network operators. They used a dynamic LTE system simulator to validate their proposal. However, they did not leverage the advents in ML based prediction.

Authors in [13] proposed a prediction scheme for channel stability using machine learning for application in 5G networks.

Finally, authors in [14] investigated dual connectivity in LTE networks by allowing UEs to connect to multiple eNBs in an aim to improve user throughput and mobility. They emphasized on the resulting technical challenges including PHR calculation and reporting. They relied on a system level simulation study to quantify their analysis.

III. System Model

PHR reporting provides the serving eNB/gNB with information about the difference between the nominal UE maximum transmit power and the estimated power for Uplink Shared Channel (UL-SCH) transmission for the serving cell.

According to the standard [15], the UEs transmit buffer status and PHR reports on the PUSCH in the following four cases. The first case is when the path loss changes above a predefined threshold ($\mathrm{dl-PathlossChange}$ in dB) at each time window defined by $\mathrm{prohibitPHR-Timer}$. The second case is at the expiration of a certain periodic timer ($\mathrm{periodicPHR-Timer}$). The third case is when the configuration or reconfiguration of the PHR by the upper layers occurs. And finally, when a secondary servicing cell is activated.

Note that the PHR reporting range is from -23 to +40 dB [15]. Positive values indicate that there is still some headroom under the maximum allowed power. It implies that the UE can transmit more data if required and if approved by the eNB/gNB UL scheduler. In contrast, negative power headroom values indicate that the per-carrier transmit power was limited at the time of the power headroom reporting. This meaning that the network has scheduled a higher data rate than the terminal can support, given the available transmission power. In such case, allocated resources are wasted.

In what follows, we provide the analytical model of PHR to identify the parameters that affect it for use in our ML engineering models.

A. Analytical Model for PHR

According to the standard on physical layer procedures [16], the UE computes its UL Transmission Power P_{TX} as follows:

$$P_{\mathrm{TX}} = min \begin{Bmatrix} P_{\max}, \\ P_0 + 10log_{10}(M) + \alpha.\mathrm{PL} + \Delta_{\mathrm{MCS}} + f_c \end{Bmatrix}$$

It is the minimum between required power and $P_{\max}$. On first hand, the maximum possible output power of the UE is fixed according to its category or class that defines its performance specifications. For example, $P_{\max}$ is of 23, 20, or 14 dBm for UE class 3, 5, or 6, respectively [17]. On second hand, the required power is proportional to all of: P_0 (SINR per single Physical Resource Block (PRB)), the number of transmitting PRBs (M), the estimated Path-loss (PL) weighted by a pre-configured fractional PL compensation factor (α), the adjustment factor for higher Modulation and Coding Schemes (Δ_{MCS}) and finally, the closed loop power-control adjustment factor (f_c). Parameter f_c is the value commanded by the eNB/gNB to the UE to adjust its transmission power according to received signal strength and quality indicators at the eNB/gNB. We consider P_0 to be constant for an operator in homogeneous environment and $f_c \sim 0$ for a fine-tuned cell [16]. We assume parameter α to be constant per type of environment (urban, suburban or rural) as it relates to the UE used frequency band and the number of neighbor cells to a hosting cell, which is our case [16]. From the P_{TX}, the PHR is calculated as $\mathrm{PHR} = \tilde{P}_{\max} - P_{\mathrm{TX}}$ [16], where $\tilde{P}_{max}$ is a computed transmission power set by the UE based on requirements elaborated in [17].

B. ML Models

We employ multiple ML models having different complexities to evaluate the prediction efficiency for the PHR. In addition, we used the neural networks-based models. We evaluated two exogenous models with selected features and two endogenous models with no required external feature engineering to be attached to the model.

Let $\mathbb{D} = \{(X^{(1)}, y^{(1)}), (X^{(2)}, y^{(2)}), \ldots, (X^{(N)}, y^{(N)})\}$ be a dataset of N records. $X = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ is a vector with d features used as predictors. Variable y is our response for a predicted PHR value.

1) Trees and Random-Forest based Models: A tree $\mathbb{T}$ is a form of a directed graph with a Vertex set $(\mathbb{V})$ and Edge set $\mathbb{E}$, such that $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$. It is a particular graph that has no simple cycles and is completely connected. Let $\mathbb{T} = (\mathbb{V}, \mathbb{E})$, where each $v \in \mathbb{V}$ has up to two descendants $\{v_l, v_r\}$ for left and right. For each vertex $v \in \mathbb{V}$, we denote by $val(v)$ a value of the vertex v, by $idx(v)$, the feature index of vertex v, and by $desc(v)$, the number of descendants of v. Thus, we define the vertex function $f(v, x)$ as follows.

$$f(v,x) = \begin{cases} f(v_l, x) & , x_{idx(v)} \leq val(v) \wedge desc(v) = 2 \\ f(v_r, x) & , x_{idx(v)} > val(v) \wedge desc(v) = 2 \\ val(v) & , \text{otherwise} \end{cases}$$

Operator $\wedge$ denotes the logical "And" operator between the conditions. The growing function of a single tree T starting from vertex v_0 is $T(x) = f(v_0, x)$. In practice, regression trees

are grown such that we have minimum squared error between the value y and the estimated value by the tree $T(x)$. A random forest $\mathbb{F}$ is a set of M trees $\mathbb{T} = \{T_1, T_2, \ldots, T_M\}$, where each tree T_i is grown on a subset of the original dataset $\mathbb{D}$ used for the training of the ML model.

2) Neural Networks: Nonlinear Auto-Regressive (NAR) is a neural network based ML technique. It may be used with Exogenous input (NARX) or without relying on hand-crafted features and parameters. In this case, it is only based on previous values of the time series.

NAR uses differentiable functions to model the system leveraging numerical optimization methods. When used without exogenous input, NAR might discover unknown high-level features from the data itself. It is a kind of dynamic filtering, in which past values of one or more time series are used to predict future values. These dynamic neural networks, which include tapped delay lines are widely used for nonlinear filtering and prediction. Provided d past values (delay parameter), on first hand, NARX predicts the series $\tilde{y}_t$ based on a series of features X_t. On the other hand, NAR predicts series values $\tilde{y}_t$ given only d past values of $\tilde{y}_t$ or y_t, in an open-loop or closed-loop scenario, respectively.

Recursive Neural Network (RNN) is a form of DNN that is well-suited for time-series prediction. Due to its recurrent structure, a cell state variable c_t is updated according to current input i_t and previous state c_{t-1}. Given a time series sequence $x = (x_1, x_2, \ldots, x_t)$, the joint probability is given by: $p(x_1, x_2, \ldots, x_t) = p(x_1).p(x_2|x_1)\ldots.p(x_t|x_1, \ldots, x_{t-1})$. The hidden layer of the RNN model provides an output which is $h_t \sim p(x_t|x_1, \ldots, x_{t-1})$. RNNs are trained using stochastic gradient descent or its similar variants. Long Short Term Memory (LSTM) was proposed [18] to avoid having vanishing or exploding gradient. Differently from the RNN, where a repeating module has a single hyperbolic tangent $\tanh$ layer, LSTM has four layers (forget f_t, input i_t, candidate $\hat{c}_t$, and output o_t). They are used to calculate the new cell state (c_t) and the new hidden output (h_t). Each layer uses an activation function that is either the logistic sigmoid (σ) or $\tanh$ of the weighted entry (w) topped by a bias (b). Thus, it is capable of learning Long-term dependencies using Short-Term Memory. The LSTM unit is depicted in Fig. 1 and formulated as follows.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{1a}$$
$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{1b}$$
$$\hat{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{1c}$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{1d}$$
$$c_t = f_t.c_{t-1} + i_t.\hat{c}_t \tag{1e}$$
$$h_t = o_t.\tanh(c_t) \tag{1f}$$

IV. Problem Formulation and complexity Analysis

In this section, we formulate the problem of predicting PHR values as an optimization problem and analyze the complexity of each of the studied ML models. Recall that our goal is to estimate a function $\tilde{y} = \langle \beta, X \rangle$ such that $(\tilde{y} \approx y)$ where the regularized error is minimum. Using Ridge regression

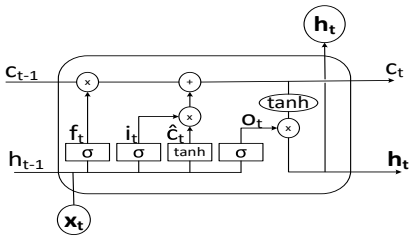


Fig. 1: Long Short Term Memory Modular Diagram

formulation, the controlled version of the objective function Z can be expressed as follows.

$$\min_{\beta \in \mathbb{R}^d} Z = \lambda \|\beta\|^2 + \frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i - y_i)^2 \tag{2}$$

where $\beta \in \mathbb{R}^d$ is the regression parameter vector and $\lambda \in \mathbb{R}$ is a scalar allowing to control the complexity of the forecast according to the chosen set of parameters. We notice that for $\lambda = 0$, the effect of the parameters is neglected and only the Mean Squared Error (MSE) is considered (the second term), while for high values of λ, the null β leads us to our goal.

Let us now assess the complexity of each of the studied models. Starting from ML-based regressions, and denoting by Y all the $y^{(i)}, i = 1..N$, the computation of β is about completing the following matrix operation $(X^T X)^{-1} X^T Y$. Thus, the time complexity is: $\mathcal{O}(Nd^2 + d^3)$, where Nd^2 results from the product $X^T X$ and the d^3 results from the matrix inversion of $X^T X$. As for the space complexity, it is $\mathcal{O}(Nd + d^2)$, as storing $X^T X$ and its inverse is $\mathcal{O}(d^2)$ and storing X costs $\mathcal{O}(Nd)$ floats.

As for the tree-based regressions: we denote by D the depth of the deepest tree in the forest $\mathbb{F}$. Knowing that the number of nodes in a binary tree is $(2^{D+1} - 1)$, we can say that the forest $\mathbb{F}$ has a worst space complexity of $\mathcal{O}(2^D M)$. Access time complexity is $\mathcal{O}(\log(n))$, where n is the number of nodes.

Finally, for the Neural Networks (NN) complexity, recall first that the computation of NN consists of two parts: a forward and a backward pass. In the forward pass, the computation consists of additions, subtraction and multiplication or memory operations. In the backward pass, there is a need to compute the difference between the output and target value, and propagate the gradient all the way back to the beginning. Working on N rows, a matrix multiplication has a asymptotic run-time of $\mathcal{O}(N^3)$. Denoting by n_{layers} the number of layers, the time complexity in the forward pass is $\mathcal{O}(N^3.n_{\text{layers}})$. The complexity of activation operations is $\mathcal{O}(N.n_{\text{layers}})$. Knowing that $\forall N \geq 1, N^3 + N \leq 2N^3$, we simplify the summation $(N^3 + N).n_{\text{layers}})$ to conclude that the total run-time complexity in the forward pass is $\mathcal{O}(N^3.n_{\text{layers}})$. Similarly, the time complexity of the backward pass is $\mathcal{O}(N^3.n_{\text{layers}}.n_{\text{gi}})$, where n_{gi} is the count of gradient iterations.

V. Performance Evaluation

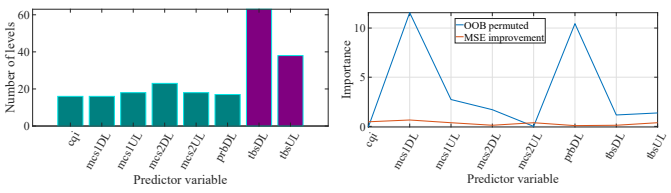
In this section, we evaluate the performance of our proposed models using an OAI-based [19] 5G experimental prototype. We present our methodology to generate the dataset used in our simulations, followed by a presentation of the results.

A. Dataset Generation and Simulation Environment

Using our 5G prototype [7], a test dataset has been generated as follows. During simulation time (Sim. time), Metrics, collected every second from FlexRAN [19], include configuration data of the two UEs and the hosting cell/slice parameters in addition to performance data of the MAC layer. We extracted a subset of the collected data, including all of: PHR, Channel Quality Indicator (CQI), MCS, number of PRBs and Transport Block Size (TBS). The ML models are implemented in MATLAB [20].

B. Simulation Scenarios

Fig. 2 depicts the features importance in terms of the predictor variables. In order to cater different prediction models including offline and online scenarios, we identified three features subsets as follows. 1) Full set of Features (FF) in which the prediction process depends on all features. 2) Reduced set of Features (RF), where the prediction process depends only on the two most important features. And 3) No Features set (NF) where the model is independent from any engineered feature and just relies on previous observed values. We note that for ML-based regressions, and in particular the tree-based ones, the algorithm, proposed in our earlier work [7] tends to split predictors with many unique values (levels). It prioritizes continuous variables, over discrete ones with fewer levels as depicted in Fig. 2a. As our data is heterogeneous, we considered using the curvature, known as interaction tests for split-predictor selection. To decrease the features from our dataset, we compute the predictors importance values by permuting Out-Of-Bag (OOB) observations among the trees and summing gains in the MSE due to splits on each predictor, as depicted in Fig. 2b. Since prediction time increases with the number of predictors in trees and forests, we created a model using as few predictors as possible. Thus, we evaluated the performance of Random forests using the best two predictors only (RF) and the neural networks with NF. We considered the neural networks with NF and we evaluated the (NAR) with 3 delays using multiple training methods. Note that the delay number is equal to the number of stored previous values of y. Finally, we considered the RNN LSTM using multiple optimizers. We evaluate the Prediction accuracy based on the MSE, Root-Mean Squared Error (RMSE), Mean Absolute Error (MAE), coefficient of determination R-squared (R^2) that are all centered around calculating the gap between the predicted value and the observed one. Unlike the MAE that treats all errors uniformly, the RMSE stresses on large prediction errors and de-stresses on small ones. Finally, the R^2 measures the proximity of data to the fitted regression line.



(a) Number of levels among predictors (b) Predictor importance estimation

Fig. 2: Deducing Reduced Features out of Full Features

TABLE I: Simulation Parameters and Values

Parameter	Value	Parameter	Value
k	10	NL	30
q (Fine)	4	ker (Fine)	0.71
q (Medium)	12	ker (Medium)	2.8
q (Coarse)	36	ker (Coarse)	11
q (Ensemble)	8	LR	0.1
d	2 ‖ 3	ILR	0.005
N_h	200	LRDP	125
N_{epoch}	30	LRDF	0.2
GT	8	Sim. time (s)	37073

TABLE II: Full Features (FF) Simulation Results

Class	Forecast	RMSE	R^2	MAE
Linear	Linear	5.2874	0.59	4.0795
	Interaction Linear	5.0786	0.62	3.8216
	Robust Linear	5.3929	0.57	3.989
	Stepwise Linear	5.0209	0.63	3.8134
Tree	Fine	4.3661	0.72	30953
	Medium	4.4468	0.72	3.1601
	Coarse	4.4964	0.7	3.227
SVM	Linear	5.4060	0.57	3.9732
	Quadratic	5.0878	0.62	3.6559
	Cubic	16.438	0.297	9.7541
	Fine Gaussian	4.7019	0.68	3.3028
	Medium Gaussian	4.8889	0.65	3.5253
	Coarse Gaussian	5.3101	0.59	3.975
Gaussian Process	Rational Quadratic	4.3581	0.72	3.1127
	Square Exponential	4.4555	0.71	3.1712
	Matern 5/2	4.4401	0.71	3.1657
	Exponential	4.3712	0.72	3.1203
Ensemble of Trees	Boosted Trees	4.6415	0.68	3.5402
	Bagged Trees	5.3909	0.57	4.1355
Random Forests	Full Features (FF)	4.3293	0.73	3.0778
	Reduced Features (RF)	5.0287	0.63	3.8435
NARX 3 delays	Levenberg-Marquardt	5.86	0.9	4.941
	Bayesian Regulation	5.8934	0.9	4.672
	Scaled Conjugate Gradient	6.5338	0.9	4.651

C. Simulation Results

In order to validate our predicted output with the FF, we used k-fold cross-validation that consists of partitioning the original sample of data into k equal sized sub-samples. A single sub-sample is retained as the validation data for testing the model, and the remaining $(k-1)$ sub-samples are used as training data. The cross-validation process is then repeated k times, with each of the k sub-samples used exactly once as the validation data. The k results are then averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

For single trees, we considered different values q of minimum leaf size to denote the type of the tree (Fine, Medium or Coarse). As for the ensemble of trees (Boosted or Bagged), we used a number of learner NL and a learning rate LR. For Gaussian Support Vector Machine (SVM), we used different kernel scales (ker) according to its type (Fine, Medium or Coarse). For LSTM, we considered a number of hidden units (N_h), Number of Epochs (N_{epoch}), Gradient Threshold (GT), Initial Learning Rate (ILR), Learning Rate Drop Period (LRDP), and Learning Rate Drop Factor (LRDF). Simulation parameters are listed in Table I.

Tables II and III report the RMSE, R^2 and MAE performance metrics for the used prediction models. We can see

TABLE III: No Features (NF) Simulation Results

Class	Forecast	RMSE	R^2	MAE
NAR	Levenberg-Marquardt	5.8700	0.90	4.198
3 delays	Bayesian Regulation	4.2000	0.89	4.489
	Scaled Conjugate Gr.	6.6500	0.88	4.91
RNN	ADAM Optimizer	2.9652	0.69	2.0538
LSTM	SGDM Optimizer	3.1063	0.63	2.2143
	RMSProp Optimizer	3.0281	0.66	2.1658

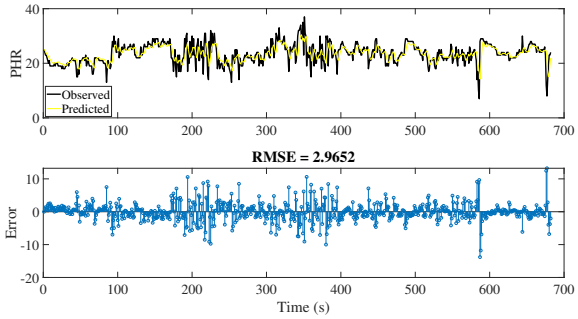


Fig. 3: Observed vs Predicted and Error Comparison over time

that LSTM using ADAM optimizer provides the lowest RMSE. LSTM are efficient at extracting patterns in input feature space that spans over long sequences, as it leverages its memory state to predict, which is ideal for our recurrent PHR prediction.

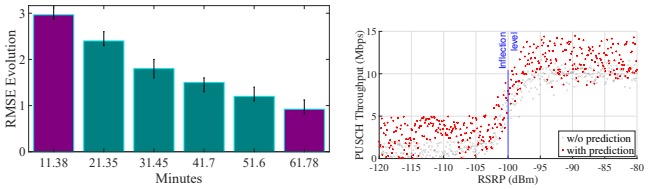
Fig. 3 depicts the observed PHR using OAI and predicted PHR using LSTM for 682 seconds (∼10 minutes). Visually, inspecting the trends, we can see that the evolution of both graphs are very close except when it comes to sudden changes shown in high peaks or drops triggered by external environmental factors. This is clearly seen in the bottom subplot showing that the instant value of the error with less than 10 units in magnitude, during these first ∼10 minutes.

In addition, we can observe in Fig. 4a that the RMSE is decreasing over time, as a result of continuous improvement on the long run, which proves its suitability for application. Indeed, when additional training data are accumulated, the prediction accuracy will be enhanced.

Finally, we plot in Fig. 4b the PUSCH throughput versus the Received Signal Reference Power (RSRP). We can see that the RSRP value of -100 dBm acts as an inflection level separating two zones of throughput value. The throughput is enhanced with prediction as a consequence of an early awareness by the eNB/gNB of required PRBs corresponding to the UE output power. We can explain this behavior as follows. When the RSRP is less than -100 dBm (poor radio conditions), the transmitted power of the UE is increased. This causes higher interference and thus the number of PRBs is decreased, limiting the data rate. Conversely, for higher values of RSRP (above -100 dBm), the throughput is increased till reaching a saturation level as it is affected by the MCS and the buffering of the operating system. All in all, we can conclude that the prediction of the PHR helps in increasing the throughput and reducing the probability of wasted resources.

VI. Conclusion

In this paper, we have presented several ML-based approaches to predict the power headroom, for an enhanced



(a) RMSE decreasing over time (Minutes) (b) Throughput Comparison

Fig. 4: RMSE convergence with DNN and impact on the Throughput

Interference Mitigation and Traffic Adaptation (eIMTA) application. We formulated Machine Learning Feature-based prediction, in addition to neural networks approaches with and without features. We also considered a reduced set of features and found that it results in minor deteriorated performance. Particularly, we have observed that deep neural networks, Long Short Term Memory (LSTM) requiring no features, outperforms other ML models in terms of prediction accuracy and feature independence. Implementing the LSTM in our 5G prototype using OpenAirInterface (OAI), we observed that the Root Mean Square Error between real and predicted PHR decreases with time and the throughput is enhanced.

Acknowledgement

This work was partially supported by FUI SCORPION project (Grant no. 17/00464), CNRS PRESS project (Grant no. 239953), “Azm & Saade”, and the Lebanese University.

References

[1] A. Kukushkin, *The Road to 5G*. Wiley Online Library, 2018.
[2] “3GPP TS 36.306: User Equipment radio access, Rel. 12,” Jul. 2015.
[3] V. Pauli, Y. Li, and E. Seidel, “Dynamic TDD for LTE-A and 5G,” *Nomor Research GmbH, Munich, Germany, Tech. Rep*, 2015.
[4] Center for Advanced Technology in Telecommunications (CATT), “Summary of discussion PHR for TDD eIMTA,” https://list.etsi.org/.
[5] J. Lohr, H. Suzuki, M. Feuersanger, A. G. E. Von Elbwart, and T. Iwai, “Power headroom reporting for non-scheduled uplink,” Jun. 2015.
[6] J. Loehr, C. Wengerter, M. Feuersaenger, and T. Tamura, “Efficient extended power headroom for semi-persistent scheduling,” Jun. 2016.
[7] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, “Machine learning based resource orchestration for 5g network slices,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, December 2019.
[8] W. Kim, Z. Kaleem, and K. Chang, “Power headroom report-based uplink power control in 3gpp lte-a hetnet,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 233, 2015.
[9] H. Martikainen, I. Viering, and B. Wegmann, “Dynamic range aware LTE uplink P0 optimization in HetNet,” in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.
[10] A. A. Esswie and K. I. Pedersen, “Inter-cell radio frame coordination scheme based on sliding codebook for 5g tdd systems,” in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019.
[11] Z. Yang, K. Liu, H. Xiong, and H. Wei, “An Advanced Power Control Algorithm Based on PHR,” in *International Conference on Computational Intelligence and Communication Networks (CICN)*, Dec 2015.
[12] J. Turkka and J. Puttonen, “Using LTE Power Headroom for Coverage,” in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, Sep. 2011.
[13] S. Bakri, M. Bouaziz, P. A. Frangoudis, and A. Ksentini, “Channel stability prediction to optimize signaling overhead in 5G networks,” in *IEEE International Conference on Communications (ICC)*, June 2020.
[14] S. C. Jha, K. Sivanesan, R. Vannithamby, and A. T. Koc, “Dual connectivity in lte small cell,” in *IEEE Globecom Workshops*, Dec. 2014.
[15] “3GPP TS 36.321: Medium Access Control Protocol, r15,” Oct. 2019.
[16] “3GPP TS 36.213: Physical Layer proc. v15.7.0 (Rel. 15),” Oct. 2019.
[17] “3GPP TS 36.101: UE radio transmission and reception, r15,” Oct. 2019.
[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[19] “OpenAirInterface (OAI),” https://www.openairinterface.org/.
[20] “Matrix Laboratory: MATLAB,” https://mathworks.com/.