Chapter 5

Large Scale Predictive Modeling for Micro-Simulation of 3G Air Interface Load

Radosavljevik, D., van der Putten, P.

Published in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 1620–1629 (2014).

This chapter outlines the approach developed together with the Radio Network Strategy and Design department of T-Mobile Netherlands, part of the Deutsche Telekom group, 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 30,000 models produced in one day. 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, as an example of a systematic approach to the deployment step in the KDD process. This system was a part of a standard business process in T-Mobile Netherlands for more than two years. This operator became a competence center for predictive modeling for micro-simulation of 3G air interface load for three other operators of the Deutsche Telekom group. A similar approach, based on different network parameters was also developed by the operator for 4G networks.

5.1 Introduction

This chapter 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 for more than two years. In order to accommodate the continuing strong increase of mobile internet traffic, the operator's Radio Network Department had to continuously monitor and upgrade the 3G Radio Access Network. This required an Air-Interface load forecast for every radio cell in the network, including indications of denial or interruption of delivering service. However, such a detailed forecast was not readily available. Furthermore, there was 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 was out of the question. Hence, the research question for this chapter is:

How can data mining be used to predict 3G mobile network interface load and simulate it under different scenarios?

In this chapter we present a fully automated approach that generates multivariate linear regression models on a grand scale, using primarily open source tools. The key business benefit of this research is that it solved a very complex and high impact business problem that could not be approached by using general planning approaches.

Traditional approaches for mobile network load forecasting have a number of practical issues. These are most often analytical or Monte Carlo based approaches (Mäder and Staehle, 2004). The load formula used is typically a general purpose analytical model, derived from physics knowledge and theory rather than from modeling on actual data, let alone being based on data from a specific operator. Furthermore, the inputs required are difficult to measure and forecast, or do not relate to changes in customer behavior such as call usage, which is easier to understand and

use for scenario purposes. Our approach makes it easier to translate customers and their usage into network load. To our knowledge, this is the first time a data mining approach has been used for Air Interface load prediction of a 3G mobile network.

In terms of business benefits, the exact return is confidential, but cellular network infrastructure forms a major part of an operator's 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 worldwide invest more than 20 billion USD into cellular network infrastructure. 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 experience due to traffic congestion, which can ultimately lead to churn. Last but not least, Internet access is recognized as a right by law in several countries, as a part of the rights to Freedom and Expression of Opinion. By adding capacity at the right places, operators provide a valuable social service, given the growing importance of communications and social media in everyday life. This is a data mining application in telecommunications that does not raise the usual privacy concerns; on the contrary it serves a social function by mediating high quality internet access.

An early version of our approach has been published in Radosavljevik, van der Putten and Kyllesbech Larsen (2012). Since then, our system has been rolled out in full use in T-Mobile Netherlands, the operator where it was developed. None of the other operator companies in the Deutsche Telekom group used a similar fine grained approach. Therefore, a more universal approach applicable to the other operators, too had to be developed. This involved dealing with complexities such as different network equipment vendors using different performance management systems and lack of certain measurements we have introduced in Radosavljevik et al. (2012). Nevertheless, the results of our new approach were very positive. Hence, this operator became a competence center for predictive modeling of 3G Air Interface load and it was performing this task for operators of the same telecommunications group in four countries.

Whilst the core intelligent data analysis algorithms used were not novel, we applied these on a large scale by modeling individual radio cells across a variety of dimensions (Section 5.3 motivates why we modeled at cell level). This has also been embedded into a simulation framework targeted at non-data miners using 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. Extensive simulations have been carried out by the operating companies using the system, and also novel use cases for scenario simulation analysis have been developed and applied.

As discussed, the technical novelty is not determined by the complexity of the base estimators used. We used 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 (van der Putten and van Someren, 2004). What makes this problem out of the ordinary is the massive number of models. For each of cell in the network we create four models to predict different kinds of outcomes, resulting in a total of 30,000-100,000 models, depending on the amounts of cells in the network. 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 behavior, as well as the cellular network itself change constantly.

Finally, we did not just deploy the forecasted loads. The underlying regression formulas were provided by the data miners to the end user analysts as simple spreadsheets, which enabled them to tune various simulation and forecasting scenarios without further involvement from the data miners. This turned out to be not just a practical benefit, but a major opportunity for the business as a range of simulation use cases were explored that were not envisioned by the data miners up front.

We think that this approach, including the concept of decoupling data mining from forecasting and simulation processes, can easily be replicated and applied to problems from other industries. Examples of this are problems that require similar predictive models and simulation of networked systems on a large scale, such as for instance sensor networks, retail outlet planning, supply chain logistics and revenue predictions for products with a complex billing process (which we have already applied, see chapter 6).

The rest of the chapter is structured as follows. Section 5.2 describes the load parameters. Section 5.3 discusses the complex nature of network load and how to approximate it, including our motivation for modeling at the granular cell level. Section 5.4 describes the construction of the load formulas and forecasting of future network load using simulation, as well as other simulation scenarios. Limitations and future work are discussed in Section 5.5. Finally, we present our conclusions in Section 5.6.

5.2 Defining the Load Parameters

In this section we will describe how we measured the load for a cell, plus the underlying attributes that we used to predict future load. Both output and input parameters were measured per individual cell per hour.

5.2.1 Output Parameters

The communication between a network site/tower (radio network element that provides access to the mobile network) and a user's mobile device is separated into

downlink communication- directed from the site to the mobile device and uplink communication- directed from the mobile device to the site. Each physical site contains radio antennae that typically create three geographic cells of the mobile network. These cells share the physical resources of the network site.

Therefore, the Air-interface load for a cell consists of the Downlink Load (DL) and Uplink Load (UL). Multiple measures of both DL and UL can be devised. A cell is considered to be in overload if either the uplink or the downlink load is above a certain threshold. When a cell is in overload, customers demanding its radio resources cannot be served adequately. Obviously, all network sites containing cells in overload require an adequate upgrade.

Most of the background literature on telecom networks is related to network optimization or load control rather than load prediction (Geijer Lundin, Gunnarsson and Gustafsson, 2003; Muckenheim and Bernhard, 2001; Natalizio, Marano and Molinaro, 2005; Yates, 1995). In our previous research (Radosavljevik et al., 2012) we used the following measurements of load as output parameters: Count of RAB (Radio Access Bearer) Releases Due To Interference (Yates, 1995), Average Noise Rise (Geijer Lundin et al., 2003) and 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 a very low number of models that could be generated because of too many zero-values.

In Radosavljevik et al. (2012), we used the following parameters as measures for downlink load: Percentage of Consumed Downlink Power (Muckenheim and Bernhard, 2001) and Count of "No Code Available" Situations (Natalizio et al., 2005).

However, some of these measures were specific to Nokia Data Warehouse (Nokia Siemens Networks, 2008), a performance management tool deployed at T-Mobile Netherlands where our research originated, or were not measured by other operators that were looking to use our system. Therefore, we used universal measurements, which are applicable to performance management systems of other vendors, such as Ericsson (Ericsson, 2013), Huawei (Huawei, 2013) or MyCom (MyCom, 2013). Therefore, in our new approach we picked measurements that are both compliant to the 3gpp Mobile Broadband Standard (3gpp, 1999) and universally defined and measured across the operators which are part of the Deutsche Telekom group.

Percentage of Uplink Load, also known as UL Carrier Load Percentage (3gpp, 1999) is the ratio between the total received power level on that carrier and the maximum acceptable level of interference.

$$UL\_LOAD = 100 * (1 - \frac{1}{10^{\frac{MeanRTWP - MinRTWP}{10}}}) \tag{5.1}$$

where MeanRTWP is mean Received Total Wideband Power per cell; MinRTWP is minimum Received Total Wideband Power per cell, used as the noise floor. In other

words MeanRTWP is mean of the power assigned to users, while MinRTWP is the power measured when no users are using cell resources.

Percentage of Downlink Load, also known as DL Carrier Load Percentage (3gpp, 1999) is the ratio between the total transmitter power level on that carrier and the maximum acceptable transmitter power.

$$DL\_LOAD = 100 * 10^{\frac{MeanTCP - MaxTxPower}{10}} \tag{5.2}$$

where MeanTCP is mean Total Transmitted Power per cell; MaxTxPower is maximum transmit power of the cell. In other words MeanTCP is mean of the power assigned to users, while MaxTxPower is the cell power capacity.

We used two additional measures for downlink load, based on code capacity, namely Code Utilization and Count of "No Code Available" Situations. 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. For example, voice calls require 12.2 Kbps (translates into 2 codes), while Data Sessions can require up to 14.4 Mbps (which would consume all the codes of that cell).

Code Utilization Measures the fraction of codes used vs. codes available at the cell. It is averaged over an hour.

Count of "No Code Available" Situations -After all the codes have been assigned, the next device that requests a code from the cell, gets a "no code available" message and cannot use the cell resources. This variable measures the count of occurrences of this message per hour, and will be abbreviated as NCA.

5.2.2 Input Parameters

In Table 5.1 we provide a list of input parameters, as well as the description for each parameter we used for modeling. All these variables are measured per hour. 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 cell resources. Forecasts for future values of the input parameters were available at the operator. In our earlier research on this topic (Radosavljevik et al., 2012), we used additional measures from the network management tool Nokia Data Warehouse. However, due to constraints mentioned in Subsection 5.2.1, namely different performance management tools from different vendors, not all of these could be measured. Therefore, in comparison with our previous research, we reduced the input parameter set by excluding the following measures: 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, Average Voice Call Users, Maximum HSUPA users, Maximum HSDPA users and Total Active RABs, as they could not be measured.

Table 5.1: List of Input Parameters

Variable	Description
Average Count of Release 99 Uplink users	Average number of users that consumed uplink cell resources on a R99 capable device (up to 384 Kbps).
Average Count of Release 99 Downlink users	Average number of users that consumed downlink cell resources on a R99 capable device (up to 384 Kbps).
Average Count of HSUPA users	Average number of users that consumed uplink cell resources on a HSUPA (High Speed Uplink Packet Access) capable device (up to 5.76 Mbps).
Average Count of HSDPA users	Average number of users that consumed downlink cell resources on a HSDPA (High Speed Downlink Packet Access) capable device (up to 14.4 Kbps).[a]
Count of RRC attempts	Radio Resource Control (RRC) attempts are related to the signaling exchange between the mobile device and the network cells. There can only be one RRC connection open per mobile device at a time.
Count of Data Session RAB Attempts	Radio Access Bearer (RAB) is necessary to be assigned to a user in order to make voice call or a data session. Multiple RABs can be assigned to the same device. This variable measures the RAB attempts (not necessarily successful) for a data session in a cell in an hour.
Count of Voice Call RAB Attempts	This variable measures the RAB attempts (not necessarily successful) for a voice call in a cell in an hour. It is the only variable that addresses usage of voice services exclusively.
Average Downlink Throughput	Average per hour of the sum of downlink bandwidths consumed by all users served by the cell.
Average Uplink Throughput	Average per hour of the sum of uplink bandwidths consumed by all users served by the cell.

[a] Most of the current mobile devices are HSDPA capable. Theoretically, even higher speed scan be achieved for both HSUPA and HSDPA. But, an HSDPA device can also be assigned to a R99 downlink (slower) channel, if there are no HSDPA cell resources available.

5.3 Approximating the Load

Traditional approaches for mobile network load forecasting are most often analytical or Monte Carlo based approaches (Mäder and Staehle, 2004). However, the inputs required are difficult to measure and forecast, or do not relate to customer behavior such as call usage which is easier to understand and use for scenario purposes.

Most of the data mining literature on load forecasting is related to electrical networks. A good overview is presented in Feinberg and Genethliou (2010). 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 where 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 Svoboda, Buerger and Rupp (2008), comparing time series (moving averages and dynamic harmonic regression) with linear and exponential regression. Also, Support Vector Regression was used by Bermolen and Rossi (2009) 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 5.2). Therefore, the load parameter (output) can be expressed as $L = f(x_1, x_2, , x_n)$. 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):

$$L(x) = \sum_{i=0}^{m} \sum_{j=1}^{n} User_i * Resource_j + \sum_{y=1}^{Z} interference(x, y) \tag{5.3}$$

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 different load parameters (outputs) and input parameters described in Section 5.2, on an hourly basis during 1,5-8 weeks, depending on the operator. This provided approximately between 200 and 1,000 instances per cell or 20,000,000 instances in total on a network of 20,000 cells.

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

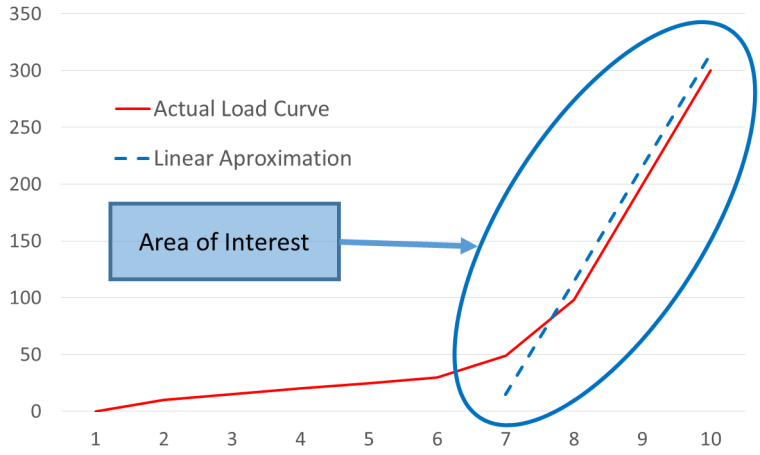

Figure 5.1: Actual Load vs. Linear approximation

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.

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 concept of load pockets explained by Feinberg and Genethliou (2010).

The choice of linear regression (Witten and Frank, 2005) was made due to several reasons. First of all, even though the distribution of the values of each of the load measures we were trying to predict varied between close to linear and close to exponential, we were only interested in the higher values of the load curve, and this can be approximated quite well with linear regression, as shown on Figure 5.1. For this purpose, before constructing the regression formulas, we removed 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 in most cases (Figure 5.1).

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 (van der Putten and van Someren, 2004). 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 allowed non data miners to simulate future network load based on changes in the various types of network traffic, using simple tools they are familiar with.

5.4 Building the Load Formulas

In this section we will describe how the models were 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, the process of forecasting the future load and additional simulation scenarios.

5.4.1 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.

Radio Network Performance Management System. As stated above, this research was using data from four different operators of the Deutsche Telekom group. Most of them had radio networks produced by different vendors, which meant that also different Radio Network Performance Management Systems were used for data collection of both the input and the output parameters. In this research, we used Performance Management Systems of Nokia (Nokia Siemens Networks, 2008), Ericsson (Ericsson, 2013), Huawei (Huawei, 2013) and MyCom (MyCom, 2013), depending on the operator. These software tools were already a part of the Network/IT infrastructure of the operators. They contain technical parameters related to the mobile network performance. The most important feature of these tools for our research was that they contained hourly aggregates of all the input and output parameters we used (Section 5.2). These are the only domain specific tools from our process.

Load Prediction and Simulation Data Mart. This is an Oracle Database 10g-64 bit v10.2.0.5.0 (Oracle, 2011) used for all our task specific data preparation and manipulation.

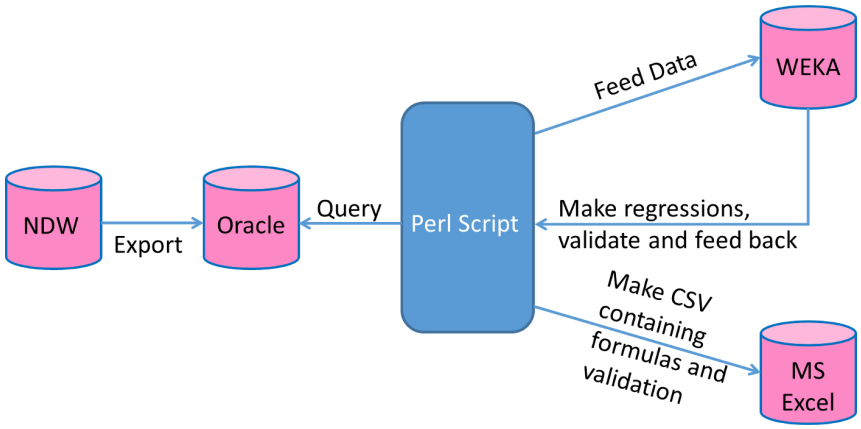

Figure 5.2: Communication Graph of the Tools used

Due to the fact that the necessary input and output parameters were stored at different tables in the respective Performance Management Systems, we needed a separate database where we could manipulate the data easier (e.g. merge tables, create indexes, and build the final flat table). In the case of Nokia Data Warehouse (Nokia Siemens Networks, 2008) this reduced the duration of the data collection and data preparation process from two weeks to one day by productizing data collection. Because we were 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 (Hall et al., 2009), an open source data mining platform, was used for building the linear regression formulas and validating them. Of course, any other tool capable of deriving linear regression could also be used for this purpose. That said, our approach showed that even a research focused open source tool like WEKA can be used in critical commercial settings, at high complexity (e.g. 20.000 cells, 4 models each, around 1000 instances each).

Strawberry Perl for Windows v5.12.3 (Strawberry Perl, 2011) is an open source scripting language which we used in order to create the script that is the core of this approach. Our script created WEKA input files by querying the Oracle database, generated the regression models by executing calls to WEKA, and stored 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 (Microsoft Corporation, 2010), 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 built using scaling factors based on handset/Internet usage developments (internal to the operator).

5.4.2 Process Description

A graph of how our approach used these tools to derive and store the regression models is presented on Figure 5.2. First, the data was extracted from the Network Performance Management Tools, e.g. Nokia Data Warehouse (NDW). The core of our approach is a Perl (Christiansen and Torkington, 2003) script that automated the derivation of regression formulas for each cell. This script executed calls to WEKA and queried the Oracle Database. It works in the following manner:

1. *Get list of cells from the database*
2. *For each cell*
   - 2.1 *Run a query on the database to isolate only the data related to that cell (all the input and output parameters).*
   - 2.2 *Make separate files for each of the load output parameters*
   - 2.3 *For each of the load output parameters*
     - 2.3.1 *Filter out all instances where the load is 0*[1].
     - 2.3.2 *Select only relevant variables for the regression formula of that cell, using a wrapper approach*
     - 2.3.3 *Build the linear regression formula and store it in a separate file.*
     - 2.3.4 *Use 10-fold cross-validation to validate the model.*
     - 2.3.5 *Store the formula, the number of instances used to build the regression formula, 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 formulas, we used a wrapper (Kohavi and John, 1997) approach. Wrapper approaches automatically select the best variables for predicting the outcome, taking into account the algorithm to be used, which in our case was linear regression.

Wrapper approaches do not necessarily perform better or worse than filter approaches (Tsamardinos and Aliferis, 2003). 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 (Witten and Frank, 2005). 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 provided a good estimate of the accuracies of these formulas. Of course,

[1] We did not want noisy data. Cells/instances with no load are of no interest.

we did test them on completely new data sets, 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.

5.4.3 Results and Discussion

Using this process we were able to run 30,000 regressions per day, by just one click. This does not necessarily result in 30,000 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 outcome variables were rare. Furthermore, cells that did not show any load by the means of the output variables were not of interest for our problem situation. For practical purposes, we will only present the modeling results for two of the four output variables we used to describe the air interface load in Section 5.2. We chose to present the results for the uplink and downlink load. All tables have the same structure. In the first column we list bands (intervals) of the output variables Downlink Load and Uplink Load, respectively. The second column contains the count of cells that fall into the respective bands. The third column presents the average count of non-zero instances (NZI) in each band. 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 presents the average Correlation Coefficient (CC) between the predicted and actual values of the 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 of the Regression Modeling for Downlink and Uplink load for four different countries are shown in Tables 5.2-5.9. In Tables 5.2 and 5.3 we present the modeling results of T-Mobile Netherlands, the same operator published in Radosavljevik et al. (2012). However, the operator was still undergoing a full network swap during our research, which means every cell in the network was either already replaced or about to be replaced by a new one from a different network equipment vendor. At the moment of research, this operator was running both networks in parallel, which created an additional level of complexity. The results presented in Tables 5.2 and 5.3 are referring to the modeling process on the swapped part of the network using the new vendor's equipment. Hence, the total number of cells is smaller than reported in Radosavljevik et al. (2012). For this reason, and the fact that we are presenting different output variables in this chapter, the results of Radosavljevik et al. (2012) and these results should not be compared.

Table 5.2: Regression Modeling Results for Downlink Load (DL) for Country Operator 1

Downlink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$DL < 1$	642	/	/	/
$1 <= DL < 5$	132	504.2	0.906	99%
$5 <= DL < 10$	450	528.5	0.92	100%
$10 <= DL < 20$	2995	507.7	0.914	100%
$DL >= 20$	4120	511.1	0.955	100%

Table 5.3: Regression Modeling Results for Uplink Load (UL) for Country Operator 1

Uplink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$UL < 1$	643	579	0.254	1%
$1 <= UL < 5$	693	513.94	0.536	96%
$5 <= UL < 10$	2880	522.06	0.676	100%
$10 <= UL < 20$	3405	516.14	0.756	100%
$UL >= 20$	718	503.05	0.776	99%

Table 5.4: Regression Modeling Results for Downlink Load (DL) for Country Operator 2

Downlink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$DL < 1$	683	/	/	/
$1 <= DL < 5$	2379	155.4	0.777	80%
$5 <= DL < 10$	9406	247.9	0.824	96%
$10 <= DL < 20$	4550	271.7	0.846	95%
$DL >= 20$	1697	284.7	0.872	92%

Table 5.5: Regression Modeling Results for Uplink Load (UL) for Country Operator 2

Uplink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$UL < 1$	431	172.8	0.554	17%
$1 <= UL < 5$	138	247.5	0.649	84%
$5 <= UL < 10$	909	273.9	0.565	87%
$10 <= UL < 20$	9649	294.7	0.617	95%
$UL >= 20$	7816	288.8	0.801	98%

Table 5.6: Regression Modeling Results for Downlink Load (DL) for Country Operator 3

Downlink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$DL < 1$	387	/	/	/
$1 <= DL < 5$	2447	142.5	0.854	84%
$5 <= DL < 10$	2428	155.9	0.907	95%
$10 <= DL < 20$	2114	175.5	0.935	99%
$DL >= 20$	746	184.6	0.945	100%

Table 5.7: Regression Modeling Results for Uplink Load (UL) for Country Operator 3

Uplink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$UL < 1$	160	56.6	0.382	11%
$1 <= UL < 5$	715	120.6	0.482	77%
$5 <= UL < 10$	1909	146.4	0.546	92%
$10 <= UL < 20$	3195	162.4	0.677	98%
$UL >= 20$	2143	171.2	0.668	96%

Table 5.8: Regression Modeling Results for Downlink Load (DL) for Country Operator 4

Downlink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$DL < 1$	5	952.4	0.449	100%
$1 <= DL < 5$	604	949.7	0.705	100%
$5 <= DL < 10$	3801	958	0.776	100%
$10 <= DL < 20$	4802	957.9	0.807	100%
$DL >= 20$	1020	958.2	0.848	100%

Table 5.9: Regression Modeling Results for Uplink Load (UL) for Country Operator 4

Uplink Load	Count of Cells	Avg Count of NZI	Avg Correlation Coefficient	Models Built vs. Number of Cells
$UL < 1$	0	/	/	/
$1 <= UL < 5$	0	/	/	/
$5 <= UL < 10$	674	948.2	0.461	98%
$10 <= UL < 20$	4187	955.5	0.598	100%
$UL >= 20$	5371	957.2	0.669	100%

The results can be evaluated by using two criteria: the Average Correlation Coefficient and the Ratio of The Models Built (the last two columns in Tables 5.2 to 5.9). The Ratio of the Models built for both Downlink Load and Uplink Load grew alongside the number of non-zero instances for operators in all four countries, which was to be expected. We chose the Correlation Coefficient because it is a relative measure and therefore more intuitive than the Mean Average Error or the Root Mean Square Error. The Correlation Coefficient was also much easier to explain to the end users than the latter two error measures. As mentioned above, we report the Average Correlation Coefficient of each load band. The confidence intervals for the Average Correlation Coefficient at 95% confidence level were not wider than ± 0.02 for any operator in any of the Downlink or Uplink load bands, due to the relatively low standard deviations.

Furthermore, 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 Average Correlation Coefficient between actual and predicted values also grows as the load is higher, both for Downlink and Uplink Load. In the lowest load bands, the performance of the models is not good. However, this was of no interest, as these were not the situations that we were trying to predict. These cells were not likely to be in overload in the foreseeable future.

However, when analyzing the Average Correlation Coefficient (ACC) between the predicted and actual values there is a visible difference between Downlink and Uplink load: The ACC for Downlink Load (Tables 5.2, 5.4, 5.6, 5.8) was much higher than the ACC for Uplink Load (Tables 5.3, 5.5, 5.7, 5.9). Uplink Load seems more difficult to predict using linear regression. There are two possible reasons for this: A crucial input parameter (predictor) may be missing; or the Uplink Load has less of a linear nature.

Last but not least, model performance across operators cannot be compared due to differences in network vendors, software versions, geography, population density and smartphone penetration rates (which cause higher network load). The importance of the smartphone penetration and population density was also confirmed by the automated feature selection, where variables such as the combined throughput (uplink or downlink), which is highly influenced by smartphones and the number of HSDPA/HSUPA users per hour (which are smartphone users) were the most often selected when building the respective load formulas.

5.4.4 Forecasting the Load

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

scope of this research). A cell was marked for upgrade[2] if any of the four output variables used as measures of Uplink or Downlink load, was above a predefined critical value.

This is done in the following way:

1. *For each output variable*

   2.1 *For each cell*

   2.3.1 *Select the top 100 instances of the output variable and its corresponding values for the input variables.*

   2.3.2 *Make averages of these input variables.*

   2.3.3 *Scale the input variables up or down, according to scaling factors developed by a traffic model.*

   2.3.4 *Feed the scaled values of the input parameters into the regression formula for the output variable for that cell*

   2.3.5 *If the resulting value is higher than the critical threshold for that output variable, the cell should be upgraded.*

The forecasting model provided better and more sophisticated forecasts and as such supported better network investment decisions, which account for the major part of the entire operator CAPEX cost. In simple terms, no money was wasted by investing in unnecessary network upgrades, providing two benefits: lower cost and redirecting investments into areas that had a higher impact on positive network experience for the customers.

In addition, note that this part of the process was 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. In our view, this approach may be applicable to many other domains.

5.4.5 Applications- Simulation Scenarios

Initially, the only application of this research envisioned by the authors was the deployment scenario for forecasting future load and predicting necessary network upgrades due to "regular" traffic growth, as described in Subsection 5.4.4. This has already been used in four country operators belonging to the Deutsche Telekom group.

However, due to the flexibility of the approach, meaning using simple tools such as Microsoft Excel for implementation, the system developed a life of its own: the

[2]Technically speaking, the network site which generates the cell is upgraded, not the cell itself

end users in T-Mobile Netherlands, the operator where this research was originally developed, started creating simulation scenarios suited for different needs.

Step 2.3.3 of the algorithm described in the previous subsection mentions feeding scaling factors for the input parameters **based on a traffic model**. What if this traffic model was to be replaced by a different one? In that case, a new simulation scenario would be generated. Using our approach, all it takes to generate a new scenario is to change the values of the input parameters in MS Excel. The output of the model (Downlink and Uplink Load) would be automatically recalculated and the user could immediately see the effect. We will explain a few actual use case scenarios in the following paragraphs.

One of the first use cases generated was to predict future network load and evaluate network investments, based on proactive localized marketing campaigning. It was a co-operation between the Marketing and Network Technology Department. The Marketing Department provided their campaign description and expected benefits, namely new customers and increased service usage, which were trended in terms of the input parameters described in Section 5.2. These were fed into the model as described in Subsection 5.4.4, so the increased future load could be predicted and the necessary network improvements can be made, even before the marketing campaign was launched. A very similar scenario was in use for opening new stores, due to the fact that increased number of customers was expected when opening a brick-and-mortar store. This allowed for the network to be prepared to accept the new customers without impacting the experience of the existing ones.

Another very powerful application of this model was evaluating a business case for adding a new wholesale client- or an MVNO (Mobile Virtual Network Operator). This is an operator that does not own a network; instead a MVNO is renting the network of a bigger telecom operator in order to provide services. In this case, the localized traffic growth for predicting the future load was based on the location (or the evaluation of) the customers of the MVNOs and their respective service usage. These were then trended and fed into our model (via MS Excel) in order to evaluate the necessary network improvements, so that no degradation of service for the customers of the host operator would occur. However, these upgrades come at a certain cost, which is attributed to accepting the MVNO onto the host network. If the benefits (revenues) generated by accepting the MVNO are lower than the costs incurred, the business case is negative and therefore, rejected. This approach was used in the country where the research originated to reject a business case for adding an MVNO. Furthermore, a MVNO business case was evaluated for another operator from the same telecom group.

Last but not least, this approach was used as one of the criteria to determine the strategy for the network swap and deployment of LTE (4G) network in T-Mobile Netherlands B.V. As mentioned in Subsection 5.4.3, the operator was undertaking a major network infrastructure investment, namely replacing the entire radio network (every site) in order to modernize it and allow for deployment of 4G. Of course an

undertaking of this size cannot be performed all at once; hence clusters of cells were being planned for replacement at a certain time. Our load prediction method was one of the criteria used for giving priority to certain clusters, thus reducing the need of unnecessary investments into the "old" network. The underlying assumptions here were that the "new" generation radio network would have more efficient resource use and therefore could handle the load better, and that a certain amount of customers would start using 4G services, therefore offloading the 3G network.

5.5 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 sites, which reduce the load of the existing ones, cannot be taken into account in advance. It is necessary to collect a new data set and rebuild the regression formulas, in order to incorporate all these changes into the model. This is why the process described in this chapter was scheduled for execution every 3-4 months.

Next, we intend to improve the predictions for uplink load. One method would be searching for additional input parameters to improve the performance of predicting uplink load using linear regression. Alternatively, we could look for a substitute for linear regression better suited for modeling uplink load on the cells where linear regression does not deliver. However, this algorithm should not substantially slow down the whole process and must be easily transferable to MS Excel, in order to keep the flexibility and the ease of building simulation scenarios.

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. However, it should be noted that there are a lot of factors impeding a direct comparison. As stated 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 (Caruana, 1997). 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 an exception for a non-select small number of cells. Next, 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.

Furthermore, we do intend to investigate additional simulation scenarios for our approach, beyond those described in Subsection 5.4.5. Last but not least, this research has been implemented in four operators of this telecom group. Other operators from the same group are planned to follow, with their own use cases and applications.

5.6 Conclusions

In this chapter we presented a very simple yet effective approach of deploying 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 linear regression, because it is easier to understand them. Once these simple approaches gain acceptance, and familiarize companies with data mining, opportunities to apply more advanced techniques will arise.

In our result section we showed 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 grew, so did the quality of the model's predictions. We willingly sacrificed the models' performances within cells with very low load, 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 30,000 per day) were 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. At the moment, the commercial tools for this purpose offer only load predictions based on single variable regressions (MyCom, 2013), which is not as robust as our approach.

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 would need to be upgraded every 3-4 months, but this was also scheduled as a part of the 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 allowed 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 Deutsche Telekom, a large telecom operator group with branches in many European countries. Our research was used in four of the countries where this group is present.

Potentially the greatest benefit of our approach is the decoupling of the data mining process from simulation scenarios. This was accomplished by exporting the models into Excel sheets after they have been have been generated by our data mining process. Then, the end users, a team of radio network analysts who are not data miners, were able to use these formulas resulting from a data mining process for forecasting the future network load. This allowed them to simulate multiple traffic scenarios by scaling the current input parameters, which was as simple as changing values in their respective columns in Excel. These scenarios included "regular growth" scenarios, evaluations of network investments necessary to accommodate localized user growth due to targeted marketing campaigns, adding a new wholesale client (an MVNO- Mobile Virtual Network Operator) and prioritizing clusters for deployment of new technologies such as LTE/4G.

Next, we would like to point out the possibility of applying our research onto problems other than telecom network load. This approach would be applicable to any other industry where large scale regression models are necessary. This can be accomplished simply by replacing the data source, in this case the Radio Network Performance Management Tools, with a data source suitable for the industry that would like to apply our research. The decoupling of the data mining process from the simulation scenarios makes our approach more general to situations where detailed simulations are necessary, but the domain experts are not data miners. We already tested this approach for cluster based revenue predictions, which is a topic from the finance domain (see chapter 6).

Last but not least, perhaps one of the most interesting aspects of our approach is the extremely low cost. Given that we used the existing IT infrastructure (Server, Radio Network Performance Management Tools, Oracle, Excel) combined with open source tools (WEKA, Perl), the only costs that incurred were the 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. These amounts are insignificant compared to the actual investments being made into the network.

Addressing the research question posed in section 5.1, we have shown how data mining can be used to predict 3G mobile network interface load, and simulate it under different scenarios. In a nutshell, we have used relatively simple algorithms to create a large number of predictive models, therefore making possible predicting the load

on a cell level. We have used tools known to the end users to deploy these models, allowing them to use different scenarios for the input parameters. Acceptance was gained by decoupling the data mining process from the end users, but keeping the transparency that linear regression offers combined with tooling familiar to them.