Spatio-Temporal Ensemble Prediction on Mobile Broadband Network Data

Samulevicius, Saulius; Pitarch, Yoann; Pedersen, Torben Bach; Sørensen, Troels Bundgaard

Published in:
Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th

DOI (link to publication from Publisher):
10.1109/VTCSpring.2013.6692765

Publication date:
2013

Document Version
Accepted author manuscript, peer reviewed version

Spatio-Temporal Ensemble Prediction on Mobile Broadband Network Data

Saulius Samulevičius, Yoann Pitarch, Torben Bach Pedersen
Department of Computer Science
Aalborg University
E-mail: {sauliuss, ypitarch, tbp}@cs.aau.dk

Troels Bundgaard Sørensen
Department of Electronic Systems
Aalborg University
E-mail: tbs@es.aau.dk

***Abstract*—Facing the huge success of mobile devices, network providers ceaselessly deploy new nodes (cells) to always guarantee a high quality of service. Nevertheless, keeping turned on all the nodes when traffic is low is energy inefficient. This has led to investigations on the possibility to turn off network nodes, fully or partly, in low traffic loads. To accomplish such a dynamic network optimization, it is crucial to predict very accurately low traffic periods. In this paper, we tackle this problem using data mining and propose Spatio-Temporal Ensemble Prediction (STEP). In a nutshell, STEP is based on the following two main ideas: (1) since traffic shows very different behaviors depending on both the temporal and the spatial contexts, several prediction models are built to fit these characteristics; (2) we propose an ensemble prediction technique that accurately predicts low traffic periods. We empirically show on a real dataset that our approach outperforms standard methods on the low traffic prediction task.**

I. Introduction

To sustain the rapidly increasing demand of mobile services, network operators have adopted a densification strategy by ceaselessly setting up new network equipment. While this strategy is necessary for maintaining the minimum quality of service (QoS) during the peak periods, it most often leads to a excess use of energy when traffic is normal or low. Two arguments motivate this statement. First, dense deployment of nodes (cells) creates overlapping areas in the network, *i.e.*, within these areas, multiple nodes can serve the same users. During the high traffic hours nodes are used efficiently. However when the number of active users and the traffic goes down, some of the nodes are operating at low utilization level. Second, it has been recently shown in [1] that traffic load has small impact on the total energy consumption of an active base station node. Consequently, the potential of energy savings by turning off some nodes depending on predicted traffic demand is high.

Such network optimization strategy would require solving two complementary open research problems. The first problem is to very accurately predict low traffic periods at the node level. As it is experimentally shown later, the standard statistical technique, *e.g.*, ARIMA, fail in achieving this task. Second, once some nodes have been elected as candidates for being turned off at a certain moment, it is crucial to maintain a trade off between two contradictory goals: saving as much energy as possible while maintaining satisfactory QoS. Indeed, reducing the number of operating network nodes during low traffic period would lead to an increased utilization level and impact on the QoS for the remaining nodes.

In this paper, we investigate the first problem: what data mining can do to tackle the low traffic prediction problem, and propose STEP (Spatio-Temporal Ensemble Prediction), an accurate technique to perform traffic prediction in mobile broadband networks. Our approach mainly relies on two assumptions. First, as stated in [2], traffic varies a lot from cell to cell and from hour to hour. Consequently, instead of considering the complete set of traffic data to compute one single prediction model, it is decomposed into smaller subsets to compute separate models for each cell and each day hour. Second, based on the idea that unity is strength, ensemble techniques encounter a large success to solve classification or prediction tasks. Basically, such approaches combine single models and aggregate their respective predictions to make the final decision. It has been recently shown that these approaches often perform better than single classifiers/prediction models [3]. We thus propose an ensemble prediction model to achieve our goal. Given a training set, an accuracy score is calculated for each model to evaluate how good it is when trained on the given set. Based on these accuracies, a strategy is developed to aggregate these separate predictions. Results obtained on real mobile broadband network traffic data show that our approach performs better than a state-of-the-art approach, ARIMA.

The remainder of this paper is structured as follows. Section II introduces some background knowledge. In Section III we formally describe STEP and illustrate it on an example. Section IV provides some experimental results and Section V draws the conclusion of this work and open some future directions.

II. Background

A. Toward a Dynamic Optimization Strategy

We illustrate steps in the dynamically optimized network through the example provided in Fig. 1. A dashed line represents a coverage area within which users are served during the *busy* hours, while the solid line represents possible coverage area when the traffic is *low*. Users in the network are denoted as U_1,. . ., U_n. The network is constructed by nodes C_1,. . .,C_k. To optimize the energy consumption, we consider two node states: On (the node is broadcasting) and Off (the node is turned off), *e.g.*, if we predict *low* traffic from 2AM to 6AM,

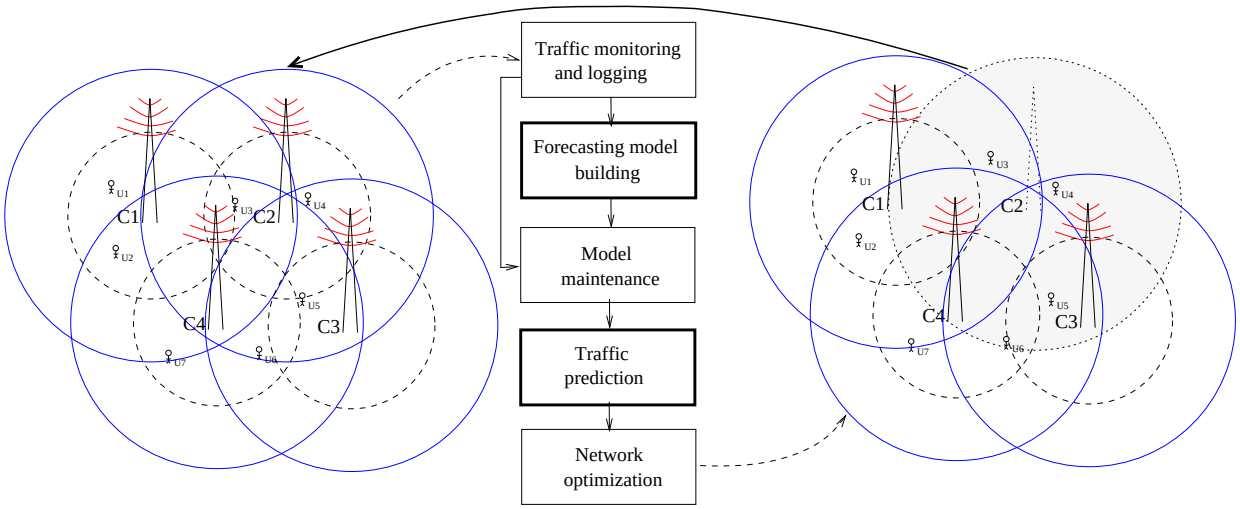


Fig. 1. Network optimization strategy at 3AM (left - not optimized, historically monitored network data, right - network with optimization)

the node can be turned off at 2AM and should be turned on at 6AM.

To clarify the process of network optimization, let us look at 3AM, node C_2 in Fig. 1. Historical data shows that C_2 was serving low traffic at 3AM, the neighbor nodes C_1, C_3, and C_4 were turned on serving low traffic as well. Assume the prediction for C_2 at 3AM will be low (the node will serve only few users). Intuitively C_2 could be turned off, as the coverage area of C_2 can be served by the neighbor nodes C_1, C_3, and C_4. In this paper, we propose an efficient solution for predicting periods (see Fig. 1) where a subset of the nodes can be potentially turned off. The study of selecting the optimal subset for achieving the best energy savings while maintaining the minimum QoS guaranties is left for further studies.

B. Prediction

Prediction methods are developed to give estimates of the future. Schematically, prediction is a multiple-step process. During the first *learning step*, a model based on the historical data is built and the model parameters are estimated. This model is then used during the second *prediction step*. In case of changing distribution the third *model evaluation step* enables the model recomputation whenever the prediction error rate exceeds a certain threshold. In this paper we consider a few different techniques used for time series predictions: ARIMA (AutoRegresive Integrated Moving Average) [4], EGRV (Engle, Granger, Ramanathan, and Vahid-Arraghi model) [5] and ensemble methods [3].

ARIMA is a popular time series forecasting technique. It has been adapted in various research areas, *e.g.*, stock, natural gas, IP traffic prediction, etc. Due to variety of use cases, ARIMA is selected as the baseline technique.

Initially EGRV was designed for the short-term forecasting of electricity demand. EGRV is based on building multiple regression models. A model is built for each hour of the day, *e.g.*, a model for 1AM is built using historical data for 1AM, and so on for 2AM,... Predictions are made using the model built for a specific hour, *e.g.*, to predict traffic at 1AM the model built for 1AM is used. We do not evaluate EGRV predictions with our data since it is not designed to fit with MBN data.

The EGRV splitting strategy fundamentally differs from the ARIMA. Indeed, whereas ARIMA use the whole training set to learn a single predictive model, EGRV relies on its subdivision and multiple prediction models. For instance, let us consider Figs. 2 and 3, which confront how data are selected for prediction model building. In Fig. 2, a time series

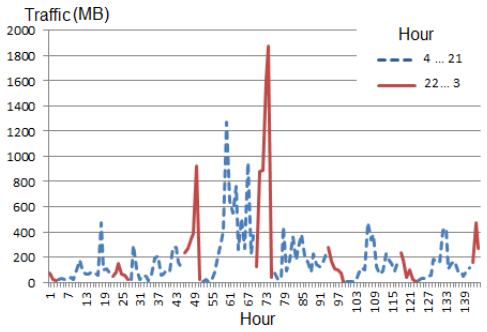


Fig. 2. ARIMA time series (training data 144 hours)

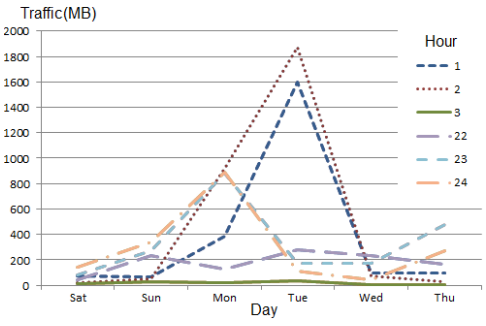


Fig. 3. EGRV time series (6 single hour models)

representing the traffic for one cell for six consecutive days is displayed. This time series directly serves as input to ARIMA to compute *one* prediction model that will be used to predict traffic values. To show the differences between EGRV and ARIMA, 6 consecutive hours (22, 23, 24, 1, 2, 3) have been selected. These selected 6 hours in the ARIMA model, see Fig. 2, are marked as solid lines, while in EGRV 6 models (one for every hour) are constructed, see Fig. 3.

Ensemble prediction is based on taking several different prediction models and using them to build one ensemble model. The analysis of data allows selecting the most accurate methods. Intuitively most of the methods would predict different values, therefore the same most common prediction is taken. Combining prediction methods and selecting the most common predicted value provides higher confidence and better results compared to predictions from one individual method.

C. Data Description and Evaluation Metrics

Data Description. We denote by $\mathcal{D}_T^{Raw} = d_1^{Raw}, \ldots, d_n^{Raw}$ the training set used to compute prediction models. Each d_i^{Raw} (with $1 \leq i \leq n$) is in the form $d_i^{Raw} = (cellId_i, , date_i, hour_i, traffic_i)$. In this paper, we are interested in predicting if traffic will be either *low* or *high* rather than predicting the absolute traffic value. For this reason, we introduce a user-defined threshold, denoted by σ, to discretize each $traffic_i$ into the set of classes $\mathcal{C} = \{high, low\}$. Fig. 4 illustrates such a discretization strategy. The extended training set is denoted by $\mathcal{D}_T = d_1, \ldots, d_n$ such that each d_i (with $1 \leq i \leq n$) is in the form $d_i = (cellId_i, , date_i, hour_i, traffic_i, class_i)$ (with $class_i \in \mathcal{C}$).

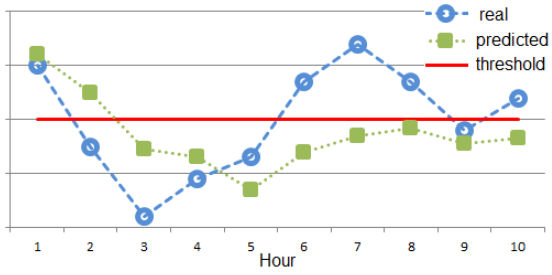


Fig. 4. Illustration of the traffic value discretization strategy

Evaluation Metrics. We now introduce the evaluation metrics we used to measure the effectiveness of STEP.

- The *global accuracy* represents the ratio of correct predictions over the total number of performed predictions. Considering the example displayed in Fig. 4, the global accuracy is $\frac{5}{10}$ (*i.e.*, 50%).
- The *utility* captures the number of hours when the prediction is correct and traffic is classified as *low*, *e.g.*, utility equals 4 in Fig. 4. Intuitively, this measure represents the cumulated number of hours when nodes can be safely turned off.
- Based on the utility measure, the *potential energy savings* can be calculated. For that we consider the energy consumption model for a base transceiver station (BTS) described in [1]. The energy required to power a node is defined as P_{BTS} and the cost for turning on and off a BTS is defined as P_{oper}. The length of potentially *low* traffic period when a BTS could be turned off is denoted as T_{low}. Energy savings $P_{savings}$ at a single BTS for a *low* traffic period can be calculated using equation 1.

$$P_{savings} = P_{BTS} * T_{low} - P_{oper} \quad (1)$$

- The *local precision* [6] is similar to the global accuracy but for a given class. For instance the precision of the *low* class in Fig. 4 is $\frac{4}{8}$ since the *low* class has been *predicted* 8 times, *i.e.*, hours 3, 4, 5, 6, 7, 8, 9 and 10, and *was correct* 4 times, *i.e.*, hours 3, 4, 5 and 9. Intuitively, the *local precision* measures how accurate we are at predicting the selected class.
- Given a class, the *local recall* [6] represents the ratio of correct predictions on this class over the number of elements which really belong to this class. For instance the recall of the *low* class in Fig. 4 is $\frac{4}{5}$ since the *low* class has been measured 5 times, *i.e.*, hours 2, 3, 4, 5 and 9, and prediction was correct 4 times, *i.e.*, hours 3, 4, 5 and 9. Intuitively, the *local recall* measures the ability of capturing all the elements which belong to the selected class.

III. Spatio-Temporal Ensemble Prediction (STEP)

Our approach relies on two main principles: splitting the training set spatially and temporally to isolate distinct behaviors and applying an ensemble strategy to predict *low* or *high* traffic periods. These ideas are described separately in the following two subsections. Finally, a detailed example is used to illustrate our proposal.

A. The spatio-temporal splitting strategy

As stated in [2], MBN traffic shows very different behaviors from cell to cell and from hour to hour. This observation motivates us to adopt *splitting* strategy inspired by the EGRV. We have empirically observed in [2] that traffic values mostly depend on both $cellId$ and $hour$ attribute values. $\mathcal{D}_T$ is thus split into smaller subsets, denoted by $\mathcal{D}_T[x, y]$, according to these attributes values. Formally, $\mathcal{D}_T[x, y]$ is defined by $\mathcal{D}_T[x, y] = \{d_i \,|\, cellId_i = x \text{ and } hour_i = y\}$. Each of these subsets will serve as a training set to compute the prediction model associated with the *context* $[x, y]$. Consequently, when prediction needs to be performed, the first step is to identify which model should be used depending on the values of attributes *cellId* and *hour*, *e.g.*, prediction of the traffic class of $cellId = 1$ at 1AM will be based on the model trained with $\mathcal{D}_T[1, 1AM]$.

B. The prediction strategy

Ensemble techniques are known for successfully solving classification and prediction tasks. Basically, such approaches combine single models and aggregate their respective predictions to make the final decision. It has been recently shown

that these approaches often perform better than single classifiers/prediction techniques [3]. For this reason, we adopt such a strategy to tackle the problem of traffic prediction. In this paper, four simple models *minimum (MIN)*, *maximum (MAX)*, *average (AVG)* and *median (MED)* are built. Nevertheless, despite this apparent simplicity, our methodology to aggregate separate predictions obtains very good results compared to the state-of-the-art as shown in the next section.

In the rest of this subsection, we consider $\mathcal{D}_T[x, y]$ as the training set and describe how the ensemble prediction model is built. Essentially, this construction relies on two steps. First, a score for each single model $M_i \in \{MIN, MAX, AVG, MED\}$, denoted by $Acc(M_i, \mathcal{D}_T[x, y])$, is calculated. Intuitively, this measure evaluates how good M_i is to predict elements in $\mathcal{D}_T[x, y]$. To do so, a cross-validation strategy is adopted [7]. In a nutshell, each element $d_i \in \mathcal{D}_T[x, y]$ is first isolated. M_i is then trained on the rest of the training set and prediction is finally compared with the *real* value. The accuracy reflects the ratio of the good predictions in this iterative process. Formally, $Acc(M_i, \mathcal{D}_T[x, y])$ is defined as:

$$Acc(M_i, \mathcal{D}_T[x, y]) = \frac{\sum_{d_i \in \mathcal{D}_T[x,y]} |M_i(\mathcal{D}_T[x, y] - d_i) = class(d_i)|}{|\mathcal{D}_T[x, y]|}$$

such that $M_i(\mathcal{D}_T[x, y] - d_i)$ is the class predicted by M_i trained on $\mathcal{D}_T[x, y] - d_i$, $|M_i(\mathcal{D}_T[x, y] - d_i) = class(d_i)|$ equals 1 if the prediction is correct and 0 otherwise, where $|\mathcal{D}_T[x, y]|$ is the number of elements in the training set.

Once accuracy scores have been calculated for the four considered models, their respective predictions need to be aggregated. For doing so, the following strategy is adopted. The class predicted by the model showing the highest accuracy is chosen. In case several single models have the same highest accuracy but different classes, the majority rule strategy is applied, *i.e.*, the outputted class is the majority one that has been predicted by the models with the highest accuracy. Finally, if none of these two rules can be applied, *i.e.*, our approach is not able to precisely determine if the node can be turned off at this time, a *safe* policy is adopted and the predicted class will be *high*.

C. Illustrative example

To illustrate the above-described methodology, let us consider a small dataset presented in Fig. 5 which represents the traffic associated to $cellId = 1$ for the six previous days and the current time (dashed line) - Thursday at midnight. We assume that the threshold equals 200, *e.g.*, on Monday at 1AM, traffic is *high* whereas it is *low* on Thursday at 1AM, and we aim at predicting if traffic will be either *low* or *high* on Friday at 1AM for this cell. The subset used for training models, *i.e.*, $\mathcal{D}_T[1, 1AM]$, is highlighted by the box in Fig. 5.

The accuracy score calculation process for the model *MED* is detailed in Table I. In the first line, we evaluate if the traffic associated with Thursday at 1AM (the second column) can be predicted using the five remaining elements (the first column). The result of the *MED* calculation is displayed in column 3. Since both real and median traffics are below the threshold,

	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	
1	74	64	384	1601	100	98	?	?	⋯
2	18	54	922	1874	73	25	?	?	⋯
3	15	25	19	38	8	7	?	?	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
22	44	233	128	281	231	162	?	?	⋯
23	82	269	881	170	172	476	?	?	⋯
24	146	343	888	111	42	270	?	?	⋯

Fig. 5. Network traffic data (Day, Hour, traffic MB)

$|MED(\mathcal{D}_T[1, 1AM] - d_i) = class(d_i)|$ equals 1. By repeating this process, we obtain $Acc(MED, \mathcal{D}_T[1, 1AM]) = \frac{4}{6}$. This process is repeated for the four models. The respective predictions and accuracies are displayed in Table II.

TABLE I
ACCURACY CALCULATION

$\mathcal{D}_T[x, y] - d_i$	d_i	MED	$\lvert MED(\mathcal{D}_T[1, 1AM] - d_i) = class(d_i)\rvert$
74, 64, 384, 1601, 100	98	100	1
74, 64, 384, 1601, 98	100	98	1
74, 64, 384, 100, 98	1601	98	0
74, 64, 1601, 100, 98	384	98	0
74, 384, 1601, 100, 98	64	100	1
64, 384, 1601, 100, 98	74	100	1
$Acc(MED, \mathcal{D}_T[1, 1AM])$			4/6

TABLE II
ENSEMBLE METHOD ACCURACY, HOUR 1

Model	Prediction	Accuracy
MIN	Low	4/6
MAX	$High$	2/6
AVG	$High$	1/6
MED	Low	4/6

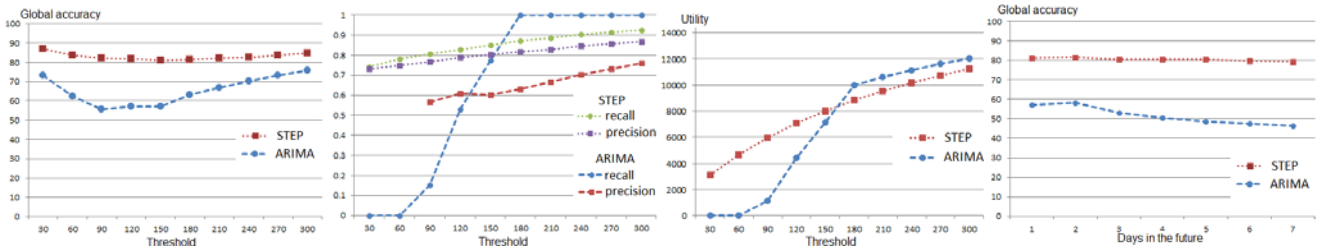
The results provided in Table II show that models *MED* and *MIN* share the highest accuracy. Since both models predict *low* traffic for the $cellId = 1$ at 1AM, STEP will predict *low* for this spatio-temporal context.

IV. EXPERIMENTAL STUDY

A. Dataset Description

To evaluate STEP, we used real network data from a major EU city. 660 mobile broadband network nodes deployed in 229 sites are used for the building models and the predictions. The mobile network is monitored for 46 consecutive days during the 1st quarter of 2012. Date, time, network node name and the sum of traffic in megabytes within one hour are used.

We now compare our proposed STEP with ARIMA for different threshold values. ARIMA(1,1,3) model building and predictions were performed using R programming language. The threshold for the evaluation period is selected taking the minimum and the maximum traffic values and splitting in



(a) (b) (c) (d)

Fig. 6. (a) Prediction accuracy; (b) Recall and precision for *low* periods; (c) Utility; (d) Accuracies for predictions in the future

intervals size of 30. Half of the data set (first 23 days) is used for model building (STEP and ARIMA). We focus on the lower threshold values (up to 300) since mobile broadband network can be optimized only when the traffic is *low*.

B. Global accuracy

In Fig. 6a, STEP and ARIMA average prediction accuracies are provided for the next 24 hours. As we can see STEP is considerably better for traffic predictions with lower (more realistic) threshold values. For a threshold of 90, STEP shows 26% higher prediction accuracy.

C. Recall and precision metrics

Fig. 6b displays *low* class prediction results. The *low* class precision is important when estimating potential energy savings, as this case is when a cell can potentially be turned off. As we can see, potential savings for STEP are always higher than for ARIMA for all threshold values.

The 100% *recall* for ARIMA for thresholds above 180 is because ARIMA intuitively "gives up" predicting *low* and *high* and just predicts everything as *low*.

D. Utility

We provide utility for the next 24 hours with different threshold values in Fig. 6c. As in [1], we consider that a node carrying *low* traffic consumes $P_{BTS} = 137W$ per hour and the operational cost for the node is zero, $P_{oper} = 0$. The energy savings for different threshold values can now be calculated using equation 1 and the utility.

Considering possible energy savings for the next 24 hours, our proposed STEP gives potential savings of up to 20% (*i.e.*, $P_{savings} = 430kWh$) for a threshold of 30, while ARIMA for the same threshold shows no energy saving possibilities. With higher threshold values, the ARIMA *utility* increases as a result of the "always predict *low*" behavior discussed in IV-C. This is because the utility measure is very simple and does not penalize the wrong *low* guesses of ARIMA. Thus, the actual potential savings for ARIMA would be lower.

E. Prediction validity in the future

We evaluate STEP and ARIMA prediction accuracies for mid-term (7 days) with threshold equal to 150, see Fig. 6d. The figure shows that STEP accuracy has a slight $\sim$2% accuracy decrease over the 7 days, while ARIMA accuracy for the same period decreases more than 10%. Thus, our proposed STEP model is much more stable over longer time periods and requires less maintenance for keeping high accuracy.

V. Conclusion

Accurately predicting low traffic periods in MBNs is the first necessary step to develop a dynamic network optimization strategy. In this paper, we achieved this goal by proposing STEP (Spatio-Temporal Ensemble Prediction) which combines a spatio-temporal data splitting and an ensemble prediction technique. Experimental results showed that (1) STEP for lower thresholds outperforms (by 26% in the best case) the well known forecasting technique ARIMA, and (2) potential energy savings enabled by STEP is at least 20%. Our work could be extended in several directions. First, a more precise utility metric that penalizes wrong *low* guesses should be designed. Second, we are convinced that the prediction accuracy can be increased by incorporating additional knowledge, *e.g.*, frequent patterns, points of interest, and events. Third, once potential candidates have been elected for being turned off, it is challenging to determine which ones can be really turned off while preserving a minimum QoS.

VI. Acknowledgements

The authors would like to thank Telenor Denmark for providing network traffic information for the analysis.

References

[1] G. Micallef, P. Mogensen, and H.-O. Scheck, "Cell size breathing and possibilities to introduce cell sleep mode," in *Wireless Conference (EW), 2010 European*, pp. 111 –115, april 2010.

[2] S. Samulevicius, T. B. Pedersen, T. B. Sørensen, and G. Micallef, "Energy savings in mobile broadband network based on load predictions: Opportunities and potentials," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pp. 1 –5, may 2012.

[3] T. Dietterich, "Ensemble methods in machine learning," *Multiple classifier systems*, pp. 1–15, 2000.

[4] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics, Wiley, 2008.

[5] R. Ramanathan, R. Engle, C. W. J. Granger, F. Vahid-Araghi, and C. Brace, "Essays in econometrics," ch. Short-run forecasts of electricity loads and peaks, pp. 497–516, New York, NY, USA: Cambridge University Press, 2001.

[6] C. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge University Press Cambridge, 2008.

[7] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, IJCAI'95, (San Francisco, CA, USA), pp. 1137–1143, Morgan Kaufmann Publishers Inc., 1995.