On the Feasibility of Deploying Cell Anomaly Detection in Operational Cellular Networks

Gabriela Ciocarlie, Ulf Lindqvist, Kenneth Nitz
SRI International
Menlo Park, California, USA
{gabriela.ciocarlie,ulf.lindqvist,kenneth.nitz}@sri.com

Szabolcs Nováczki
NSN Research
Budapest, Hungary
szabolcs.novaczki@nsn.com

Henning Sanneck
NSN Research
Munich, Germany
henning.sanneck@nsn.com

***Abstract*—The Self-Organizing Networks (SON) concept includes the functional area known as self-healing, which aims to automate the detection and diagnosis of, and recovery from, network degradations and outages. In this paper, we build on our previous work [19] and study the feasibility of an operational deployment of an adaptive ensemble-method framework for modeling cell behavior. The framework uses Key Performance Indicators (KPIs) to determine cell-performance status. Our results, generated using real cellular network data, show that the computational overhead and the detection delay are sufficiently low for practical use of our methods to perform cell anomaly detection in operational networks.**

***Index Terms*—Self-Organizing Networks (SON), cell anomaly detection, Self-Healing, performance management, Key Performance Indicators**

I. Introduction

To meet the expectations for virtually "unlimited" capacity and "ubiquitous" coverage of cellular networks, Self-Organizing Networks (SON) [1] provide increased automation of network operations with optimized resource utilization. Automated features need to be properly integrated with the existing operator processes and embedded into the legacy Operation, Administration and Maintenance (OAM) architecture. Among different components, the SON architecture includes configuration, optimization, and troubleshooting capabilities that aim to satisfy self-configuration, self-optimization, and self-healing requirements.

In this paper, we focus on the feasibility for deployment of self-healing capabilities, which reduce the operator effort and the outage time to provide faster maintenance. Specifically, the problem that we study is the feasibility of operational deployment of cell anomaly detection. In previous work [19], we used Key Performance Indicators (KPIs), which are highly dynamic measurements of cell performance, to determine the state of a cell. Moreover, we employed techniques that can cope with concept drift, which is defined as the phenomenon where the normal behavior of the system legitimately changes over time (e.g., by the increasing amount of user-induced traffic demand). Previous results, generated using real cellular network data, suggested that the proposed ensemble method automatically and significantly improves the detection quality over univariate and multivariate methods while using intrinsic system knowledge to enhance performance. This paper presents a feasibility study for operational network deployment of the cell anomaly detection framework, while also introducing a new component to the framework: a triggering mechanism to train and age the set of models in the ensemble pool.

978-1-4799-0913-1/14/$31.00

II. Cell Anomaly Detection

Our cell anomaly detection framework [19] aims to determine the relevant features needed for detecting anomalies in cell behavior based on the KPI measurements. Because KPIs are measurements that are collected as ordered sequences of values of a variable at equally spaced time intervals, they constitute a time series and can be analyzed with known methods for time-series analysis. An anomaly in a time series can be either a single observation or a subsequence of a time series with respect to a normal time series. We refer to *testing* as the comparison of a set of KPI data to a model of the normal state established by an earlier observed set of KPI data referred to as *training* data. *Ground truth* is defined as the labels associated with the data points that indicate whether the data represents a real problem or not.

The main hypothesis is that no single traditional time-series anomaly detection method (classifier) could be able to provide the desired detection performance. This is due to the wide range in the types of KPIs that need to be monitored, and the wide range of network incidents that need to be detected. Consequently, an ensemble method, which combines different classifiers and classifies new data points by taking a weighted vote of their prediction, was proposed [19]. It effectively creates a new compound detection method that, with optimized weight parameter values learned by modeling the monitored data, can perform significantly better than any single method. Moreover, the ensemble method can also enable an increased level of automation. Next, we will briefly describe the ensemble-method approach under study, and we refer the reader to more details in our previous work [19].

A. Ensemble Method for Determining KPI Degradation Level

The ensemble-method framework applies individual univariate and multivariate methods to the training KPI data, leading to the construction of a pool of different predictors.

A univariate time series is a time series that consists of single observations recorded sequentially over equal time

increments. Consequently, individual KPIs collected for each cell are univariate time series. We used different methods for modeling the KPI behavior and to test data against the built models which used Empirical Cumulative Distribution Function (ECDF) [6], Support Vector Machine (SVM) [7] and autoregressive, integrated moving average (ARIMA) models.

A multivariate time series is a time series that consists of multiple observed features recorded concurrently over equal time increments. Consequently, the set of all KPIs collected for each cell are considered a multivariate time series. The same type of model proposed for the univariate time series was extended to the multivariate level. Our framework used SVM and Vector Auto-regressive (VAR) models for the multivariate case [8]. Two different implementations of the ARIMA and VAR modeling were used: static "o," in which only one model is created; and dynamic "m," in which multiple models are created over time.

Using the pool of predictors, the predictions obtained on the KPI data under test (i.e., being subject to detection) along with the weights allocated to each predictor lead to the computation of the KPI degradation level (i.e., the deviation of a KPI from its normal state). The proposed methods rely on context information (available for cellular networks) extracted from human-expert knowledge, Configuration Management (CM) data, or confirmed Fault Management (FM) input data to make informed decisions. We define confirmed FM data as the machine-generated alarms that were confirmed by human operators.

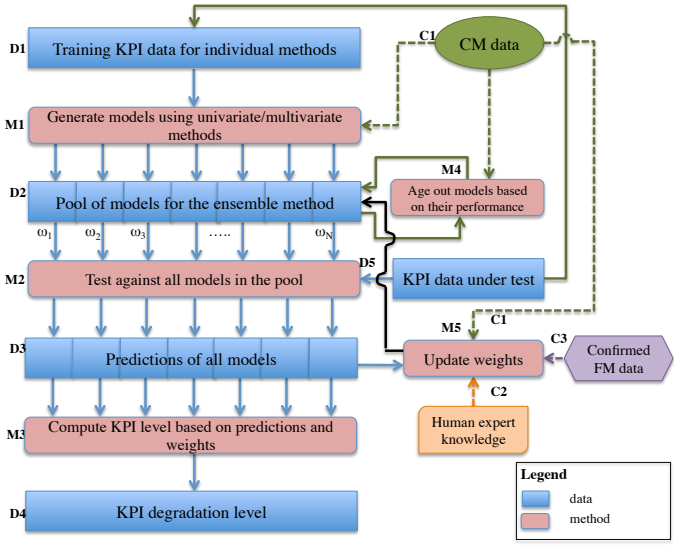

Fig. 1. Overall approach of the ensemble method applied to a single cell in a cellular network [19]. Data is depicted in blue rectangles and methods in pink rectangles with rounded corners. The remaining elements indicate different context information. The dashed lines indicated that an event is triggered in the presence of new evidence/data.

Figure 1 presents the details of the ensemble method under study, where we distinguish between data, methods, context information, and human-expert knowledge. Each cell is characterized by a set of KPI measurements generated as a stream of data. The ensemble method is applied to each cell. The proposed ensemble method implements a modified version of the weighted majority algorithm (WMA) [9] that returns a KPI degradation level in the range [0,1] and that uses the context information for updating the weights and creating new models.

- Given the pool of models (**D2**) trained using univariate and multivariate algorithms (**M1**) on the training dataset (**D1**), the stream of KPIs is used in a continuous fashion as the testing dataset (**D5**). Each model in the pool of models has a weight, ω_i, associated with it. For the initial pool of models, all models have the same weight value assigned ($\omega_i = 1$).
- Any CM change determined automatically (**C1**) triggers the testing dataset to also become the training KPI dataset, after which the method for generating a new set of models (**M1**) is executed. Models are aged using an exponential decay aging mechanism.
- The result of the testing phase (**M2**) is a set of KPI-degradation-level predictions provided by each individual model in the pool of models (**D3**). Some of the predictions are binary (a KPI degradation level of 0 represents normal and 1 represents abnormal), and some have continuous values in the $[0, 1]$ range.
- Human-expert knowledge (**C2**), confirmed FM data (**C3**, and CM change information (**C1**)) trigger the update weights method (**M5**), which penalizes the models in the pool of predictors based on their prediction with regards to the ground truth ($\omega_i \leftarrow \beta * \omega_i$, where $\beta \in [0, 1]$).
- The result of (**M5**) is an updated pool of models (**D2**) with adjusted weights which continue to be used in the testing mode.
- All the predictions in (**D3**) along with the weights associated with the corresponding models are used in a modified weighed majority approach (**M3**) to generate the KPI degradation level, where $\tau \in [0, 1]$ is the threshold that determines whether data is deemed normal or abnormal.

$$q_0 = \sum_{KPI<\tau} \omega_i,\ q_1 = \sum_{KPI\geq\tau} \omega_i \tag{1}$$

The τ value is not dependent on any KPI semantic and can be tuned based on the requirements of the operational environment.

- The result of (**M3**) is the KPI degradation level (**D4**) associated with each KPI measurement of each cell.

$$\overline{KPI\_level} = \begin{cases} \dfrac{\sum_{KPI\geq\tau} \omega_i * KPI\_level_i}{\sum_{KPI\geq\tau} \omega_i}, \text{if } q_1 > q_0 \\ \dfrac{\sum_{KPI<\tau} \omega_i * KPI\_level_i}{\sum_{KPI<\tau} \omega_i}, \text{if } q_1 \leq q_0 \end{cases} \tag{2}$$

III. Deployment Evaluation

This section presents a feasibility study of deploying the ensemble-method in an operational environment. In our pre-

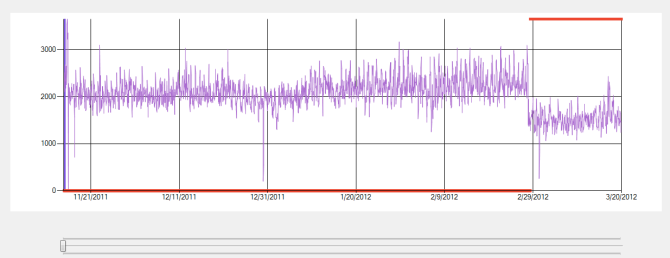

Fig. 2. The visualization tool allows the user to select sections of the data points, which are labeled either normal or degraded (see red horizontal bold lines).

vious work [19], we evaluated the ensemble method on a KPI dataset containing data from 70 cells of a live mobile network and showed that it provides significant detection performance improvements over stand-alone univariate and multivariate methods. In this paper, we extend the analysis of the same dataset to evaluate our framework for practical use. For each cell, 12 KPIs were collected every hour for four months, from 11/15/2011 to 03/19/2012. The KPIs have different characteristics: some of them are measurements of user traffic utilization (such as downlink or uplink data volume or throughput), while others are measurements of call control parameters (such as drop-call rate and successful call-setup rate).

A. Ground Truth

To train and evaluate all the proposed methods, we needed access to ground truth. Ground truth is defined as the labels associated with the data points that indicate whether or not the data represents a real problem. The experimental dataset did not have ground truth associated with it. To address this limitation, we manually generated labels for the provided data based on engineering knowledge applied to KPI-data visual inspection.

Because a full manual inspection of the data would be very time consuming, we leveraged the visualization tool that we implemented for the anomaly detection process to also label the data from the 70 cells. For this purpose, we enhanced the visualization tool with the capability of inserting labels for a given KPI using visual inspection (a human-expert could use this feature to provide input to the tool).

Figure 2 presents an example of label generation for a specific KPI. A human-operator can use the slider to select the sections of the data that are considered normal or not, based on engineering knowledge. Using this tool, labels were determined for all cell data, and then used to evaluate the performance of all the proposed methods, indicating their capability of classifying the data according to these labels.

Figure 3 presents the average percentage of data points labeled as representing a problem/degradation for all cells. The percentage is less than 15% for the majority of the cells, with few exceptions. After a more detailed analysis, we discovered that the majority of the cells on average had few degradation periods (fewer than two) (see Figure 4); however, these periods

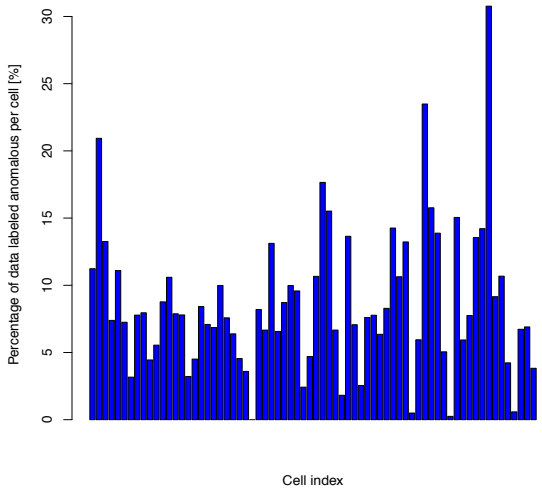

Fig. 3. Average of the percentage of data points labeled as degraded (using the visualization tool) for all cells

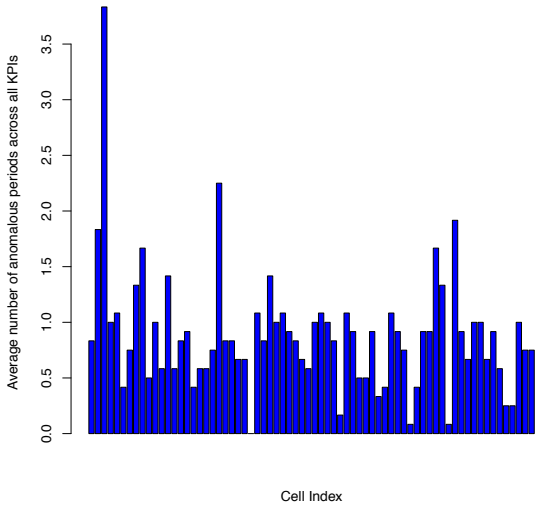

Fig. 4. Average number of degradation periods for all cells

persisted for a long time.

B. Use of Contextual Information as a Trigger

As described in Section II, we developed a triggering mechanism to train and age the set of models, and we implemented it as a user interaction in the visualization tool.

Figure 5 presents the triggering mechanism and the output of the ensemble method given a set of labels (as part of our visualization tool). The top graph illustrates the raw KPI measurement for a particular cell, which after 2/28/2012 exhibits a significant drop in value. When labeled data (illustrated in blue after the value drop) is available, our framework uses it to adapt and make better predictions. This scenario is equivalent to using labels in the form of CM changes, a confirmed FM alarm, or specific human-expert knowledge (e.g., some external conditions like a special event causing high traffic load). In this particular example, we could assume that the CM data indicated that a change had been made in the system

Fig. 5. Data labeling (top) and KPI degradation level computation (bottom). In red, the precomputed KPI degradation levels; in blue, the KPI degradation levels computed based on user input (marked as blue in the top graph.)

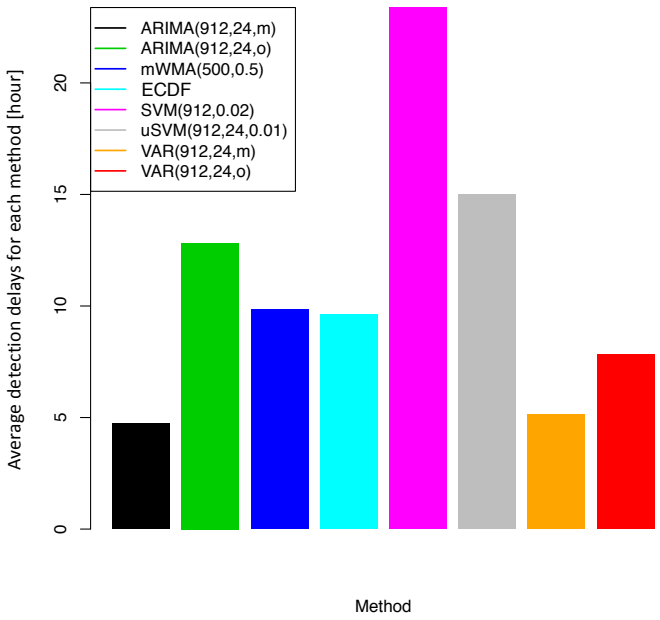

Fig. 6. Average detection delays for each proposed method (the methods in the legend are represented in the graph, in order, from left to right).

and that the change in the KPI measurement is normal. This shows the capability to generate knowledge on one hand (e.g., to detect / localize high load during a special event the operator is unaware of) or to take into account available knowledge (e.g., about a special event an operator is aware of) in its detection decisions (i.e., not to raise a detection event in that case).

The bottom graph illustrates the KPI degradation level computed with and without considering the labeled data. We observe that without the labels, the system would deem the data after 02/28/2012 as abnormal (given the high KPI degradation value in red), and that while using labels, our system adapts to the change (given the low KPI degradation level in blue). When labels are available, the ensemble method creates new models for the pool and uses the label information to adjust the weights accordingly (being a dynamic method that copes with concept drift). The ensemble-method implementation included the aging mechanism that would trigger the removal of models based on performance and age in case the maximum number of models (10 in our implementation) was reached.

C. Detection Delay

Next, we wanted to determine how well the proposed methods perform in terms of the detection delay to further assess their deployment feasibility. We define the detection delay as the time difference between the timestamp at which an anomalous period starts (given the labeled data) and the timestamp at which a method detects the first anomalous data point for that period (smaller values indicate better performance). To determine if a data point is normal or abnormal, we used $\tau = 0.5$.

Figure 6 presents the average detection delay for each proposed univariate or multivariate method. We observe that the longest delay is exhibited by the multivariate SVM method (more than 24 hours), while the shortest ones are the dynamic ARIMA and VAR methods (less than 5 hours). However, the detection delay cannot be considered in isolation; rather, it should be considered together with the false positive and detection performance. The method that classified the data closest to the labeled data while maintaining a low false positive rate was the ensemble method, and its detection delay is just under 10 hours.

The detection delay is dependent on different parameters such as the rate at which data is sampled (i.e., hourly measurements for the data considered in our experiments), the window size or the periodicity used for the different univariate and multivariate methods (i.e., 24 hours), and the threshold used for determining how to label data as normal or anomalous (i.e., $\tau = 0.5$). Given all these dependencies, the detection delay exhibited by the different methods is suitable for cell anomaly detection. However, if the sample rate at which KPI measurements are collected were increased, a corresponding faster detection would be expected.

D. Computational Performance

To complete our analysis, we further analyzed the time and space complexity of the proposed ensemble method and of the univariate and multivariate methods to determine the feasibility of deploying the methods in a real environment.

Theoretically, the SVM training method offers $O(m^3)$ time and $O(m^2)$ space complexity, where m is the training size [10], [11]. However, the practical SVM implementations use powerful approximation algorithms that can scale up to large datasets [12], [13], [14].

The ECDF method provides $O(mnk^2)$ time complexity and $O(nk)$ space complexity, where m is the number of training/testing windows, n is the number of selected profiles, and k is the window size (the KS test is done in $O(k^2)$).

TABLE I
COMPUTATIONAL PERFORMANCE EVALUATION FOR THE DIFFERENT UNIVARIATE AND MULTIVARIATE METHODS

Models	Training time (s)	Testing time (s)
ARIMA	27.67	19.65
ECDF	13.73	38.48
uSVM	21.11	337.53
SVM	6.05	39.10
VAR	5.04	25.33

The computational complexity for an ARIMA model is $O(k^3h)$ [15], where $k = max(p, d, q + 1)$, and p, d, and q are the non-negative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model, and h is the history length based on which the prediction is made.

The VAR method has the highest computational complexity, given that it is a multivariate method. However, libraries such as the ones in R [16] implement highly optimized algorithms that can cope with large datasets.

The ensemble method makes predictions based on the outcome of all univariate and multivariate methods. Thus, it inherits the methods' complexities as it makes all the weight updates and predictions in linear time.

We also performed a computational performance evaluation for all the univariate and multivariate methods (Table I). We ran our experiments on a Mac OS X Lion 10.7.4 machine, with a 1.7 GHz Core i5 processor and 4 GB of memory. The code was all written in R. Note that the training time is the average time for training the KPI models for 10 cells for 906 hours of data, and the testing time is the average time for testing 2,117 hours of data against the KPI models for 10 cells. As illustrated by Table I, the computational overhead is minimal, given that the rate at which data is sampled is 1 sample per hour.

IV. DEPLOYMENT AND OPERATIONAL DISCUSSION

Given the proposed framework, a few aspects need to be considered in order to deploy and use it in an operational environment. The first aspect relates to the way the framework can be deployed. The framework is envisioned for different types of network operators, different KPIs collected at different sample rates, and different maintenance and configuration management. It is designed such that it copes with this diversity; however, different parameters need to be calibrated during the deployment phase:

- τ is one of the most important parameters that needs to be calibrated, given that its value influences the false positive and detection rates. First, the acceptable false positive rate for the proposed environment is decided. For a given dataset and its ground truth, both training and testing phases are performed for different values of the τ. The τ value that corresponds to the acceptable false positive rate, while maximizing the detection rate, will be considered as the optimal value. This whole process can be automated, requiring only a dataset and its associated ground truth as the input.
- *window size* / *history size* / *periodicity* are the next parameters that are determined, given the environment in which the framework is expected to run. The first two parameters are related to the rate at which the KPI measurements are collected. More important, they are also proportional to the periodicity of the KPI measurements. We conjecture that the diurnal periodicity will be exhibited by most of environments, given that it also depends on diurnal human behavior. Moreover, both window size and history size need to be automatically determined by leveraging the existence of a dataset and its ground truth, and determining the best-performing window size and history size. We recall that, for example, the window size influences the detection delay (i.e., if the window size is large, the detection delay will increase), so it is important to find the optimal values for both window size and history size.

Another important aspect related to the deployment of the framework is the selection of the KPI measurements used for cell degradation detection. The process must also take into consideration the expert knowledge. Once the framework is deployed, it enters the operational mode in which, using the built profiles, the KPI levels are generated. The visualization component provides the mechanism for comparing multiple cells or KPIs and also for taking input from the operator, as ground truth information. The whole framework provides a situational awareness capability which, together with a diagnosis framework, leads to an automated mechanism for taking recovery actions.

V. RELATED WORK

Our proposed framework aims to detect partial and complete degradations in cell-service performance. In the past, research addressed the cell-outage detection [2] and cell-outage compensation [3] concepts. For the problem of cell-outage detection, Mueller et al. [2] proposed a detection mechanism that uses neighbor cell list (NCL) reports. They distinguished between three types of sleeping cells: a degraded cell that still caries some traffic, but which is not fully operational; a crippled cell which is characterized by a severely decreased capacity due to a significant failure of a base station component; and a catatonic cell that is completely inoperable. The proposed algorithm used the NCL reports to create a graph of visibility relations. Cells represented the vertices, the edges were generated based on the NCL reports, and the number of mobile terminals that reported a neighbor relation gave a wedge weight. The algorithm monitored the changes in the visibility graph, comparing two successive graphs and determining when a node became isolated. Compared to our work, Muller's approach was limited to only catatonic-cell detection, while not every isolated node reflected an outage situation. While related, cell-outage compensation [3] approaches are complementary to our work. Their goal is to automatically adjust the parameters of cells neighboring a failed cell/site

such that the coverage is maximized (e.g., by using radio-level compensation methods).

Another approach for estimating failures in cellular networks was proposed by Coluccia et al. [17] to analyze events at different levels: transmission of IP packets, transport and application layer communication establishment, user level session activation, and control-plane procedures. They proposed a method for estimating the failure probability on all these layers. All considered events were associated with an individual user in the mobile network and would have a binary outcome: success or failure. One identified challenge was that the individual rate of requests varies widely across users. D'Alconzo et al. [18] proposed an anomaly detection algorithm for 3G cellular networks that detects events that might put the stability and performance of the network at risk. The proposed algorithm was a change-detection algorithm applied on different independent features at different timescales. The main identified challenge was the need to cope with the non-stationarity and seasonality exhibited by real network traffic, while detecting anomalous events that affect multiple mobile users at the same time.

More recently, detection of general anomalies have also been addressed [4], [5], [6]. However, to the best of our knowledge, our approach is the first to employ an adaptive ensemble method that copes with concept drift, while demonstrating the feasibility of an operational deployment as shown by the performance evaluation results.

VI. Conclusions and Future Work

We tested the ensemble-method framework on a dataset consisting of KPI data collected from a real operational cell network. The experimental results expand on the significant detection performance improvements over stand-alone univariate and multivariate methods [19], and illustrate the capability of coping with the concept-drift problem (Section III-B). The results also show that the computational overhead and the detection delay are sufficiently low for practical use of our methods to perform cell anomaly detection in operational networks.

We are currently planning experimental evaluation of our cell anomaly detection method in a network operator setting. We are also integrating our detection component with a diagnosis engine that will combine the detector output with other information sources to assist operators in determining the cause of a detected anomaly. The methods presented here also serve as the foundation for our research in other areas of network operation, specifically to evaluate the impact of configuration changes on critical measures of network performance.

Acknowledgment

We thank Lauri Oksanen, Kari Aaltonen, Richard Fehlmann, Christoph Frenzel, Péter Szilágyi, Michael Freed, and Christopher Connolly for their contributions.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori (eds.), "LTE Self-Organizing Networks (SON): Network Management Automation for Operational Efficiency," Wiley, 2012.

[2] C. M. Mueller, M. Kaschub, C. Blankenhorn, and S. Wanke, "A Cell Outage Detection Algorithm Using Neighbor Cell List Reports," International Workshop on Self-Organizing Systems, 2008.

[3] M. Amirijoo, L. Jorguseski, R. Litjens, and L.C. Schmelz, "Cell Outage Compensation in LTE Networks: Algorithms and Performance Assessment," 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), 15–18 May 2011.

[4] A. Bouillard, A. Junier and B. Ronot, "Hidden Anomaly Detection in Telecommunication Networks," 8th International Conference on Network and Service Management (CNSM), Las Vegas, Nevada, 22–26 Oct. 2012, pp. 82–90.

[5] P. Szilágyi and S. Nováczki, "An Automatic Detection and Diagnosis Framework for Mobile Communication Systems," IEEE Transactions on Network and Service Management, Vol. 9, No. 2, June 2012, pp. 184–197.

[6] S. Nováczki, "An Improved Anomaly Detection and Diagnosis Framework for Mobile Network Operators," 9th International Conference on Design of Reliable Communication Networks (DRCN 2013), Budapest, Mar. 2013.

[7] S. Rüping, "SVM Kernels for Time Series Analysis," In R. Klinkenberg *et al.* (eds.), LLWA 01—Tagungsband der GI-Workshop-Woche Lernen—Lehren—Wissen—Adaptivität, Forschungsberichte des Fachbereichs Informatik der Universität Dortmund, pp. 43–50, Dortmund, Germany, 2001.

[8] B. Pfaff, "VAR, SVAR and SVEC Models: Implementation Within R Package vars," Journal of Statistical Software, Vol. 27, Issue 4, 2008.

[9] N. Littlestone and M.K. Warmuth, "The Weighted Majority Algorithm," Inf. Comput. 108, 2, 1994.

[10] D. M. Green and J. A. Swets, Signal Detection Theory and Psychophysics. New York, NY: John Wiley and Sons Inc. ISBN 0-471-32420-5, 1966.

[11] I. W. Tsang, J. T. Kwok, and P.-M. Cheung. 2005. Core Vector Machines: Fast SVM Training on Very Large Data Sets. J. Mach. Learn. Res. 6 (Dec. 2005).

[12] "Soft Margin Classification," http://nlp.stanford.edu/IR-book/html/htmledition/soft-margin-classification-1.html

[13] "kernlab", http://cran.r-project.org/web/packages/kernlab/index.html

[14] "LIBSVMA Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm

[15] K. Deng, A. Moore, and M. Nechyba, "Learning to Recognize Time Series: Combining ARMA models with Memory-based Learning," IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, 1997, pp. 246–250.

[16] "The R Project for Statistical Computing," http://www.r-project.org/

[17] A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "Bayesian Estimation of Network-Wide Mean Failure Probability in 3G Cellular Networks," In PERFORM, Vol. 6821, Springer (2010), pp. 167–178.

[18] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "A Distribution-Based Approach to Anomaly Detection and Application to 3G Mobile Traffic," Global Telecommunications Conference (GLOBECOM) 2009.

[19] G. Ciocarlie, U. Lindqvist, S. Nováczki, H. Sanneck, "Detecting Anomalies in Cellular Networks Using an Ensemble Method," 9th International Conference on Network and Service Management (CNSM), Zürich, Switzerland, 14–18 Oct. 2013, pp. 171–174.