On the Potential of Ensemble Regression Techniques for Future Mobile Network Planning

Jessica Moysen, Lorenza Giupponi, Josep Mangues-Bafalluy
Centre Tecnològic de Telecomunicacions de Catalunya-CTTC
Av. Carl Friedrich Gauss 7, 08860 Castelldefels (Spain)
$\{jessica.moysen, lorenza.giupponi, josep.mangues\}@cttc.es$

Abstract—**Planning of current and future mobile networks is becoming increasingly complex due to the heterogeneity of deployments, which feature not only macrocells, but also an underlying layer of small cells whose deployment is not fully under the control of the operator. In this paper, we focus on selecting the most appropriate Quality of Service (QoS) prediction techniques for assisting network operators in planning future dense deployments. We propose to use machine learning as a tool to extract the relevant information from the huge amount of data generated in current 4G and future 5G networks during normal operation, which is then used to appropriately plan networks. In particular, we focus on radio measurements to develop correlative statistical models with the purpose of improving QoS-based network planning. In this direction, we combine multiple learners by building ensemble methods and use them to do regression in a reduced space rather than in the original one. We then compare the QoS prediction accuracy of various approaches that take as input the 3GPP Minimization of Drive Tests (MDT) measurements collected throughout a heterogeneous network and analyse their trade-offs. We also explain how the collected data is processed and used to predict QoS expressed in terms of Physical Resource Block (PRB)/ Megabit (MB) transmitted. This metric was selected because of the interest it may have for operators in planning, since it relates lower layer resources with their impact in terms of QoS up in the protocol stack, hence closer to the end-user.**

Index Terms—**Machine Learning, Big Data, Quality of Service, Prediction, Network planning, Minimization of Drive Tests**

I. Introduction

Current 4G networks are generating a huge amount of data during their normal operation in the form of control, management and data measurements. This data is expected to increase in 5G due to different aspects, such as densification, heterogeneity in layers and technologies, additional control and management complexity in Network Functions Virtualisation (NFV) and Software Defined Network (SDN), advent of the Internet of Things (IoT), increasing variety of applications and services, each with distinct traffic patterns and QoS/Quality of Experience (QoE), etc. Only small amount of this data is currently stored, and a lot of valuable information is actually discarded after usage [1]. Therefore, there is a need for a more systematic approach for extracting relevant information out of this wealth of operational data for the benefit of the operator, and eventually, the end-user.

In this context, mobile operators face multiple Network Management (NM) challenges for future 5G networks, such as: a) Managing future network complexity in terms of densification of scenarios, heterogeneous nodes, applications, Radio Access Technologiess (RATs); b) Managing the dynamicity of networks, where some femto nodes are controlled by users, energy saving approaches are in place, and active antennas play a crucial role; c) Improving QoS by increasing link rates and reducing latency; d) Managing virtualized infrastructures based on the SDN/NFV paradigms and cloudification. The high level objective of an operator is to build networks, which are self-aware, self-adaptive, and intelligent. In this context, Machine Learning (ML) can be used as a tool to allow the network to learn from experience, improving performances, and big data analytics can drive the network from reactive to predictive. The exploitation of past information is highly relevant when planning new deployments, but has hitherto been hard to achieve.

In this paper, we focus on designing a tool for QoS prediction, able to assist network operators in smartly planning future dense deployments. The objective is to predict QoS in a given area, given a certain deployment where the interference patterns are extremely variable due to the very high frequency reuse. 3GPP already provides an interesting data base to collect useful data for the purpose of QoS estimation. The MDT feature has been introduced by 3rd Generation Partnership Project (3GPP) since Release 10. Among the targets, there are the standardization of solutions for coverage optimization, mobility, capacity optimization, parametrization of common channels, and QoS verification [2]. Since operators are also interested in estimating QoS performance, in Release 11, the MDT functionality has been enhanced to properly dimension and plan the network by collecting measurements indicating throughput and connectivity issues [3].

The problem of QoS prediction, estimation and verification has been studied in the literature in [4], [5]. Here, the authors address the MDT QoS verification use case by identifying and estimating different Key Performance Indicators (KPIs) and correlating them with common node measurements to establish whether the UE is offered an acceptable QoS. However, previous work mainly targeted traditional macrocell scenarios, and so, do not face the challenges of dense HetNets. In our preliminary work [6], we focus on a more complex multi-layer heterogeneous networks, where we predict QoS independently

This work was made possible by NPRP grant No. 5-1047-2-437 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors. The work of J. Moysen is also supported by grant TEC2014-60491-R.

of the physical location of the UE. Preliminary results show that by abstracting from the physical position of the measurements, and finding patterns in PHY layer measurements collected from different regions of the network, we can provide better estimations of QoS in other arbitrary regions. Furthermore, previous work suggests that regression analysis has a better performance in a reduced space by considering distance-based dimensionality reduction, i.e., we observe that the best result is obtained when we consider only the information coming from the serving and the strongest neighboring cells as input features. As a result, in this paper, we exploit this background to focus on improving network planning based on MDT and QoS prediction aiming at minimizing the PRB/MB offered [7]. We select this metric because it presents a series of interesting features for our purpose. First, it combines resources that are relevant for the operator (physical resource blocks) and others that are relevant for the end-user (megabits of data) into a single metric. Second, in terms of network planning, it is also of interest because minimizing this metric would allow serving users with the same QoS by consuming less resources, and so, being more cost-effective. Therefore, we believe this metric allows linking QoS optimization with cost-related network planning done by operators.

We consider 4 algorithms: (1) k-Nearest Neighbours (k-NN), (2) Neural Networks (NN), (3) Support Vector Machines (SVMs), and (4) Decision Trees (DT). In order to improve their accuracy in prediction, we resort to ensemble methods, which are a common tool in SL to improve accuracy. We focus on Bagging and AdaBoost. These methods offer the opportunity to find patterns and relationships between input and output variables, which due to the inherent complexity of communications and interference patterns, in the complex heterogeneous scenarios dense deployment, would not be readily apparent or possible to be captured through analytical approaches. In addition to this, to deal with the huge amount of features describing our input feature search techniques, we propose to apply regression analysis in a reduced space rather than in the original one. We focus on Principal Component Analysis (PCA) and Sparse Principal Component Analysis (SPCA), which are the most representative feature extraction and feature selection algorithms [8]. We perform a study of the regression techniques using the treated data, we analyse their performance, and draw conclusions on their interest for network planning.

As for network planning tools per se, they in general focus on RF coverage planning (e.g., CelPlan [9]) and not directly on QoS offered to end-users and the resources the operator needs to offer it. On the other hand, our techniques allow predicting the PRB/MB offered in an arbitrary point of the network based on measurements collected throughout the network. By integrating the schemes proposed in our paper in a network planning tool, operators would be able to find the most appropriate deployment layout so as to minimize the resources (i.e., the cost) they need to deploy to offer a given QoS in a newly planned deployment.

The paper is organized as follows. Section II introduces the techniques that we use to construct the proposed method. The system model is described in Section III. Section IV describes the data analysis steps followed. Section V presents our QoS prediction results followed by the implications of their potential application to network planning (section VI). Finally, Section VII concludes the paper.

II. Key concepts

Learning is the process of gaining knowledge by instruction or study, and the discovery of new facts by experience. Computer modelling of learning processes that are able to introduce such capabilities in computers is the main challenge of ML. ML studies computer algorithms for learning to complete a task, or to make predictions based on observations, i.e., it is about learning to do better in the future based on what was experienced in the past. ML improves the performance of a particular set of tasks by creating a model that helps find patterns through learning algorithms. For that, the construction of a dataset is needed. The dataset contains training samples (rows), and features (columns), and is divided in 2 sets. The training set to train the model, and the test set to make sure that the predictions are correct. ML is generally roughly classified into: a) Supervised Learning (SL), b) Unsupervised Learning (UL), and c) Reinforcement Learning (RL). In this paper we focus on SL and UL approaches, since the focus is on data analysis.

a) Supervised Learning: SL is a Machine learning technique which takes training data (organized into input and desired output) to develop a predictive model, by inferring a function $f(\mathbf{x})$, returning the predicted output $\hat{y}$. The input space is represented by a n-dimensional input vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T \in \mathbf{R}^n$. Each dimension is an input variable. In addition a training set involves m training samples $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$. Each sample consists of an input vector $\mathbf{x}_i$, and a corresponding output y_i. Hence $x_i^{(j)}$ is the value of the input variable $x^{(j)}$ in training sample i, and the error is usually computed via $|y_i - \hat{y_i}|$. SL techniques can be classified depending on whether they predict discrete or continuous variables into classification and regression techniques, respectively. Since our problem is a regression problem, i.e., y_i is continuous in nature, we select the following regression models:

1) k-NN can be used for classification and regression [10]. The k-NN method has the advantage of being easy to interpret, fast in training, and the amount of parameter tuning is minimal.
2) NN is a statistical learning model inspired by the structure of a human brain where the interconnected nodes represent the neurons to produce appropriate responses. NN support both classification and regression algorithms. NNs methods require parameters or distribution models derived from the data set, and in general they are susceptible to over-fitting [11].
3) SVMs can be used for classification and regression. This method in general shows high accuracy in the

prediction, and it can also behave very well with non-linear problems when using appropriate kernel methods. Also, when we cannot find a good linear separator, kernel techniques are used to project data points into a higher dimensional space where they can become linearly separable. Hence the correct choice of kernel parameters is crucial for obtaining good results [12].

4) DT is a flow-chart model, which supports both classification and regression algorithms. Decision trees do not require any prior knowledge of the data, are robust, and work well on noisy data. However, they are dependent on the coverage of the training data as, as for many classifiers, and they are also susceptible to over-fitting [13], [14].

In order to enhance the performance of each learning algorithm described before, instead of using the same data set to train we can use multiple data sets by building an ensemble method. Ensemble methods are learning models, which combine the opinions of multiple learners. This technique has been investigated in a huge variety of works [15], [16], where the most useful techniques have been found to be Bagging and AdaBoost [17]. Bagging manipulates the training examples to generate multiple hypothesis. It runs the learning algorithm several times, each one with different subset of training samples. AdaBoost works similarly, but it maintains a set of weights over the original training set, and adjusts these weights by increasing the weight of examples that are misclassified, and decreasing the weight of examples that are correctly classified [18].

b) Unsupervised Learning: UL is a ML technique, which receives only inputs **x**, and it lets the computer learn by itself. Data are given without labels and the objective is to find a structure in this. Different schemes are available like clustering, and dimensionality reduction. The goal is to construct a representation of **x** that can be used for predicting future inputs without giving the algorithm the right answer [19]. Since our problem is the huge amount of potential features our system may have as input, in our previous work [6], we suggest that the regression analysis has a better performance in a reduced space. We focus on dimensionality reduction, which is the process of reducing the number of random variables under consideration, and can be divided into Feature Extraction (FE) and Feature Selection (FS) methods. Both methods seek to reduce the number of features in the dataset. FE methods do so by creating new combinations of features (e.g. PCA), which project the data onto a lower dimensional subspace by identifying correlated features in the data distribution. They retain the Principal Components (PCs) with greatest variance and discards all others to preserve maximum information and retain minimal redundancy [8]. Correlation based FS methods include and exclude features present in the data without changing them. For example, SPCA, which extends the classic method of PCA for the reduction of dimensionality of data by adding sparsity constraint on the input features.

In this paper, we exploit UL techniques for dimensionality reduction, whose output is fed into an ensemble method consisting of Bagging/AdaBoost to manipulate the training examples. The SL techniques under evaluation are then applied.

III. System model

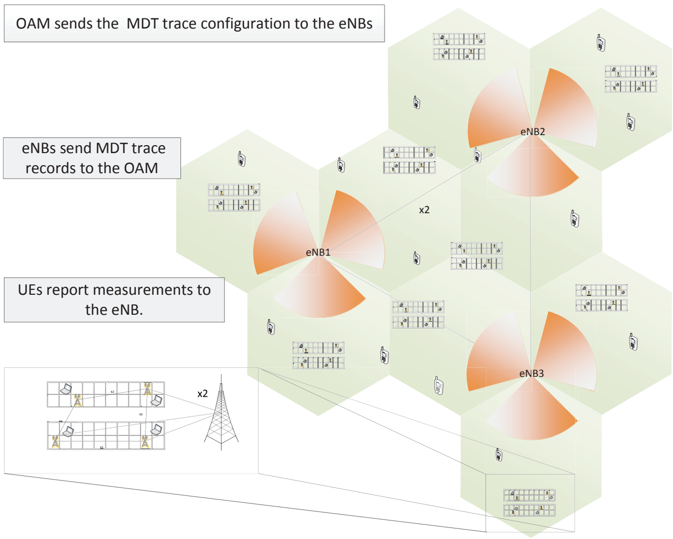
OAM sends the MDT trace configuration to the eNBs
eNBs send MDT trace records to the OAM
UEs report measurements to the eNB.
eNB1
eNB2
eNB3
x2
x2

Fig. 1: Overview of the heterogeneous wireless network.

We consider a heterogeneous wireless network (see Figure 1), whose system performance has been evaluated in the ns3 LTE-EPC Network Simulator (LENA) platform based on Long Term Evolution (LTE) Release 11 [20]. The scenario that we set up consists of 3 Enhanced Node Base stations (eNBs) with three sectors, which results in 9 cells and 19 UEs with transmit power equal to 46 dBm. The small cell network is based on the dual stripe scenario with 1 block of 2 buildings. We consider 30 blocks in the coverage area of the macro cell. Each building has one floor, with 20 apartments, which results in 40 apartments per block. The Home eNodeB (HeNB) deployment ratio is 0.5, and the activation factor is 1, which results in 20 HeNBs, each one located in an independent apartment [21]. The HetNet scenario is given in Table I.

TABLE I: HetNet scenario.

Macrocell scenario	**Value**
eNB Tx Power	46 dBm
Num. of cells	9
Num. of macro UEs	19
Small cell scenario	**Value**
HeNB Tx Power	23 dBm
Num. of Femto blocks	30
Num. of HeNBs per block	20
Num. of home UEs per HeNB	4
Num. of home UEs per block	80
Num. of HeNBs	600
Num. of home UEs	2400

In order to obtain an overview from the scenario described in Table I, we create a Radio Environment Map (REM). Figure 2 shows a $[100 \times 100]$ matrix of $10,000$ values that represent the Reference Signal Received Power (RSRP) with respect to the cell that has the strongest signal at each point. Each point corresponds to one pixel of 10cm by 10cm.

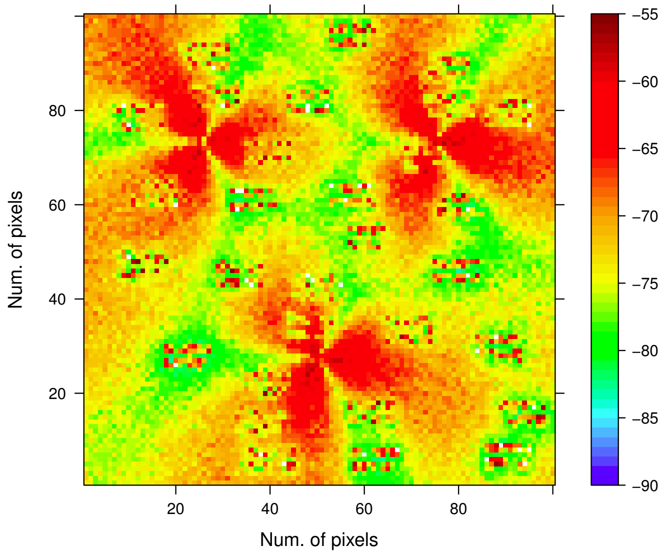

Fig. 2: REM, which represent the RSRP(dBm) with respect to the cell that has the strongest signal at each pixel.

The parameters used in the simulations are given in Table II.

TABLE II: Simulation parameters.

Parameter	Value
PropagationLossModel	HybridBuildings
Scheduler	Round Robin
AMC model	4-QAM, 16-QAM, 64 QAM
Transport protocol	User Datagram Protocol (UDP)
Traffic model	Constant bit rate
Cell layout	radius: 500m
Bandwidth;Num. of RBs	5MHz;25
Simulation time	0.25s

IV. Data analysis procedure

Our approach is divided into three phases. First we collect the data (IV-A), then we prepare them (IV-B), and finally, we analyse them through the proposed regression analysis methods (V). At a high level, the collected data is first fed into a dimensionality reduction step, whose output is then fed into an ensemble method that manipulates the training examples to generate multiple hypotheses by applying Bagging/AdaBoost. The learning algorithm (one of the four regression models explained in section II) is then called to produce a regressor. Finally, in order to evaluate the accuracy of the predictions, the performance of the learned function is measured on the test set.

A. Collecting the data

We collect for each UE: (1) the RSRP, and (2) the Reference Signal Received Quality (RSRQ) coming from the serving and neighbouring HeNBs. The size of the input space is $[l \times n]$. The number of rows is the number l of UEs in the scenario, and the number of columns corresponds to the number of measurements n, i.e., the 600 RSRPs and the 600 RSRQs coming from the serving and neighbouring HeNBs. The size of the output space is $[l \times 1]$, which corresponds to the QoS performance associated to the measurements in terms of the PRB/MB transmitted. For the evaluation of the QoS performance, the PRB/MB transmitted is considered as QoS indicator, because by minimizing this metric, users with the same QoS would be served. As a result, operators would reduce costs. This is why we select this metric, given that we study the interest of using the ML techniques under evaluation in a network planning context.

B. Preparing the data

Once data are collected, we proceed with the data preparation.

1) In order to improve the accuracy of each learning algorithm, namely k-NN, NN, SVM, and DT, we manipulate the training examples to generate multiple hypothesis following Bagging and AdaBoost methods. In each iteration, the ensemble method (Bagging or AdaBoost) draws a training set of size m. The base learning algorithm is then called to produce a predictor.
2) For each test value, we predict the PRB/MB transmitted, and evaluate performance against the actual value in terms of the Root Mean Squared Error (RMSE) as follows:

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

where p is the length of the test set, $\hat{y}_i$, indicates the predicted value, and y_i is the testing value of one data point i. In order to compare the RMSE with different scales, the input and output variable values are normalized by,

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}$$

where y_{max} and y_{min} represent the maximum and minimum values in the output set.
3) We reduce the high dimensional space to a space of fewer dimensions by implementing 2 approaches: 1) the FE-PCA, and 2) the FS-SPCA. The kinds and amounts of measurements that each approach takes into account are described as follows:
   a) FE-PCA. The input features are selected as a result of the PCA implementation. Once the algorithm has identified correlated features in the data, we retain c PCs with greatest variance and discard all others to retain minimal redundancy. We analyse Figure 3, which shows the cumulative contribution of each PC to the original data's variance, namely $\sigma^2 = \sum_{i=1}^{n} \sigma_i^2$, where σ_i^2 corresponds to the variance of the $i-$th principal component. Notice that, the PC1 accounts for the greatest possible

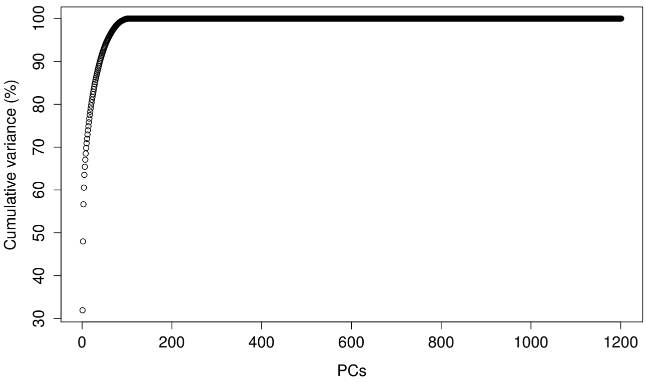

Fig. 3: Importance of components.

variance in the data set, the second one (PC2) accounts for the next highest variance, and so on. We observe that we can obtain more than 90% of cumulative variance if we consider only the first 100 PCs. However, in order to know how many PCs to retain, we focus on Figure 4. This Figure

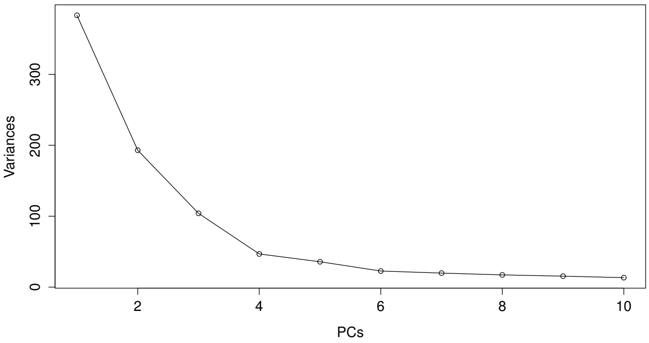

Fig. 4: Variability in the data.

shows the variability of the data set as a function of the $c = 10$ PCs. That is, the figure shows the variances (y-axis) associated with the PCs (x-axis). We can see, that with the first c PCs we already capture the main variability of the data. Since the PCA's goal is to extract as much variance with the fewest PCs, for further analysis we retain only the first 10 PCs.

b) FS-SPCA. The number of input features corresponds to the output generated by the SPCA implementation, in which we have promoted sparsity up to the selection of the q features that give us the most useful information. That is, by adding sparsity constraint on the input features, we promote solutions in which only a small number of input features capture most of the variance.
The number of features is obtained by adjusting the weights over the training examples, i.e., as we increase the weight of SPCA, the number of features is reduced. Figure 5, shows the NRMSE as a function of different number of features selected by the SPCA. It can be observed that for the number of features $q = 92$ we obtain an optimum in terms of NRMSE. As a consequence, we promote sparsity down to 92 features.

c) Distance based Dimensional Reduction (DR). As a benchmark we select a distance based approach where we reduce the dimensionality of the input space, based on the most promising results obtained in [6], where we select only the following features:
    i) $signals_{sc}$: RSRP and RSRQ from the serving HeNB,
    ii) $signals_{st}$: RSRP and RSRQ from the strongest neighbouring HeNB,
    iii) $signals_{2st}$: RSRP and RSRQ from the two strongest neighbouring HeNBs,

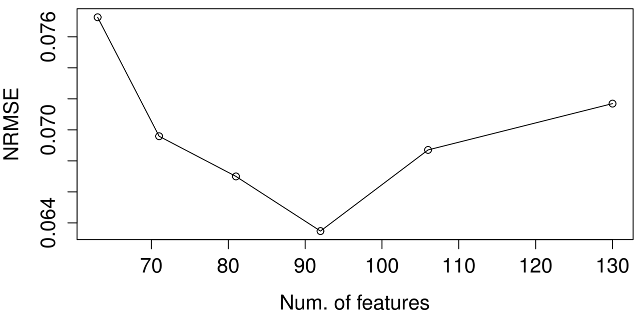

Fig. 5: NRMSE as a function of the number of features

V. Results

In this section, we analyse the performance results of the 4 regression approaches described in Section II. In order to build an effective learning algorithm, the learning parameters given in Table III have been chosen based on experience. In particular, for each regression model we applied grid search algorithm to perform hyperparameter optimization [22]. Furthermore, we benchmark the FE-PCA, FS-SPCA and distance based DR approaches described in Section IV-B-3, to the distance Location Dependent (LD) scheme, which takes the physical position of the samples as input, i.e., the predicted value is that corresponding to the physically closest training sample.

Table IV summarizes the accuracy performance of the algorithms, which is expressed in terms of $1-NRMSE\times100$. From Table IV, we observe the following:

1) By abstracting from the physical position of the measurements we can provide better estimations, i.e., with proper models and parameter setting, we improve the prediction accuracy with respect to a distance LD scheme.
2) With proper formal dimensionality reduction techniques, such as, FE-PCA and FS-SPCA, we improve the prediction accuracy with respect to a distance DR based

TABLE III: Learning parameters.

Parameter	**Bagging**	**AdaBoost**
Num. iter.	1000	100
k-NN		
Num. of neighbours (k)	3	5
Distance	Euclidean	Euclidean
NN		
Size	1200	1200
Maxit	10000	10000
Decay	0.01	0.01
DT		
Num. of trees (T)	500	300
SVM		
Kernel	RBF	RBF
Epsilon (ϵ); Cost C	$0.2; 5$	$0.2; 5$

approach. In particular, we observe that if we transform the whole data by applying FE-PCA, we notice that this approach provides better results than the FS-SPCA approach.

3) FS-SPCA is a very useful approach if we are interested in excluding features to retain minimal redundancy. In this context, we can reduce the dimensionality of the data up to 92 features and still maintain almost the same accuracy, i.e., we observe that if we consider the 92 features, we lose only 1% of accuracy with respect to FE-PCA.

4) When we build ensemble methods, SVM and NN regression models perform better when they are bagged than when they are boosted. This was to expect, as Bagging combines many weak predictors (i.e., the predictor is only slightly correlated with the true prediction) to produce a strong predictor (i.e., the predictor is well-correlated with the true prediction). This works well for algorithms where by changing the training set, the output changes. The opposite behaviour can be found in k-NN and DT regression models, i.e., when these algorithms are boosted the models tend to provide better results than when they are bagged. That is, in order to improve the performance of AdaBoost, we use sub-optimal values, k for k-NN, and T for DT (described in Table III), i.e., we use values that are not that good, but at least better than random. As a result AdaBoost does its job properly. This is not the case for SVM and NN. Since these learning algorithms do not have an input parameter that we can adjust to obtain a weak predictor without affecting the accuracy of the model, the probability that these algorithms provide better performance when are boosted than when are bagged is lower. Some initial results can be found in [23].

5) By applying different regression models and in particular, when the SVM regression model is bagged, we improve by 11% the overall accuracy of the prediction with respect to the distance LD scheme. Moreover, while in overall, all the regression models exhibit high accuracy (over 90%), there is still a significant reduction in error between the learning algorithms. For example, in terms of the NRMSE, k-NN exhibits an error of 10%, while SVM halves this value to 5%.

TABLE IV: Overall model accuracy

Approaches	**Regression model**	**Bagging**	**AdaBoost**
1. FE-PCA	1.1 k-NN	90.33%	91.98%
	1.2 NN	93.44%	92.28%
	1.3 SVM	$\mathbf{94.70}\%$	94.07%
	1.4 DT	92.84%	93.60%
2. FS-SPCA	2.1 k-NN	89.69%	90.88%
	2.2 NN	92.41%	91.78%
	2.3 SVM	93.62%	92.87%
	2.4 DT	91.22%	92.08%
3. Distance based DR	3.1 k-NN	88.43%	89.11%
	3.2 NN	90.09%	88.85%
	3.3 SVM	91.80%	90.98%
	3.4 DT	89.26%	90.20%
4. Distance LD	4.1 Physical position	83.03%	83.92%

Given the complexity of current and future networks, there is the need for clearly understanding what parameters are relevant and what are not when planning the network for offering a certain QoS. The results of this paper show that the massive exploitation of MDT-based data captured in a complex and heterogeneous operational network allows predicting the QoS in another point of the network. This is why we believe this is an important result for planning new deployments while complying with the QoS requirements the operator targets. More specifically, the SVM ensemble with Bagging is the most promising technique for its better performance and high accuracy.

VI. Implications for network planning

Our results suggest that predicting QoS metrics that relate the interest of the operators and that of the users (i.e., PRB/MB) is feasible with high accuracies, which we believe is an improvement with respect to those tools mostly focusing on RF predictions exclusively.

Another remarkable result as far as planning is concerned is that our results confirm that measurements gathered at arbitrary points of the network throughout its lifetime can be exploited to plan other arbitrary future deployments, hence exploiting historical operational data in a more systematic way than has hitherto been done.

Furthermore, the results obtained from dimensionality reduction techniques show that handling all these historical operational data would require huge amounts of storage as well as processing capabilities. In this respect, dimensionality reduction techniques, such as those proposed in this paper, can make these requirements less stringent. In fact, FE-PCA would present less inputs to the SL step of the data processing chain at the cost of a prior processing of features. On the other hand, FS-SPCA would simplify the initial feature processing, since selected features are taken as they are, but the cost would be the higher storage need (i.e., 92 inputs vs. 10 inputs to the SL step). Therefore, it will depend on the specific network and operator to select whether computing or storage should be optimized and to decide whether the price paid in terms of accuracy is acceptable.

VII. Conclusion

In this paper, we applied machine learning techniques for data analysis of QoS measurements of heterogeneous networks. The goal was to predict QoS and evaluate their potential application to network planning of complex mobile networks. We compare results from different regression techniques, namely k-NN, NN, SVM, DT for different amounts and kinds of input features selected by applying dimensionality reduction techniques. Additionally, we build ensemble methods that combine multiple learners to enhance the performance of each regression analysis in a reduced space. We showed that: 1) data analysis through regression techniques can be done in the reduced space more accurately than in the original space, 2) in regards to the analysis of the data in the reduced space, we notice that by creating new combinations of features (FE-PCA), or reducing the number of input-features under consideration (FS-SPCA), we preserve maximum information and retain minimal redundancy. For example, considering heterogeneous kinds of inputs (e.g., RSRP and RSRQ), we benefit the SPCA, as we can significantly reduce the dimensionality of the dataset without changing the data. That is, by promoting sparsity we get features capturing a maximum of variance. Therefore, we can exclude a significant amount of features, and include the features that give the most useful information, 3) while in overall, all the regression models exhibit high accuracy, bagged SVM learning model is the one that better fits our needs, and exhibits more accurate predictions. As a consequence, we can provide better estimation of QoS in a complex heterogeneous scenario. In conclusion our results suggest that predicting QoS metrics that relate the interest of the operators and that of the users (i.e., PRB/MB) is feasible with high accuracies, and that such metric can be predicted in an arbitrary point of the network based on historical operational data gathered throughout the network, hence justifying the application of the presented ML techniques when planning future complex mobile networks.

References

[1] Nicola Baldo, Lorenza Giupponi, Josep Mangues-Bafalluy, "Big Data Empowered Self Organized Networks," *in proc. of The 20th IEEE European Wireless (EW) Conference; Barcelona, Spain*, 2014.

[2] Seppo Hamalainen, Henning Sanneck, Cinzia Sartori, *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. John Wiley and Sons, 2012.

[3] Johansson, J., Hapsari, W.A., Kelley, S., Bodog, G., "Minimization of Drive Tests in 3GPP Release 11," *IEEE Communications Magazine*, November 2012.

[4] F. Chernogorov and T. Nihtilä, "QoS Verification for Minimization of Drive Tests in LTE Networks," in *Proceedings of the 75th IEEE Vehicular Technology Conference, (VTC) Spring, Yokohama, Japan*, May 2012, pp. 6–9.

[5] F. Chernogorov and J. Puttonen, "User satisfaction classification for Minimization of Drive Tests QoS verification," in *24th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2013, London, United Kingdom*, September 2013, pp. 2165–2169.

[6] Jessica Moysen, Lorenza Giupponi, Nicola Baldo, Josep Mangues-Bafalluy, "Predicting QoS in LTE HetNets based on location-independent UE measurements," *in proc. of 20th IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks, Guildford (UK)*, 2015.

[7] Gordon Mansfield, "HetNet - Small Cell Placement and resulting performance," *AT&T presentation given at Mobile World Congress*, 2015.

[8] S. T. Roweis, "EM algorithms for PCA and SPCA," *Advances in Neural Information Processing Systems, The MIT Press*, 1998.

[9] CelPlan, "CelTrace $^{\text{TM}}$ Wireless Global Solutions ," *Available at: http://www.celplan.com/products/indoor/celtrace.asp*.

[10] Altman, N. S, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician 46*, 1992.

[11] Bishop M.C., "Pattern recognition and machine learning," *Busines Dia, Llc. Springer Science*, 2006.

[12] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, pp. 199–222, 2004.

[13] Quinlan, J. R., "Induction of Decision Trees. Machine Learning ," *Kluwer Academic Publishers*, pp. 81–106, 1986.

[14] Rokach, Lior; Maimon, O., "Data mining with decision trees: theory and applications," *World Scientific Pub Co Inc.*, 2008.

[15] Opitz, D.; Maclin, R., "Popular ensemble methods: An empirical study," in *Journal of Artificial Intelligence Research 11*, 1999, pp. 169–198.

[16] Rokach, L., "Ensemble-based classifiers," in *Artificial Intelligence*, 2010, pp. 1–39.

[17] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization," *MachineLearning*, vol. 40, pp. 139–157, 2000.

[18] Thomas G. Dietterich, "Machine-Learning Research: Four Current Directions," *American Association for Artificial Intelligence*, 1997.

[19] Zoubin Ghahramani, "Unsupervised Learning," *Gatsby Computational Neuroscience Unit University College London, UK*, 2004.

[20] Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), "The LENA-EPC Network simulator," *Available at: http://networks.cttc.es/mobile-networks/software-tools/lena/*, 2015.

[21] 3GPP, "(Radio) Meeting 51, Simulation Assumptions and Parameters for FDD HeNB RF Requirements, 3GPP Technical Report," Tech. Rep. TSG RAN WG4 R4-092042, may 2009.

[22] Bergstra, James; Bengio, Yoshua, "Random Search for Hyper-Parameter Optimization," *Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[23] Efrain Mayhua-Lopez, Vanessa Gomez-Verdejo, and Anibal R. Figueiras-Vidal, "Boosting ensembles with subsampled LPSVM learners."