Mass Scale Modeling and Simulation of the Air-Interface Load in 3G Radio Access Networks

Dejan Radosavljevik[1], Peter van der Putten[1], and Kim Kyllesbech Larsen[2]

[1] LIACS, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands
{dradosav,putten}@liacs.nl
[2] Deutsche Telecom AG, Landgrabenweg 151, D-53227 Bonn, Germany
kim.larsen@telekom.de

Abstract. This paper outlines the approach developed together with the Radio Network Strategy & Design Department of a large telecom operator in order to forecast the Air-Interface load in their 3G network, which is used for planning network upgrades and budgeting purposes. It is based on large scale intelligent data analysis and modeling at the level of thousands of individual radio cells resulting in 100,000 models. It has been embedded into a scenario simulation framework that is used by end users not experienced in data mining for studying and simulating the behavior of this complex networked system.

Keywords: Mobile Network, Air-Interface Load, Linear Regression.

1 Introduction

This paper reports on a deployed data mining application that has been developed by one of the largest European telecom operators and has been in continuous use ever since. In order to accommodate the continuing strong increase of mobile internet traffic, the operator's Radio Network Department has to continuously monitor and upgrade the 3G Radio Access Network. This requires an Air-interface Load forecast for every radio cell in the network. However, such a detailed forecast was not readily available. Furthermore, there is a need to simulate different scenarios for different parts of the network. Given the complexity of the problem, the dimension of the network and the repetitiveness of the task, a manual approach is out of the question.

In this paper we present a fully automated approach that generates multi-variate linear regression models on a grand scale, using primarily open source tools. The key business value of this research is that it solves a very complex and high impact business problem that cannot be approached by using simpler planning approaches. The exact return is confidential, but cellular network infrastructure forms a major part of an operator investment budget, and this is a key system for tactical and strategic network investment decisions. In the group where this operating company belongs, up to 50% of wireless CAPEX investments are going into the radio access. For reference, operators invest more than 20 billion USD into cellular network infrastructure worldwide. Our methodology is first and foremost intended to ensure that capacity is added in time and at the right place, thus avoiding inefficient investments and poor customer

J. Hollmén, F. Klawonn, and A. Tucker (Eds.): IDA 2012, LNCS 7619, pp. 301–312, 2012.

experience due to traffic congestion, which can ultimately lead to churn. The system has been rolled out to full production use. None of the other operator companies in the telecommunications group uses a similar fine grained approach.

Whilst the core intelligent data analysis algorithms used are not novel, we apply these on a large scale by modeling individual radio cells across a variety of dimensions (section 3 motivates why we model at cell level). This has also been embedded into a simulation framework targeted at non-data miners in tools they are familiar with to enable them to run low level simulation scenarios. Hence, the goal is to provide a case example of an embedded, deployed intelligent data analysis system, dealing with real world aspects such as scale and having major business impact.

As discussed, the technical novelty is not determined by the complexity of the base estimators used. We use simple linear regression models as data inspection has shown that the behavior to be predicted is primarily linear, and experiments confirmed that complex algorithms actually performed worse given the high variance associated with these models. This is not uncommon in real world data mining problems [1]. What makes this problem out of the ordinary is the massive number of models. For each of the 20,000 cells in the network we create five models to predict different kinds of outcomes, resulting in a total of 100,000 models. Model parameters are estimated using ten-fold cross validation, which increases the number of models estimated to over 1 million. This process is repeated on a regular basis, given that the customer base and its behavior, as well as the cellular network itself change constantly. Finally, we do not just deploy the forecasted loads. The underlying regression formulas are provided by the data miners to the end user analysts as simple spreadsheets, which enables them to tune various simulation and forecasting scenarios without further involvement from the data miners. We think this approach can easily be replicated and applied to problems from other industries which require similar predictive models and simulation of networked systems on a large scale, such as for instance sensor networks, retail outlet planning and supply chain logistics.

The rest of the paper is structured as follows. Section 2 describes the load parameters. Section 3 discusses the complex nature of network load and how to approximate it, including a motivation for modeling at the granular cell level. Section 4 describes the construction of the load formulas and the forecasting of the future load of the network using simulation based on these formulas. Limitations and future work are discussed in section 5. Finally, we present our conclusions in section 6.

2 Defining the Air-Interface Load Parameters

The communication between a network cell and a mobile device is separated into downlink communication- directed from the cell to the mobile device and uplink communication- directed from the mobile device to the cell. Therefore, the Air-interface load for a cell consists of the Downlink Load (DL) and Uplink Load (UL). When measuring the actual Air-Interface load typically only the maximum of the UL and DL values is taken. Multiple measures of both DL and UL can be devised. A cell is considered to be in overload if the load is above a certain threshold. When a cell is

in overload, it cannot serve additional customers that demand its resources. Obviously, all cells in overload require an adequate upgrade.

Most of the literature on telecom network is related to network optimization or load control rather than load prediction [2, 3, 4, 5]. Therefore, there was no previous knowledge on which of the measures of either Downlink or Uplink Load would be possible to predict with the least error, so several measurements of these were chosen.

We used a number of measures to characterize uplink load. Firstly, the *Count of RAB (Radio Access Bearer) Releases Due To Interference* was chosen; a RAB is a cell resource which is necessary to be assigned to the mobile subscriber/device in order for any voice/data transaction to be possible. Normally, it is released after it is no longer necessary, unless there were circumstances (e.g. interference from other users or cells) which caused it to be dropped [2]. Secondly, we used the *Average Noise Rise (ANR)*, measured per hour in dBm (Decibels per milliwatt), which is the difference between the Uplink power received in a given time when a number of users consume cell resources, and the Uplink power of the same cell when it is not serving any users at all [3]. Thirdly, we chose the *Average Noise Rise on Channels Dedicated to Release 99 Capable Devices* (refers to lower data transfer speed up to 384 Kbps). Two additional uplink measures were considered: *Count of RAB (Radio Access Bearer) Setup Failures* and *Count of RRC (Radio Resource Control) Setup Failures*. These measurements were discarded at later stages of the process due to the very low number of models that could be generated because of too many zero-values.

The parameters used as measures for downlink load were the following. Firstly, we used *Percentage of Consumed Downlink Power (CDP),* similar to Downlink Noise Rise [4]. Downlink power is a finite cell resource and it amounts to 20W. Each mobile device/user gets a portion of this, which is proportional to the bandwidth they require. In an overload situation there is no more power to be distributed. The other downlink load measure was the *Count of "No Code Available" Situations* (NCA) [5]. Each cell has 256 codes that can be assigned to a mobile device for a voice call or a data session. The higher the downlink bandwidth required, the higher the number of codes will be assigned. After all the codes have been assigned, the next devices that requests a code from the cell, gets a "no code available" message and cannot use the cell resources.

As input parameters we used different measures from the Nokia Data Warehouse [6], a tool that is used in telecom operators to monitor Radio Network Performance. Even though we included input parameters related to voice services, most of the input parameters are related to consumption of Data Services, because they require more of the cell resources. These include the following: Average Voice Call Users, Average Release 99 Uplink users, Average Release 99 Downlink users, Average High Speed Uplink Packet Access (HSUPA) users, Average High Speed Downlink Packet Access (HSDPA) users, Maximum HSUPA users, Maximum HSDPA users, Total RRC attempts, Total Active RABs, Total Voice Call RAB Attempts, Total Data Session RAB Attempts, Average Downlink Throughput, Average Uplink Throughput, Average Soft Handover Overhead Area (measures the intersection of coverage of the particular cell with other cells), Average Proportion of Voice Traffic originated in that cell (as opposed to traffic originated in other cells and handed over to that cell),

Average Proportion of Data Traffic originated in that cell. Forecasts for future values of the input parameters were available at the operator.

Both the input and the output parameters were taken on per cell per hour level.

3 Approximating and Predicting the Air-Interface Load

Most of the literature on load forecasting is related to electrical networks. A good overview is presented in [7]. Various methods have been deployed for this purpose: regression models, time series, neural networks, expert systems, fuzzy logic etc. The authors state a need for load forecasts for sub-areas (load pockets) in cases when the input parameters are substantially different from the average, which is a case similar to different cells in a mobile telecom network.

Related to mobile telecommunications, data traffic load (which is different than air interface load) focusing on a highly aggregated link has been forecasted in [8], comparing time series (moving averages and dynamic harmonic regression) with linear and exponential regression. Also, Support Vector Regression was used by [9] for link load prediction in fixed line telecommunications.

In order to forecast the future load for each cell in the network, it is necessary to understand the relationship between the input parameters (causing the load situation) and the current load. The input parameters in case of the Air Interface load are all parameters which can be made accountable for the load situation in the cell (Section 2). Therefore, the load parameter (output) can be expressed as $L=f(x_1, x_2,\ldots, x_n)$. Ideally, the load of each cell x in a given time could be expressed as the sum of all users consuming resources of that cell at the particular time multiplied by the amount of resources they use plus the interference between that cell and all the other cells in the network (in practice limited to the neighboring cells):

$$L(x) = \sum_{i=0}^{m} \sum_{j=1}^{n} User_i * Resource_j + \sum_{y=1}^{z} interference(x, y) \quad (1)$$

where m is the count of users that are using the resources of cell x, n is the count of resources of the cell x, z is count of all cells in the network and $interference(x,y)$ is the interference measured between cell x and y.

Unfortunately, there was no tool that would provide such a detailed overview. In order to approximate the load function, we recorded the five different load parameters (outputs) and 16 input parameters described in section 2, on an hourly basis during 6 weeks. This provided approximately 1,000 instances for each cell to build a predictive model, or 20,000,000 instances in total.

One of the choices to be made was whether a distinct formula for every cell shall be built or – alternatively – a common formula valid for all cells should be used. The approach where a model is created for each cell was chosen, due to the network experts' conviction that each cell is different, and a unified approach simply would not work, because some of the parameters influencing the load of each cell were immeasurable and unpredictable.

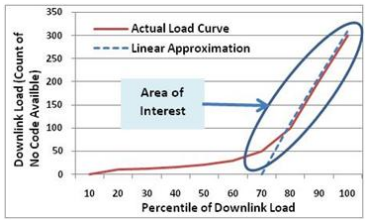

Fig. 1. Actual Downlink Load vs. Linear approximation

Next, the domain experts were intrinsically interested in being able to model cells that actually do not behave like other cells, especially when these are highly loaded. Furthermore, there would be a challenge in normalizing with respect to the varying capacity of the cells, i.e. what where the cell sized to handle. Finally, we hypothesized that not just model parameters could differ by cell, but also the optimal selection of features, similar to the load pockets explained by [7].

The choice of linear regression [10] was made due to several reasons. First of all, even though the distribution of the values of each of the load measures we are trying to predict vary between close to linear and close to exponential, we are only interested in the higher values of the load curve, and this can be approximated quite well with linear regression, as shown on Figure 1. For this purpose, before constructing the regression formulas, we remove all zero instances. Furthermore, linear regression is a very fast algorithm compared to other methods, which is very useful when it is necessary to develop a large number of models in a short time. Even though it is imaginable that better results might be achieved by using non-linear regression, regression trees, or other algorithms, this might not be necessary (Figure 1). Also, simple low variance methods such as linear regression frequently perform much better in practice than more complicated algorithms, which can very often over fit the data (e.g. high variance algorithms such as neural networks). In other words, in real world problems variance is typically a more important problem than bias when it comes to data preparation and algorithm selection [1]. Trials on a smaller sample were already made with regression trees, but apart from the visibly increased time consumption, the accuracy did not improve. On the contrary, in some instances it decreased.

Last but not least, linear regression is easy to implement, easy to explain and its results and models are easy to export for other use. Exporting the models to Excel was of crucial value, as analysts would use them in order to predict the future load of each cell, by scaling the input parameters, based on internal forecasting models. In other words, this allows non data miners to simulate future network load based on changes in the various type of network traffic, using simple tools they are familiar with.

4 Approach Description and Results

In this section we will describe how the models are being generated and put to work. This includes the tools that were used, a detailed description of the approach, the results of this mass modeling process and the process of forecasting the future load.

4.1 Tools

The tools used in this research are either open source, or can be found in the IT portfolio of any telecom operator. These are the following:

Nokia Data Warehouse [6] was used for data collection for both the input and the output parameters. This software tool was already a part of the Network/IT infrastructure of the operator. It contains technical parameters related to the mobile network performance. The most important feature of this tool for our research was that it contained hourly aggregates of all the input and output parameters we used in our research (Section 2). Obviously, any other tool that collects data about network performance could have been used. This is the only domain specific tool from our process.

Load Prediction and Simulation Data Mart. This is an Oracle Database 10g- 64 bit v10.2.0.5.0 [11] used for all our task specific data preparation and manipulation**.**

Due to the fact that the necessary input and output parameters were stored at different tables in the Nokia Data Warehouse, we needed a separate database where we could manipulate the data easier (e.g. merge tables, create indexes, and build the final flat table). This reduced the duration of the data collection and data preparation process from two weeks to 1 day by productizing data collection. Because we are rebuilding and rescoring models on a continuous and automated basis, this was a key improvement. Any other database platform (commercial or open source) could have been used. We opted for Oracle based on license availability.

WEKA 3.6.4 x64, an open source data mining platform [12], was used for building the linear regression formulas and validating them. Of course, any other tool capable of deriving linear regression can also be used for this purpose. That said, this shows that even a research focused open source tool like WEKA can be used in critical commercial settings, at high complexity (20.000 cells, 5 models each, around 1000 instances each).

Strawberry Perl for Windows v5.12.3 [13] is an open source scripting language that we used in order to create the script that is the core of this approach. Our script creates WEKA input files by querying the Oracle database, generates the regression models by executing calls to WEKA, and stores the regression formulas and the cross-validation outputs (Correlation Coefficient, Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error, and Total Number of Instances used to build the model) in csv files.

MS Excel 2010 [14] – part of MS Office 2010, was used to predict the future load of cells, using the regression formulas created by WEKA and extrapolations of the input values using a internet traffic model scaling factors based on handset/internet usage developments (internal to the operator).

4.2 Process Description

A graph of how our approach uses these tools to derive and store the regression models is presented on Figure 2. In the core of our approach is a Perl [15] script that automated the derivation of the regression formulas for each cell. This script executed calls to WEKA and queries to the Oracle Database. It works in the following manner:

```
Get list of cells from the database;
for each cell
  Run a query on the database to isolate only the data
  related to that cell (all the input and the 5 output
  parameters);
  Make separate files for each of the 5 load parameters;
  For each of the 5 load parameters
Filter out all instances where the load is 0;
    Select only the relevant variables for the regression
    formula of that cell, using a wrapper approach;
    Build the linear regression model and store it;
    Validate the model- Use 10-fold cross-validation;
    Store the formula, the number of instances used to
    build it, the correlation between the predicted and
    actual value for load, the Mean Absolute Error (MAE)
    and the Root Mean Square Error (RMSE) as reported
    from the cross-validation;
```

While generating the models/regression formulae, we used a wrapper [16] approach. Wrapper approaches automatically select the best variables for predicting the outcome, taking into account the algorithm to be used, which in our case is linear regression. They do not necessarily perform better or worse than filter approaches [17]. Our motivation to use the wrapper approach was to avoid human interaction with the model building process as much as possible, which obviously makes the process much faster.

It is worthwhile mentioning that the optimal feature and linear regression model selection were performed using 10-fold cross validation [3]. This was done in order to balance between cells with large sample of non-zero instances and cells with a smaller sample. The reported correlation coefficient, MAE and RMSE are averages from the 10 repetitions. Using 10-fold cross validation already provides a good estimate of the accuracies of these formulas. Of course, we intend to test them on a completely new dataset, not only to confirm the accuracies achieved, but also to find out when is a good time to update the model. We expect that updates should be necessary every few months, because of the reconfiguration of the network, additions of new cells and upgrades to the existing ones.

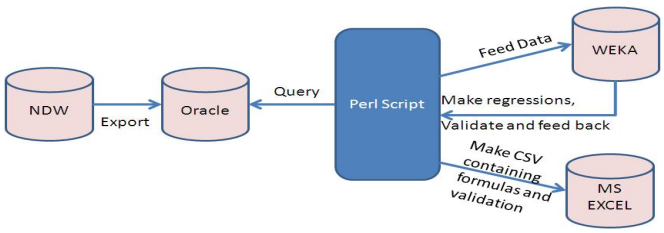

Fig. 2. Communication Graph of the Tools used

4.3 Results of the Modeling Process

Using this process we were able to run 100,000 (5 outputs for 20,000 cells) regressions in less than 1 week, by just one click. This did not result in 100,000 models, because in some cases it was impossible to derive a formula due to the large number of instances that were filtered out for zero load. But, in order to measure the load of a cell, it is sufficient that a model is generated for at least one output variable. Cases of cells where it was not possible to generate a model for any of the variables were rare. Furthermore, cells that do not show any load by the means of the five output variables, are not of interest for our problem situation. For practical purposes, we will only present the modeling results for two of the five output variables we used to describe the air interface load in section 2. We chose one Uplink Load measure- Count of RAB Releases due to Interference-RRI, presented in Table 1, and one Downlink Load measure- Count of No Code Available- NCA, presented in Table 2.

Both Table 1 and 2 have the same structure. In the first column Bands of Averages for the respective output variables RRI and NCA are given. The second column contains the count of cells that falls into this band. The third column presents the average count of non-zero instance (NZI) in each band.

Table 1. Regression Modeling Results for Count of RAB Releases due to Interference (RRI)

Count of RAB Releases due to Interference (RRI)	Count of Cells	Avg Count of NZI	Avg nonzero RRI	Avg CC	Models Built Vs Number of Cells
RRI<=1	8373				
1<RRI<2	7344	89.4	1.3	0.141	0.582
2<=RRI<3	1359	229.4	2.4	0.545	0.769
3<=RRI<5	972	296.2	3.8	0.658	0.829
5<=RRI<10	780	365.0	7.0	0.751	0.881
10<=RRI<20	503	407.6	14.0	0.810	0.905
RRI>=20	538	431.3	56.8	0.873	0.920

Table 2. Regression Modeling Results for Count of No Code Available (NCA)

Count of No Code Available (NCA)	Count of Cells	Avg Count of NZI	Avg nonzero NCA	Avg CC	Models Built Vs Number of Cells
NCA<=1	682				
1<NCA<2	792	130.5	1.6	0.454	0.775
2<=NCA<5	2229	331.5	3.3	0.635	0.971
5<=NCA<10	2321	500.5	7.3	0.773	0.994
10<=NCA<20	3420	597.2	14.7	0.836	0.994
20<=NCA<30	2706	681.3	24.9	0.862	0.994
30<=NCA<50	3858	732.0	39.0	0.861	0.998
50<=NCA<100	3063	758.1	67.8	0.872	0.996
NCA>=100	798	760.2	208.7	0.790	0.992

In other words, it presents the number of instances used to build the regression, because we only took non-zero output values into account. The fourth column, perhaps redundant, presents the average of the variable in that band. The fifth column presents the average Correlation Coefficient (CC) between the predicted and actual variables in the particular band. These Correlation Coefficients are the result of the 10-fold cross validation. The last column presents the ratio between the number of formulas that were generated and the total count of cells in each band. Namely, for certain cells it was not possible to build the regression because of a very low number of non-zero instances.

The results can also be evaluated by using two criteria: The Correlation Coefficient and The Ratio of The Models Built (the last two columns in Tables 1 and 2). Obviously, the Correlation Coefficient in both these cases (RRI and NCA) grows alongside the number of instances, which is to be expected. But, because of the choice we made at the beginning of the research, to focus on the higher levels of load and eliminate the zero values, the Correlation Coefficient between actual and predicted values also grows as the output variable value is higher. The case is similar with the Ratio of the Number of Formulas built. The difference is more evident in the case of RRI (Table 1), because this is an event that occurs less frequently than NCA. The number of formulas here is also lower, especially in cases with low RRI levels. But, as mentioned before, we are not interested in these cases.

4.4 Forecasting Future Load

Once the load formulas have been derived it is possible to forecast the future load situation if the changes in the describing parameters are known. These changes of the input parameters are described by means of scaling factors. The scaling factors are calculated by using a traffic forecast model developed by the operator (out of scope of this paper).

This is done in the following way:

```
For each output variable
  For each cell
    Select the top 100 instances of the output variable
    and its corresponding values for the input varia-
    ble;
    Make averages of these input variables;
    Scale the input variables up or down, according to
    scaling factors developed by a traffic model;
    Feed the scaled values of the input parameters into
    the regression formula;
    If resulting value > critical threshold then cell
    should be upgraded;
```

This part of the process is performed in a tool as simple as MS Excel. This was a key driver for the business success of the solution. In our experience the importance of the Deployment step in the data mining process is generally underestimated. By

providing not just the scores but also the underlying models in a format and tool that was immediately usable and tunable to end users who are not data miners, the solution was readily accepted and also used in new ways not necessarily intended by the data miners, for instance detailed simulation scenarios.

We mentioned in Section 2 that the Air interface load is calculated as a maximum of Uplink Load and Downlink Load. This means that a cell will be upgraded if it is in overload on either the Uplink or the Downlink. In terms of our approach, a cell is upgraded if any of the five output variables, used as measures of Uplink or Downlink load, are above a critical value.

5 Limitations and Future Work

The regression formulas developed by this approach can be used on a long term basis only if the mobile network stays the same (is frozen) over a longer period. But, this is not the case. The cellular network is a system of very complex dynamics. The many changes that occur, such as hardware and software updates, network reconfigurations and optimizations, as well as network upgrades and roll-out of new cells cannot be taken into account in advance. It is necessary to collect a new dataset and rebuild the regression formulas, in order to incorporate all these changes into the model. This is why the process described in this paper is scheduled for execution every 3-4 months.

Further evaluation of the quality of the derived load formulas of course also involves the comparison of the predicted load with the actually measured load in the future. It should however be noted that there a lot of factors impeding a direct comparison. As noted above, all changes to the settings of a cell within the forecasting timeframe affect the load formula, which means that after such changes the derived formula is - at least to some degree - no longer correct. For this reason it will be challenging to really quantify the accuracy of the predictive model. Developing a fair method of evaluation, which would incorporate the network changes, would be beneficial. In terms of the core algorithms, we do want to keep the benefit of using a simple, fast and robust low variance approach such as linear regression.

However, we do plan to explore a methodology that would allow us to combine a global network model with local models for each cell, for instance multitask or transfer learning [18]. In principle, we have almost infinite data available for most cells, so local models cannot be improved by a global model. Nevertheless, there could be exception for a non select small number of cells. Last but not least, a clustering approach could be devised to group cells with similar formulas or levels of load, thereby generating new knowledge for the telecom domain experts.

6 Conclusions

In this paper we presented a very simple yet effective approach of applying data mining in commercial surroundings. Unfortunately, data mining is still seen as a black box in many industries, telecom not excluded. Even though some data mining activities are taken, typically in the Marketing/Customer Retention field, there is a myriad

of other possibilities in business where data mining can be applied. In our opinion, it is better to start with simple methods, such as regression, because it is easier to understand them. Once these simple approaches gain acceptance, and familiarize the industries with data mining, opportunities to apply more advanced techniques will arise.

In our result section we show that it is easier to accomplish a target, if one is focused on it. Namely, with our approach we wanted to target cells where some load (non-zero load) occurs, in order to predict the part that really matters more correctly: the high end part of the load curve (the cells in overload). In other words, as the network load grows, so does the quality of the model's predictions. We willingly sacrificed the models' performances within the lower loaded cells, because they are of no interest.

Next, one of the key values of the approach is that a large number of regression models (close to 100,000) are developed in a very short period of time with minimum human interaction. In order to do this, we deployed a simple algorithm such as linear regression, motivated by its speed and other benefits explained earlier, a wrapper feature selection, in order to avoid human interaction, and 10-fold cross validation which makes the models statistically sound. Manually, this task would be impossible. Obviously, the possibility to generate these formulas was crucial to the operator.

Another large benefit of our approach is that after the models have been generated, they are exported into Excel sheets, which allows a team of radio network analysts, which are not data miners, to use these formulas for forecasting the future network load. This allows them to simulate multiple traffic scenarios by scaling the current input parameters. These scenarios include evaluations of network investments necessary to accommodate localized user growth due to targeted marketing campaigns or more extreme, adding a new wholesale client- or an MVNO (Mobile Virtual Network Operator).

Typically, planning network upgrades is a reactive process. Our approach makes it proactive, which was acknowledged by the operator, who has fully integrated our approach into its network upgrade planning and budgeting activities. Of course, due to the fast pace network changes, the formulas need to be upgraded every 3-4 months, but this is also scheduled as a part of a standard process. Due to confidentiality, we cannot disclose the exact return of this project, but given that the network is the key resource of an operator, the investments into its upgrades are quite sizeable. To our knowledge, this is the first time a telecom operator has applied data mining in order to create a proactive network upgrade management process. This allows the operator to manage network performance better and avoid extreme congestion situations, which can result in degraded customer experience and loss of reputation for the operator. As mentioned at the beginning, the research was performed at a large telecom operator with branches in many European countries. At the moment, our research is deployed in only one of the countries where this operator is present, but efforts are made to replicate it in the other branches as well.

Perhaps one of the most interesting aspects of this approach is the extremely low cost. Given that we used the existing IT infrastructure (Server, Nokia Data Warehouse, Oracle, Excel) combined with open source tools (WEKA, Perl), the only cost that incurred are the 1 week Processing Time Cost (of the Server) and the labor cost of the employees in this project. Also, the Oracle Database that we used can be

replaced with a less expensive or free database alternative in order to further reduce the cost, in case the potential user of our approach does not have an Oracle License.

Last but not least, we would like to point out the possibility of applying our research onto domains other than telecom. This approach would be applicable to any other industry where large scale regression models are necessary. This can be accomplished simply be replacing the data source, in this case Nokia Data Warehouse, with a data source suitable for the industry that would like to apply our research.

References

1. van der Putten, P., van Someren, M.: A Bias-Variance Analysis of a Real World Learning Problem: The CoIL Challenge 2000. Machine Learning 57(1-2), 177–195 (2004)
2. Yates, R.: A framework for uplink power control in cellular radio systems. IEEE JSAC 13(7), 3141–3147 (1995)
3. Geijer Lundin, E., Gunnarsson, F., Gustafsson, F.: Uplink load estimation in WCDMA. In: Proc. IEEE Wireless Communications and Networking Conference (2003)
4. Muckenheim, J., Bernhard, U.: A Framework for Load Control in 3rd Generation CDMA Networks. In: Proc. of the IEEE Global Telecommunications Conference, vol. 6, pp. 3738–3742 (2001)
5. Natalizio, E., Marano, S., Molinaro, A.: Packet scheduling algorithms for providing QoS on UMTS downlink shared channels. In: IEEE VTC, vol. 4, pp. 2597–2601 (2005)
6. Nokia Siemens Networks: Nokia Siemens Networks WCDMA RAN, Rel. RU10- System Library, v.1: RNC Counters – RNW Part. Nokia Siemens Networks. Proprietary and Confidential (2008)
7. Feinberg, E.A., Genethliou, D.: Load Forecasting. In: Chow, J.W., Wu, F.F., Momoh, J. (eds.) Applied Mathematics for Restructured Electric Power Systems, pp. 269–285. Springer, Heidelberg (2005)
8. Svoboda, P., Buerger, M., Rupp, M.: Forecasting of Traffic Load in a Live 3G Packet Switched Core Network. In: Proc. of 6th International Symposium on CNSDSP, pp. 433–437 (2008)
9. Bermolen, P., Rossi, D.: Support vector regression for link load prediction. Computer Networks 53(2), 191–201 (2009)
10. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Technique, 2nd edn. Morgan Kaufmann, San Francisco (2005)
11. Oracle.: Oracle Database Documentation Library, `http://www.oracle.com/pls/db102/homepage`
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
13. Strawberry Perl, `http://strawberryperl.com/`
14. Microsoft Corporation: Microsoft Excel, `http://office.microsoft.com/en-us/excel/`
15. Christiansen, T., Torkington, N.: Perl Cookbook, 2nd edn. O'Reilly, Sebastopol (2003)
16. Kohavi, R., John, G.: Wrappers for feature subset selection. In: Artificial Intelligence 1997, pp. 273–324 (1997)
17. Tsamardinos, I., Aliferis, C.: Towards principled feature selection: Relevancy, filters and wrappers. In: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistic (2003)
18. Caruana, R.: Multitask Learning. Machine Learning 28, 41–75 (1997)