Published in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)
DOI: 10.1109/VTCFall.2018.8690629

Machine Learning Based Uplink Transmission Power Prediction for LTE and Upcoming 5G Networks using Passive Downlink Indicators

Robert Falkenberg[1], Benjamin Sliwa[1], Nico Piatkowski[2] and Christian Wietfeld[1]
[1]Communication Networks Institute, [2]Department of Computer Science, AI Group
TU Dortmund University, 44227 Dortmund, Germany
e-mail: {Robert.Falkenberg, Benjamin.Sliwa, Nico Piatkowski, Christian.Wietfeld}@tu-dortmund.de

***Abstract*—Energy-aware system design is an important optimization task for static and mobile Internet of Things (IoT)-based sensor nodes, especially for highly resource-constrained vehicles such as mobile robotic systems. For 4G/5G-based cellular communication systems, the effective transmission power of uplink data transmissions is of crucial importance for the overall system power consumption. Unfortunately, this information is usually hidden within off-the-shelf modems and mobile handsets and can therefore not be exploited for enabling green communication. Moreover, the dynamic transmission power control behavior of the mobile device is not even explicitly modeled in most of the established simulation frameworks. In this paper, we present a novel machine learning-based approach for forecasting the resulting uplink transmission power used for data transmissions based on the available passive network quality indicators and application-level information. The model is derived from comprehensive field measurements of drive tests performed in a public cellular network and can be parameterized for integrating all measurements a given target platform is able to provide into the prediction process. In a comparison of three different machine learning methods, Random-Forest models thoroughly performed best with a mean average error of 3.166 dB. As the absolute sum of errors converges towards zero and falls below 1 dB after 28 predictions in average, the approach is well-suited for long-term power estimations.**

I. Introduction

Energy-constraints are a system-immanent challenge for most Internet of Things (IoT)-based devices that directly affect how long a device can autonomously operate in the field. Therefore, green communication networks aim to optimize the energy-efficiency of data transmissions, since large amounts of the available power resources are spent on communication. In cellular communication systems, such as Long Term Evolution (LTE) and upcoming 5G networks, the consumed energy for data transmissions is mainly determined by the uplink transmission power [1]. However, this crucial information is often not accessible for the application layer and can therefore not be leveraged for designing cross-layer and energy-aware communication mechanisms. Moreover, even most established LTE simulators do not consider detailed models of the consumed energy and focus on simple linear system-level considerations. Although different models for estimating the power consumption for LTE-based data transmissions exist, they rely on indicators that are not accessible in real-world scenarios as they are usually only used internally by the modem firmware and are hidden for users and developers (e.g. the number of assigned Resource Blocks (RBs) and carrier-specific constants). In this paper, we present a novel data-driven model for predicting the Uplink (UL) transmission power (TX-power) of User Equipment (UE) based on machine learning and empirical data from real traces. The proposed prediction model is highly parameterizable and can therefore be configured to only integrate the parameters that are available on a given target platform (e.g. a specific UE or a specific network simulator). Moreover, it is able to implicitly consider hidden variables that are correlated with the available information. Therefore, it is capable of filling the gap between full-featured physical layer measurements and system-level considerations. Fig. 1 provides an illustration of the prediction's data-flow.

This work presents the model in three variants, which differ in the number of available input features for the prediction.

- The *full-featured model* leverages the full feature set and exploits low-level information that is usually only available if the modem is interfaced directly using vendor-specific control commands (e.g. information about the neighboring cells).
- The *practical model* integrates indicators which are typically accessible on off-the-shelf UEs using operating system Application Programming Interface (API) abstractions. Therefore, it can provide additional information about the power consumption e.g. for live evaluation of resource-efficient data transmission schemes [2], [3].

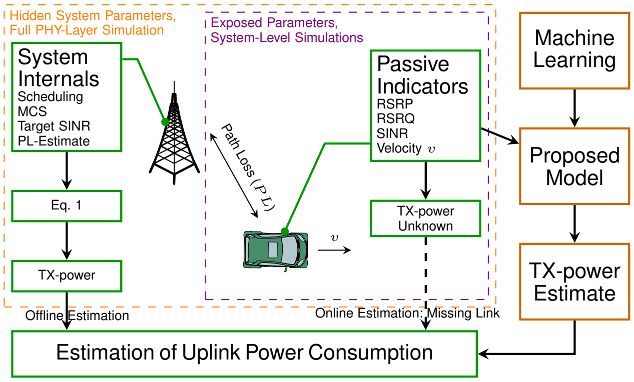

Fig. 1. Uplink power control of UE and its underlying system parameters are typically hidden to the application layers. However, this knowledge is crucial for predictions and estimations of the involved power consumption in energy-aware applications or system-level simulations, respectively. The proposed model derives this information from passive connectivity indicators.

- With the *simulation model*, we provide a lightweight mechanism for simulation frameworks that allows to make a statement about the transmission power of the mobile device even when the simulator itself does not explicitly model TX-power control. In particular, this applies to many established system-level network simulators such as SimuLTE [4] and LTE-Sim [5].

The paper is structured as follows: After giving an overview about different state-of-the-art power consumption models, we present the system model of our approach and relate it to the Context-aware Power Consumption Model (CoPoMo) [6]. Afterward, the setup of the empirical evaluations as well as the machine learning analysis is described and evaluated. Finally, we provide the underlying raw data and the capturing software in an Open Source way to guarantee a high level of transparency and enable the reader to synthesize customized models with respect to available indicators.

II. Related Work

Measuring and optimizing the energy consumption of UEs is a major research topic as the consumed energy for data transmissions is strongly related to the battery lifetime of the mobile device. Consequently, a wide range of different models has been proposed for analyzing the UL and Downlink (DL) power consumption of mobile devices. In [7], the authors propose a power model for data transfer that assumes a linear dependency of power consumption and data rate. However, it lacks a consideration of the radio transceiver's power control. The model in [8] aggregates the power consumption of distinct components of LTE UE, such as base-band processing for transmission and reception and the connected radio transceiver units. It is obtained from empirical measurements of real UEs in the laboratory and depends on parameters such as instant transmission power, received power and the associated data rate in each direction. The CoPoMo [6], however, adds numerous context and system parameters, e.g., environment, mobility, and user activity into the power estimation. Hence, the model does not only reflect the device itself but extends the scope to intrinsic and extrinsic influences that affect the device's average power consumption. Since the main power draw is issued by data transmissions rather than receptions, the model is dominated by influencing factors, which affect the TX-power and transmission time in the UL. In the basic variant, it groups distinct power states into a four-state Markov chain. Transitions between the states are expressed as service rates and arrival rates, which also depend on the radio conditions. However, the model strongly depends on the distribution of required TX-power, which can be obtained either by ray-tracing simulations or from empirical data, such as [9]. While CoPoMo is perfectly suited to provide power estimations in offline simulations, an online application is prevented in most cases by the lack of knowledge about the instant TX-power at higher protocol layers.

In contrast to Commercial Off-the-Shelf (COTS) equipment, Software-Defined Radios (SDRs) such as srsLTE [10] give full insight into the protocol stack and even allow hypothetical computations of values like TX-power at any time based on the complete knowledge of all involved parameters. Advanced approaches even enable a prediction of resource assignments based on the momentary over-all cell load [11]. However, freely available SDR implementations often suffer from limited functionality, e.g., no support for handover, low transmission power, or imperfect channel estimation. Hence they do not adequately represent typical COTS equipment.

In some cases, however, the available system and context information on COTS equipment may suffice to reconstruct or estimate hidden parameters from lower protocol layers by a machine-learning approach. A general survey on the potential of machine learning for current and future mobile networks is given in [12]. It is part of the related broader topic of anticipatory networking [13], where network infrastructure and terminals act proactively according to learned or estimated user behavior and network characteristics. While most of those approaches require cross-layer integration into multiple network components, this paper aims at a lightweight prediction model at application layer, which is suitable for both, online applications and system-level simulations, without any requirement for modifying existing network infrastructure.

III. Machine-Learning Based Solution Approach

For performing a prediction of the TX-power, an identification of an indicator set is required in a first step, which reliably reflects the influencing factors on the power control of the UE.

These indicators have to be accessible at the application layer on most LTE handsets for online applications and to be available in system-level simulators for offline applications. However, taking the data rate as an example, such application-layer indicators might be blurred by a mixture of multiple influences on the radio system (e.g., signal quality and congestion) and the intermediate operating system (e.g., buffering and Transmission Control Protocol (TCP) slow start mechanism). Therefore, it requires a collection of multiple application-layer indicators which are influenced by the same underlying effect to deduce the true cause behind the curtain of abstraction.

The second step involves machine learning to derive the complex relationship between the set of collected indicators, namely features, on the resulting TX-power of the device. The required data for this task is obtained from real traces as explained in Sec. IV. Three different machine learning methods (Random Forest, Deep Learning, and Ridge Regression) of diverse power and complexity will be applied to rate the difficulty of the learning problem and identify the most lightweight solution for this task. Furthermore, the model synthesis will be performed for different feature subsets, according to the typically available indicators in the addressed applications. In this way, we also provide a trade-off between model complexity and accuracy, with respect to the demands and constraints that are given by the particular application scenarios.

A. Relation of Downlink Indicators to Uplink Power Control

According to the LTE standard [14], the UE calculates its uplink transmission power P_{tx} according to

$$P_{tx} = \min \left(\begin{array}{l} P_{max}, \\ P_0 + 10 \log_{10}(M) + \alpha \cdot PL + \Delta_{MCS} + \delta \end{array} \right). \tag{1}$$

First of all, it includes a compensation of the estimated path loss PL, which is weighted by a pre-configured Fractional Path Loss Compensation (FPC) factor α. P_0 represents the requested Signal to Interference and Noise Ratio (SINR) per RB for the reception by the evolved NodeB (eNodeB). Together with α, these values are configured by the operator with respect to the environment and the building density. Since the target power is always related to the SINR per single Physical Resource Block (PRB), the total output power must be adjusted by the actual number of transmitting RBs M and the particularly increased SINR of Δ_{MCS} for higher Modulation and Coding Schemes (MCSs). Finally, the Transmission Power Control (TPC) formula includes a closed-loop component δ. It reflects cumulated instructions of the eNodeB to the UE to slightly increase or decrease its TX-power according to the actually received signal strength and signal quality at the eNodeB. The maximum output power of the UE, however, is limited to P_{max}, which corresponds to 23 dBm for class 3 UE.

Although the latter is known to be a constant, the remaining parameters are either system- or context-dependent and are not accessible from UE's application-layer. However, they either have direct or indirect influences on accessible indicators or at least can also be assumed or approximated as constants. The path loss PL may be tightly related to the Reference Signal Received Power (RSRP) and Received Signal Strength Indicator (RSSI) at the UE. Modulation and coding scheme Δ_{MCS} may additionally correlate to Reference Signal Received Quality (RSRQ). With this relationship and assuming the serving cell to not being congested, the number of assigned RBs M may correlate to the perceived UL data rate at the UE. Furthermore, we assume P_0 to be constant for an operator, or at least for a homogeneous environment and expect δ to average to 0 for a properly configured cell. The same assumption is applied on α: Due to its function, it is expected to be particularly constant for urban, suburban and rural environments. It may be approximated by the currently used frequency band by the UE and the number of visible neighbor cells.

B. Machine Learning Methods and Model Synthesis

We consider three machine learning techniques to learn prediction models for the TX-power. The methods differ w.r.t. their complexity. On the one hand, complex models exhibit a high capacity which allows us to learn harder functions, resulting in a lower prediction error. This implies, however, that learning an actual parametric representation of the model is harder too, e.g., training a complex model requires more computational resources. On the other hand, computing the prediction with a complex model is computationally demanding and its parametric representation requires more memory. Let $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$ be a data set of size N. Here, $\boldsymbol{x} \in \mathbb{R}^d$ is a *feature vector* and $y \in \mathbb{R}$ is the *label* that we want to predict, i.e., our goal is to estimate a function f such that $f(\boldsymbol{x}) \approx y$.

a) Ridge Regression: The ridge regression (RR) [15] is an l_2-regularized linear regression model with $f(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$. Learning RR models is done by minimizing the following regularized least-squares loss:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \lambda \|\boldsymbol{\beta}\|_2^2 + \underbrace{\frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - f(\boldsymbol{x}^{(i)}))^2}_{\text{Mean squared error (MSE)}} . \qquad (2)$$

The first term allows us to control the complexity—it penalizes models with overly large parameter values. To see this, observe that for $\lambda \to \infty$, the zero-vector will be optimal. In the other extreme ($\lambda = 0$), the minimization problem is unregularized and we retrieve the model with minimal training error. By choosing a moderately small λ, we disallow the model to adapt itself to noise or other numerical particularities of the training data which results in better generalization performance—the model's prediction will be more accurate on unseen data [16]. Linear models are particularly simple in that the number of parameters is equal to the number of features d. Due to strict convexity, the learning problem has a unique global minimizer.

b) Random Forest: Random forests (RF) [17] are a bootstrap [18] of decision trees. Suppose that $T = (V, E)$ is a directed, tree structured graph. Each vertex $v \in V$ has (at most) two children v_{left} and v_{right}. Moreover, for each vertex v, $\text{val}(v) \in \mathbb{R}$ is a real number, $\text{idx}(v) \in \{1, 2, \dots, m\}$ is a feature index, and $\text{childs}(v)$ returns the number of children. The vertex function $f(v, \boldsymbol{x})$ is then

$$f(v, \boldsymbol{x}) = \begin{cases} f(v_{\text{left}}, \boldsymbol{x}) & , \boldsymbol{x}_{\text{idx}(v)} \leq \text{val}(v) \land \text{childs}(v) = 2 \\ f(v_{\text{right}}, \boldsymbol{x}) & , \boldsymbol{x}_{\text{idx}(v)} > \text{val}(v) \land \text{childs}(v) = 2 \\ \text{val}(v) & , \text{else} \ . \end{cases} \qquad (3)$$

The prediction function of a single decision tree T with root v_0 can then be written as $T(\boldsymbol{x}) = f(v_0, \boldsymbol{x})$. Solving the corresponding learning problem is likely to have exponential runtime complexity. Thus, practical decision trees are grown heuristically by sequentially choosing $\text{idx}(v)$ and $\text{val}(v)$ such that $\sum_{i=1}^{N} (y^{(i)} - T(\boldsymbol{x}^{(i)}))^2$ is minimized. Finally, a random forest is a set of trees $\mathcal{T} = \{T_1, T_2, \dots\}$, where each tree T_i is grown on a random bootstrap sample $\mathcal{D}_i$ of the training data. The prediction of the forest is then the average ol all tree predictions: $f(\boldsymbol{x}) = (1/M) \sum_{T \in \mathcal{T}} T(\boldsymbol{x})$. Assuming that D is the depth of the deepest tree, the forest has a worst-case storage complexity of $\mathcal{O}(M 2^D - 1)$.

c) Deep Learning: Deep learning (DL) [19] is the machine learning technique which currently gets the highest attention from the research community and beyond. While classic (non-deep) approaches rely on hand-crafted features and various hyperparameters, deep learning methods aim at phrasing almost all parts of the model as differentiable function. Thus, numerical optimization methods can replace what was formerly done by hand. These methods work especially well in computer vision tasks, where a large number of semantically equivalent features is present, e.g., the pixel colors of an image. Compared to this setting, our number of features is rather small. However, we include this technique since it may discover unknown high-level features from our base features. More precisely, we use feed-forward neural networks with dropout [20]. Dropout is a regularization technique which randomly prohibits the update of some model parameters, thus, preventing the model from overfitting the training data. In contrast to linear models, the objective function is non-convex and the learning may get stuck in saddle points or weak local optima.

IV. Setup of the Empirical Evaluation

This section describes the process of model generation, which includes the acquisition of training data from public

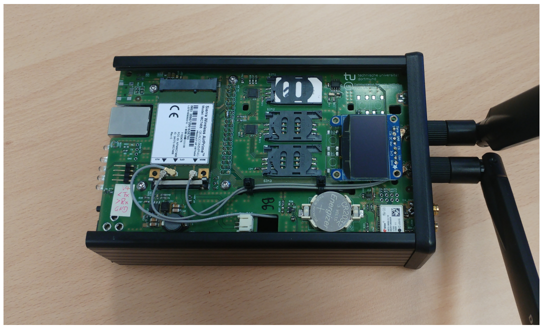

Fig. 2. Photo of the embedded V2X platform used for the measurements.

TABLE I
CAPTURED FEATURES AND ASSOCIATION TO APPLICATION-SPECIFIC PREDICTION MODELS BASED ON FULL-FEATURE SET $\mathbb{F}$, PRACTICAL SETS $\mathbb{P}1$/$\mathbb{P}2$, AND SIMULATION SET $\mathbb{S}$.

Parameter	Model	Indicated Influences(s)
Velocity	$\mathbb{F}$,$\mathbb{P}1$,$\mathbb{P}2$,$\mathbb{S}$	Distortions by fast fading
Upload size	$\mathbb{F}$,$\mathbb{P}1$,$\mathbb{P}2$,$\mathbb{S}$	Influence of TCP slow start
RSRP	$\mathbb{F}$,$\mathbb{P}1$,$\mathbb{P}2$,$\mathbb{S}$	Signal strength, distance
RSRQ, SINR	$\mathbb{F}$,$\mathbb{P}1$,$\mathbb{P}2$	Signal clarity, interference
Datarate	$\mathbb{F}$,$\mathbb{P}1$	Signal strength, allocated RBs M
RSSI	$\mathbb{F}$	Signal strength, distance
Frequency band	$\mathbb{F}$	Environment [9]
Number of neighbor cells (intra/inter freq.)	$\mathbb{F}$	Environment, cell density, interference
Cell bandwidth	$\mathbb{F}$	Exhaustion of TX-power headroom

networks and the subsequent machine learning procedure.

A. Data Acquisition

In order to gather the necessary training data for model generation, mobile measurements of a public cellular network in Germany were performed. They were taken by a developed embedded Vehicle-to-everything (V2X) platform, shown in Fig. 2, which was mounted in the rear trunk of a car. This battery-powered system is based on an ARM-processor and includes a Sierra Wireless MC7455 LTE-Advanced (LTE-A) modem for network probing. In contrast to COTS smartphones, this modem exposes its momentary TX-power as well as further detailed information about its current state and the attached cellular network. This includes a set of passive connectivity indicators, which are listed in Tab. I.

The platform was instructed by our software (source code is available in [21]) to periodically upload a file via Hypertext Transfer Protocol (HTTP) to a web-server in intervals of 30 s. During each transmission and in intervals of 1 s, the software sampled 31 parameters in total from the modem and the involved transmission process (e.g., data rate, file size). In addition, the campaign was repeated with different upload file sizes of 1 MB, 3 MB, and 5 MB. The lower bound of the interval is motivated by the required query time for polling the TX-power together with the other parameters from the modem. Although multiple values are queried in groups, single queries may take up to 250 ms depending on the modem's utilization. Short transmissions, however, might be finished between two queries of TX-power and result in a useless sample due to no upload activity at query time.

The upper bound, however, is motivated by the contract to conserve mobile traffic quota in favor of sampling over a large spatiotemporal domain but still performing measurements beyond the slow-start phase of a TCP transmission. We discuss the relevance of the slow start in Sec. V.

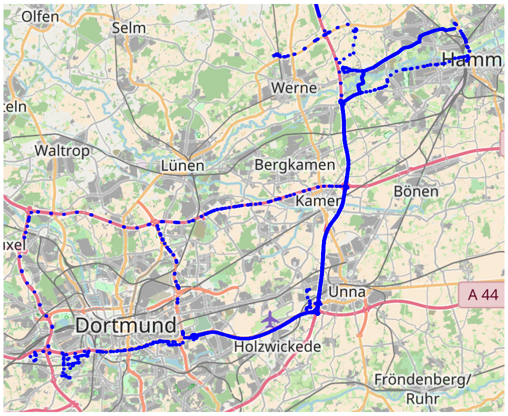

Fig. 3. Road map with locations of all data samples of the measurement campaign between two larger cities in Germany. Each blue point represents an intermediate status logging of all measured variables (cf. Tab. I) during ongoing UL transmissions. (Map: ©OpenStreetMap contributors, CC BY-SA)

The road map in Fig. 3 shows distinct measurement points along a trajectory through urban, suburban and rural environments, and hence provides data for an environment-independent model. The main route has a length of 44 km and stands out as dense line of trace points. However, the data set also includes differing routes to avoid an over-fitting to the main path. In total, 6172 samples were collected during this campaign.

In order to validate that the captured TX-power levels are representative for the environment, we plotted the Empirical Cumulative Distribution Function (ECDF) of our measurements into Fig. 4. For comparison, we also included three distributions from [9], which reflect the TX-power distributions of different environments in Sweden. That data was obtained from a large set of base stations with the support of a mobile network operator [9]. Our measurements match the distribution for urban environments from the reference set, although our the trajectory also includes suburban and rural areas. However, the leading network operators in Germany typically expand their networks along motor highways, which explains the similarities to urban environments.

Note that the fork at 0 dBm is not issued by the environment but by post-processing the data: Since the modem also reports a TX-power of 0 dBm in case of no activity, the dataset was filtered for those values to avoid ambiguities. Furthermore, the modem's maximum TX-power was 22 dBm, which is 1 dB below the general limit of 23 dBm. The power levels were validated in our laboratory by a Rohde&Schwarz CMW500 radio communication tester.

B. Application Scenarios and Involved Feature Subsets

In order to provide tailored prediction models for online applications and offline simulations, we identified four different feature subsets $\mathbb{F}$, $\mathbb{P}1$, $\mathbb{P}2$, $\mathbb{S}$, which correspond to the particularly available indicators. The **full-feature set** $\mathbb{F}$ includes all listed parameters from Tab. I and hereby covers all influences, that were outlined in Sec. III-A. It serves as a reference to the subsequent subsets, since in practice the

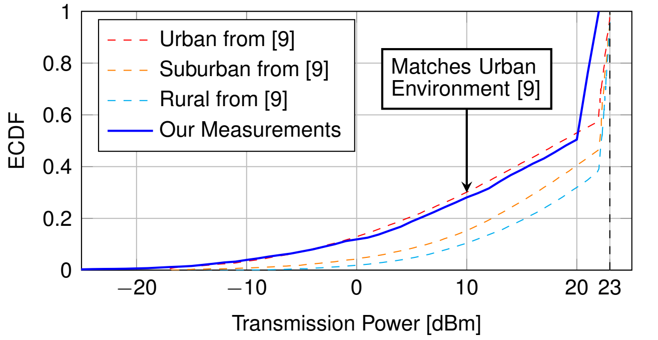

Fig. 4. ECDF of the transmission power from our measurement campaign (blue) in Germany. For comparison, the dashed lines show the results from [9], which were obtained from a mobile network operator in Sweden.

involved features are rarely available at application layer or require vendor-specific commands for access.

For **practical online-applications**, we defined feature subsets $\mathbb{P}1$ and $\mathbb{P}2$, which have been labeled accordingly in Tab. I. These are parameters, which are typically accessible on COTS UE, like smartphones or USB dongles. Both sets only differ in the single parameter *data rate*, which is only included in $\mathbb{P}2$. Although the data rate holds valuable implicit information about the number of allocated RBs and consequently indicates the necessary additional TX-power, it is only available during an active transmission. Therefore, a derived model, which involves this particular parameter, is only applicable for retrospective estimations of current and previous transmissions. It is not suitable for predictions in idle mode. This case, however, will be covered by the model, which is based on subset $\mathbb{P}1$.

Since established simulators like SimuLTE and LTE-Sim provide even fewer indicators, a minimum **subset for simulations** $\mathbb{S}$ includes only three basic parameters *velocity*, *upload size*, and *RSRP*.

C. Model Generation

For our experiments, we configured the models from Section IV-C as follows:

Random forests are trained with $|\mathcal{T}| = 64$ trees, each having maximal depth 32. The model is saturated w.r.t. the number of trees, that is, we drove $|\mathcal{T}|$ up to 128 where all forests with $|\mathcal{T}| > 64$ give essentially the same results. Restricting the depth acts as a regularization term, hence preventing the model to memorize the complete training set and enhancing the generalization performance on previously unseen data. Representing each of the m nodes by 4 values (left child, right child, threshold value, feature index), our forest can be represented by $d_{\mathrm{RF}} = m \times 4 = 1044992$ values.

Ridge regression models are trained with $\lambda = 10^{-3}$. We conducted a parameter grid search with $\lambda = 10^{-i}$ for $i \in \{0, 1, 2, 3, 4, 5\}$, where all choices resulted in a reasonable generalization performance. The number of model parameters is $d_{\mathrm{RR}} = 12$ (one per feature plus one bias term).

Our deep learning model consists of three fully connected hidden layers, each with 64 hidden nodes and non-linearity via rectified linear units (ReLU). We use an increasing dropout rate per layer which is motivated by various empirical and theoretical insights [22]. More precisely, $\mathbb{P}_{\mathrm{Dropout}} = 0.1 \times i$ on layer $i \in \{1, 2, 3\}$. The parameters are estimated by stochastic gradient descent with an adaptive learning rate and standard hyperparameters ($\epsilon = 10^{-8}, \rho = 0.99$). The total number of learnable parameters is $d_{\mathrm{DL}} = 12 \times 64 + 64^2 + 64^2 + 64 = 9024$.

All models are learned with RapidMiner 8.1 and the corresponding process files can be found online [21].

V. Numerical Results

We start by assessing the importance of all features w.r.t. the TX-power. To this end, we binned all values into 10 intervals $b_1, b_2, \ldots, b_{10}$ of equal width and computed the mutual information (MI) between each feature and the TX-power, $\mathrm{MI}(X,Y) = \sum_{i=1}^{10} \sum_{j=1}^{10} p_{i,j} \log(p_{i,j}/p_i p_j)$. Here, $Y = \text{TX-power}$, X is any of the $\mathbb{F}$ features, $p_i = \hat{p}(X = b_i)$ is the relative number of data points in which feature X is in bin b_i and $p_{i,j} = \hat{p}(X = b_i, \text{TX-power} = t_j)$ is the number of cases in which $X = b_i$ and $\text{TX-power} = t_j$ simultaneously. The MI measure how dependent two variables are on each other. In contrast to plain correlation, MI is able to capture non-linear dependence.

The MI values in descending order are RSRP 0.589, RSSI 0.543, num. of intra freq. neighbor cells 0.217, SINR 0.212, RSRQ 0.153, data rate 0.090, velocity 0.076, upload size 0.041, num. of inter freq. neighbor cells 0.013, frequency band 0.008, and cell bandwidth 0.007. The most important parameters are RSRP and RSSI, which imply the signal path loss of the downlink signal and particularly provides an estimate of the path loss in the uplink direction. However, these two indicators carry redundant information, since the models based on smaller feature sets without RSSI still provide very accurate results. The same applies to the indicator group of RSRQ, SINR, and number of neighbor cells (intra freq.), which imply the signal quality, however with a smaller importance for the full model. Frequency band, cell bandwidth, and inter-frequency neighbors have only negligible relevance for the resulting model.

We saw that the most important feature is contained in all feature seats under consideration. It is hence reasonable to expect that all feature subsets contain enough information for a decent predictive quality. Now, we provide results regarding the prediction error of our machine learning models, an analysis of side effects, and the cumulation of errors. All results are 10-fold cross-validated [16], i.e., our data set is partitioned into 10 sets. Each model is trained 10 times, where in each run, the model is trained on 9 sets and tested on the remaining set. The prediction errors of these 10 runs are then averaged. Cross-validated results are more reliable than results gathered from a single train/test split, since a single split might be strongly in favor or strongly to the disadvantage of a method. By averaging over multiple runs, such artifacts are avoided.

Results in terms of prediction error (in [dB]) of the trained models are provided in Fig. 5. The left plot shows the Root Mean Square Error (RMSE) (the square root of the MSE from equation 2) for all three models $f \in \{\mathrm{DL}, \mathrm{RF}, \mathrm{RR}\}$. On the right, we present the Mean Absolute Error (MAE) (right side) which is defined as $(1/N) \sum_{i=1}^{N} |y^{(i)} - f(\boldsymbol{x}^{(i)})|$. The RMSE weights large prediction errors heavier and tiny prediction errors slightly, while the MAE treats all errors equally. Error bars are displaying the standard deviation computed over the 10 cross-validation runs.

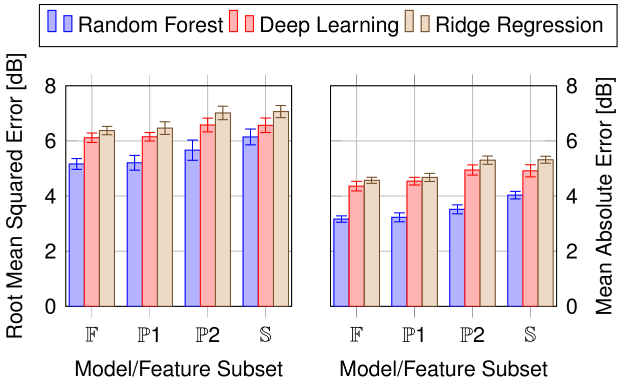

Fig. 5. Cross-validated error of trained prediction models for each feature subset (𝔽, ℙ1, ℙ2, 𝕊) and each machine learning method (Random Forest, Deep Learning, Ridge Regression) in terms of RMSE (left) and MAE (right). Lower is better.

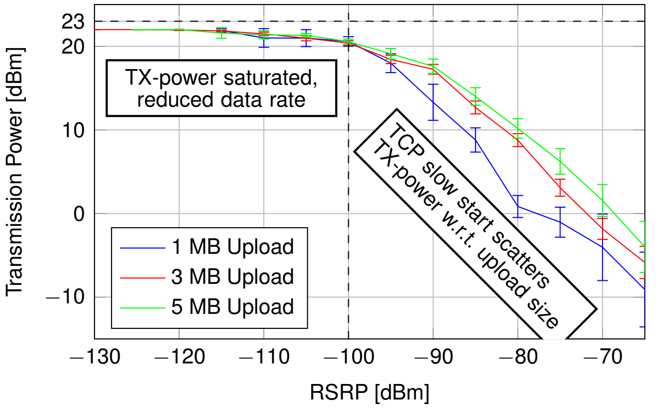

Fig. 6. Relationship between RSRP and TX-power grouped by different upload sizes. Larger uploads lead to higher TX-power levels, as the TCP slow start algorithm reaches higher data rates and involves the transmission of more RBs in parallel. The effect decreases as TX-power gets saturated.

In any case, the deviations are below 0.4 dB with maximum values of 0.368 dB and 0.220 dB for RMSE and MAE, respectively. This indicates a good model fit to unknown and independent data, which are not included in the training set. According to the results, Random Forest performs best with an MAE of 3.166 dB in the full feature subset 𝔽. However, even in the most compact feature subset for simulations 𝕊, the error raises only moderately by less than 1 dB to 4.033 dB. Therefore, 𝕊-models have only a slightly increased error compared to the other models, but rely only on the three input features velocity, upload size, and RSRP. This makes it a valuable option for system-level simulations, which model mobile networks with a very coarse-grained amount of detail.

Comparing the different machine learning methods, Random Forest outperforms the other methods on all feature subsets. However, all results have the same order of magnitude, which suggest that the residual errors are not an artifact of the utilized machine learning method, but a consequence of the uncertainty induced by an incomplete view of all influencing factors. On the other hand, as Ridge Regression produces the largest errors in this setup, a non-linear relationship of input features and resulting TX-power is suggested.

This is also visible in Fig. 6, which depicts the (single-feature) relationship between RSRP and resulting TX-power for different upload sizes. The figure shows the averaged TX-power for bins of RSRP values with a width of 5 dB. The bins b_r are aligned to RSRP values r, where $(r \mod 5) = 0$ and include values in the interval $[r, r + 5)$. Error bars are showing the 0.95 confidence intervals (CIs) of the observed samples. Two areas are visible, which differ in the extent of CI and the impact of data size on the average TX-power. They can be coarsely separated by an RSRP value of −100 dBm. Samples below that threshold, which generally indicate low or poor radio conditions, show very tight CIs and congruent TX-power levels. For larger RSRP values, however, the TX-power is affected by an increased degree of scattering and the mean value becomes data-size dependent. As pointed out in Sec. III and Eq. 1, the TX-power depends on the number of assigned RBs M. In spite of its direct relationship to the throughput, the latter is also affected by Δ_{MCS} and buffering by the operating system. Therefore, the data rate is only reflecting the average resource utilization on physical layer but not the instant situation at subframe level. Given a fixed Δ_{MCS} in Eq. 1, the uncertainty of $M = 1 \ldots 100$ leads to a possible scattering of P_{tx} in the range of 20 dB for a 20 MHz cell.

A possible explanation for the dependency of the data size is the TCP slow start mechanism. In this case, the transmitter slowly increases its data rate until the maximum data rate is reached. By the same token, the amount of assigned RBs M also slowly increases as long as spare resources are available. Consequently, small payloads, e.g., 1 MB, rarely reach full RB utilization, because the transfer is already finished in the slow start region. As the data size increases, the average throughput also grows towards a full RB allocation, and pushes the average transmission power to a higher level. However, in case of poor radio conditions (left side of Fig. 6), the number of RBs is quickly saturated by the maximum TX-power of the UE, which limits the maximum data rate to a much smaller value. Therefore, the slow start quickly reaches the maximum data rate even in case of small payloads.

Besides the accuracy of single predictions, the cumulative behavior of the prediction error has a large relevance for long-term applications. Therefore, we designed a numerical simulation to estimate the absolute sum of errors in relation to the number of executed predictions Fig. 7. The points on the curve of each method are generated as follows: For each

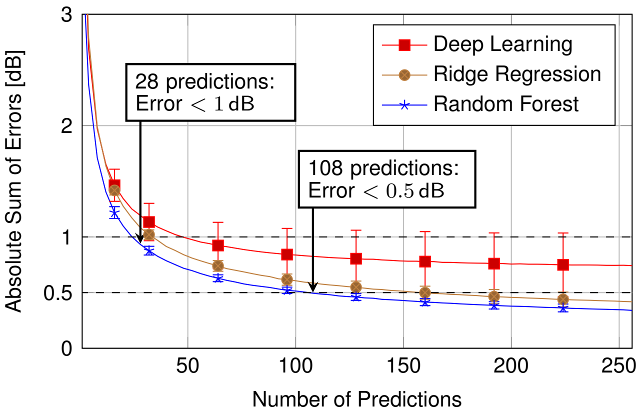

Fig. 7. Cumulated absolute error of the proposed prediction models in relation to the number of executed predictions. In average the Random-Forest model undershoots the error mark of 1 dB after 28 predictions. Therefore, in long-term applications over- and underestimations eliminate each other and provide an accurate estimate for the average TX-power.

combination of cross-validation run and "Number of predictions" l (the abscissae), we draw l random indices $i_1, i_2, \ldots, i_l$ and compute the cumulative error $e_j^l = (1/l)\sum_{i=1}^{l} y_{i_l} - \hat{y}_{i_l}$, where y is the true value and $\hat{y}$ the prediction. This is repeated $\mathcal{M} = 10^4$ times and averaged $\mathcal{E}_l = (1/\mathcal{M})\sum_{i=1}^{\mathcal{M}} |e_j^l|$ where $|\cdot|$ denotes the absolute value. Taking the absolute value is required to prevent that errors from the $\mathcal{M}$ independent runs cancel each other out. Thus, the value $\mathcal{E}_l$ is an estimate to the expected absolute error that we see, if we cumulate the pointwise errors of l predictions. It is necessary to average the result of this computation on each test set of the cross-validation to avoid overly pessimistic or overly confident estimators, i.e., the final ordinate for the abscissa l is $E_l^* = (1/10)\sum_{k=1}^{10} E_l^k$ where E_l^k is computed on the k-th cross-validation run. The estimation is carried out for each method and each $l \in \{1, 2, 4, 8, 12, \ldots, 256\}$ to yield the curves shown in Fig. 7. The same procedure is used to estimate the standard deviation of the cumulated errors of l predictions, which are shown in the plot via error bars.

With Random Forest as an example, the cumulated absolute error undershoots in average after 28 predictions an error of 1 dB, after 108 predictions it even falls below 0.5 dB. As the deviation is very small, the proposed approach is suitable for long-term applications, since the cumulated prediction error does not diverge over time.

VI. Conclusion

In order to rate the power consumption of UEs during uplink transmissions in mobile networks, knowledge about the instant TX-power is required. As this information is not accessible on most mobile devices and in many system-level simulators, this paper closes the gap by a customizable machine-learning approach. The TX-power is estimated on the basis of passive indicators, such as RSRP, velocity, and data rate, which are generally accessible in the mentioned applications. The required trace data is obtained from excessive drive tests that include periodic uplink transmissions over a public mobile network. Three machine learning methods (Random Forrest, Deep Learning, and Ridge Regression) were applied on four different feature subsets of the obtained data to provide the most accurate and lightweight predictor according to the amount of available indicators on the target platform. The full-featured model provides predictions with an MAE of 3.166 dB by using Random Forests. However, even when limiting the input features to velocity, upload size, and RSRP, the MAE only raises to 4.033 dB. For long-term applications the model maintains stability, since over- and underestimations eliminate each other and the cumulated error sum converges towards 0.

In future work, we will further investigate the impact of the TCP congestion control mechanism and include measurements of unbuffered User Datagram Protocol (UDP) transmissions into the dataset. Furthermore, we will integrate the prediction models into CoPoMo for online estimations of the momentary power consumption.

Acknowledgment

Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", projects A1, A4, and B4.

References

[1] G. Miao, "Energy-efficient uplink multi-user mimo," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2302–2313, May 2013.

[2] B. Sliwa, T. Liebig, R. Falkenberg, J. Pillmann, and C. Wietfeld, "Efficient machine-type communication using multi-metric context-awareness for cars used as mobile sensors in upcoming 5g networks," in *IEEE Vehicular Technology Conference (VTC-Spring)*, Porto, Portugal, Jun. 2018.

[3] ——, "Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5G networks," in *2018 IEEE 88th IEEE Vehicular Technology Conference (VTC-Fall)*, Chicago, USA, Aug. 2018, accepted for presentation. [Online]. Available: https://arxiv.org/abs/1805.06603

[4] A. Virdis, G. Stea, and G. Nardini, *Simulating LTE/LTE-Advanced networks with SimuLTE*. Cham: Springer International Publishing, 2015, pp. 83–105.

[5] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 498–513, 2011.

[6] B. Dusza, C. Ide, L. Cheng, and C. Wietfeld, "CoPoMo: A context-aware power consumption model for LTE user equipment," *Transactions on Emerging Telecommunications Technologies (ETT), Wiley*, vol. 24, no. 6, pp. 615–632, 2013.

[7] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 225–238.

[8] A. R. Jensen, M. Lauridsen, P. Mogensen, T. B. Sørensen, and P. Jensen, "LTE UE power consumption model: For system level energy and performance optimization," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, Sep. 2012, pp. 1–5.

[9] P. Joshi, D. Colombi, B. Thors, L. E. Larsson, and C. Törnevik, "Output power levels of 4G user equipment and implications on realistic RF EMF exposure assessments," *IEEE Access*, vol. 5, pp. 4545–4550, 2017.

[10] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An open-source platform for LTE evolution and experimentation," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, ser. WiNTECH '16. New York, NY, USA: ACM, Oct. 2016, pp. 25–32.

[11] R. Falkenberg, K. Heimann, and C. Wietfeld, "Discover your competition in LTE: Client-based passive data rate prediction by machine learning," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec. 2017, pp. 1–7.

[12] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[13] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1790–1821, thirdquarter 2017.

[14] *3GPP TS 36.213 - Physical layer procedures (Release 13)*, 3rd Generation Partnership Project Technical Specification, Rev. V13.0.1, Jan. 2016. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/36_series/36.213/

[15] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.

[16] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*, ser. Springer series in statistics. Springer, 2009.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[18] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, Jan. 1979.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] R. Falkenberg, "Machine learning based transmission power prediction: Capturing software, raw data, and exported models," Mar. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1208603

[22] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in Neural Information Processing Systems 30*, Dec. 2017, pp. 3584–3593.