A Scoring Method for the Verification of Configuration Changes in Self-Organizing Networks

Szabolcs Nováczki[1], Tsvetko Tsvetkov[2], Henning Sanneck[3], and Stephen S. Mwanje[3](✉)

[1] Nokia, Budapest, Hungary
szabolcs.novaczki@nokia.com
[2] Department of Computer Science, Technische Universität München, Munich, Germany
tsvetko.tsvetkov@in.tum.de
[3] Nokia, Munich, Germany
{henning.sanneck,stephen.mwanje}@nokia.com

Abstract. In today's mobile communication networks the increasing reliance on Self-Organizing Network(SON) features to perform the correct optimization tasks adds a new set of challenges. In a SON-enabled network, the impact of each function's action on the environment depends upon the actions of other functions as well. Therefore, the concept of pre-action coordination has been introduced to detect and resolve known conflicts between SON function instances. Furthermore, the idea of post-action SON verification has been proposed which is often understood as a special type of anomaly detection. It computes statistical measures on performance indicators at a relevant spatial and temporal aggregation level to assess the impact of a set of (SON-evoked) Configuration Management (CM) changes.

In this paper, we present such a verification technique, which utilizes Key Performance Indicator (KPI) normalization, aggregation and statistical processing for dynamically changing areas of the network. In addition, the introduced approach rewards or punishes CM changes based on their impact on the network and generates a recommendation to accept or undo them. A Coverage and Capacity Optimization (CCO) case study based on real Performance Management (PM) and CM data from an operator's Wideband Code Division Multiple Access (WCDMA) network is presented.

1 Introduction

Mobile SONs contain a potentially large number of concurrently operating SON function instances that need to be managed in order to achieve system-level operational goals [1]. Typically, this task is delegated to pre-action SON coordination, which defines rules used to prevent *known* conflicts [2,3]. One type of conflicts includes such that occur when instances of SON functions operate on

R. Agüero et al. (Eds.): MONAMI 2015, LNICST 158, pp. 3–15, 2015.
DOI: 10.1007/978-3-319-26925-2_1

shared CM parameters. Another type consists of those where the activity of one SON function instance affects the input measurements of another one. In addition to that, we can face situations where two function instances are in a direct conflict, e.g., both try to change the cell coverage area of two neighboring cells, or in a logical dependency, e.g., one changes the coverage area and the other one adjusts the handover parameters for that area.

In addition to using existing engineering knowledge to manage and coordinate SON function instances, it has been proposed to also verify the automated operation of the set of SON function instances [4]. This type of verification can be considered as a special type of anomaly detection that allows detecting previously *unknown* problems. The verification process itself comprises of three steps [5]: (1) defining the verification area, (2) running an anomaly detection algorithm, and (3) performing diagnosis on the impacted network elements. During the first step the network is partitioned in sets of cells, also sometimes called observation areas or scopes of verification, that are being under assessment. During the second step several KPIs are aggregated for each of those network regions and an anomaly detector is triggered to assess them. During the third step the decision is made to either accept the given changes or rollback bad performing areas to some previous network configuration.

In this paper we present verification method that follows those three steps. First of all, it divides the network in verification areas based on the CM changes that are occurring over a certain time span. Then, it starts an assessment interval during which it continuously monitors the performance impact of those changes and either rewards or punishes them based on whether they had a positive or negative influence on the network. As an outcome, our method generates a recommendation to either accept or reject the given CM change(s).

Our work is structured as follows. Section 2 describes how we assess CM changes occurring in the network and how our verification method works. In Sect. 3 we explain how we gather the performance data and how we compute the statistics that allow us to detect an anomalous network behavior. Section 4 outlines the results after using our method on real network data. Our paper concludes with the related work and a summary.

2 Assessment of CM Changes

In this section we describe our CM assessment method used to evaluate CM changes from performance effects point of view and use its output to place recommendations to accept or undo the corresponding CM change. The high level overview of the mechanism is shown in Fig. 1. In the following we are going to introduce each step.

2.1 Assessment Request

Our assessment method can either be triggered by a SON coordinator after accepting a CM change request [4], or after analyzing the changes in tables of a CM history database. In the latter case we expect that there is no direct interface

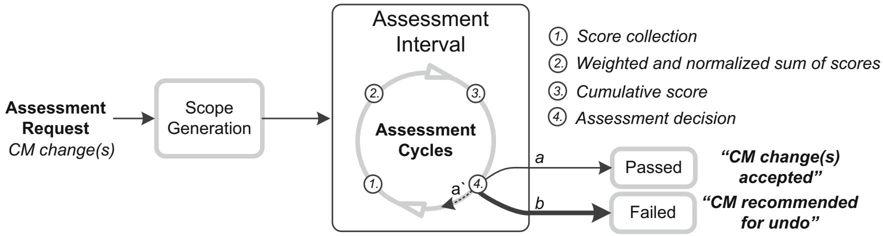


Fig. 1. Overview of the CM assessment method

between the assessment function and the initiator of the CM change. Typically, this happens when we perform manual CM changes or when we allow functions that are out of the scope of the coordinator to get active.

2.2 Scope Generation

The scope of a CM change is the set of cells that might be affected by the corresponding parameter adjustment. In research, several approaches of how to select the scope of verification (also called verification or observation area) have been introduced. A common technique is to compute a verification area by taking the impact area of the SON function instance whose activity is being under assessment [4]. Furthermore, areas of dense traffic, difficult environments and known trouble spots can be considered during the selection process as well [6]. Another possible solution is to consider the cell neighbor relations, e.g., by taking the first degree neighbors of the reconfigured cell [7]. In a mobile network two cells are neighbors when they have a common coverage area so that a handover of User Equipments (UEs) can be made.

After selecting the scope of a CM change, we observe the scopes of other CM changes and try to determine whether they can be combined. CM changes that are generating the same scope (or scopes with significant overlap) and within the same KPI granularity interval are merged. For example, if we have KPIs with hourly granularity the CM changes that have been generated within the same hour belong to the same scope. The resulting merges are handled as one entity during the rest of the process.

2.3 Assessment Interval Preparation

The first thing we do during the assessment interval preparation is the specification of the number of assessment cycles. It should be noted, however, that the interval does not have a fixed length for each set of CM changes, also called a *CM change group*. Instead, it depends on two CM parameter properties: (1) the time and (2) the type of the CM change. The first parameter allows us to make sure that only relevant system operation time is part of the assessment interval. For instance, we may have the desire to not only measure during the day but also at night. The second parameter, gives us the propagation time of the change.

Furthermore, the selection may depend on CM pattern knowledge. For example, if we know that a CM pattern is usually rejected, we may already recommend it for an undo before even starting the assessment interval. In this way, we can inform the SON Coordinator or the human operator that a harmful CM pattern is planned to be applied.

The second thing that we (optionally) do is to calculate a baseline cell level of the cells included in a scope. Note that the cell level is a high level performance indicator for the given cell, as described in Sect. 3. This particular cell level is used as a reference point to measure relative performance effects of the CM change group compared to the performance before the CM change was applied. Different methods of how this can be achieved have been described in [4,7,8].

2.4 Assessment Strategy Selection

The input of the scope generation step is the CM change group, and a set of cells composing the *scope of the CM assessment*. Since multiple parallel CM change groups can enter the assessment interval, we have decided to allow each CM change group to have its own assessment interval.

The assessment interval itself consists of assessment cycles which are driven by the selected assessment strategy. The goal of the assessment strategy is to collect and derive information about the performance impacts of the CM change so that the assessment can come to a decision. The output of the assessment cycle can either be that the CM change is *recommended to be accepted*, or to be *undone*. The CM change is accepted if it passes all the assessment cycles and rejected when it fails a single one.

2.5 Scoring of CM Changes

CM change(s) or change groups collect positive or negative scores based on the performance changes of cells in the scope. On the one hand we reward CM changes with positive scores that improve the performance of bad performing cells and/or do not impair the well performing ones. On the other hand we punish CM changes with negative scores that impair the performance of well performing cells and/or do not improve the performance of bad performing ones. CM changes accumulating significant negative scores are recommended for undo while CM changes accumulating positive scores are recommended to be accepted. In order to compute those scores, we need to perform the following steps:

- *Preparation*: calculate the baseline cell level B_{cl}, which is used as a performance reference point during the assessment.
- *Score collection*: each cell in the scope of a CM change is eligible to give assessment scores, which reflect the relative performance change of the cell since the CM change was applied.
- *Weighted average score computation.*
- *Cumulative scores*: the average score is added to the cumulative assessment score of the CM change.

– *Assessment decision*: the cumulative score is used to decide if the CM change passed or failed the actual assessment cycle.

During the preparation phase, B_{cl} is computed for all cells in the scope by taking the average cell level that has been reported during the last 24 h. The score collection phase is based on a *scoring function* used by each cell in the scope to provide feedback about the change of performance compared to the performance before the CM change was applied. The output of the function is the assessment score, which is defined as the function of the relative and signed difference of the actual cell level and the baseline cell level. In this paper we call this score the delta cell level D_{cl}.

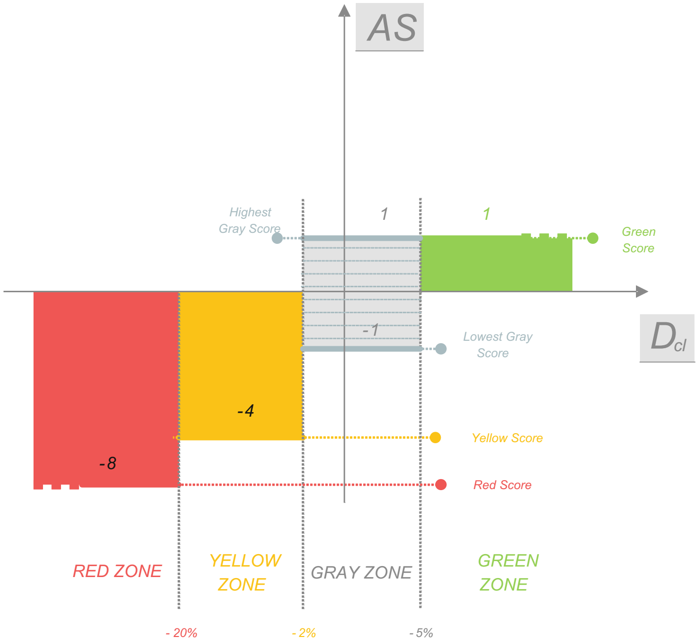
As shown in Fig. 2(a), our scoring function has four zones: green, yellow, red, and gray zones. The green zone defines the score if the cell experiences significant performance improvements while the yellow and red zones define the score if it shows moderate or significant degradation. Furthermore, the scores defined by green, yellow, and red zones are the same for all cells in the scope. This, however, does not apply for gray zone scores which may differ for each cell in the scope. The main purpose why we have designed the function to have an additional zone is to observe changes when there is no significant change in performance. Contrary to the most common assumption, which tells that no change in performance is a "good thing" and we should reward or at least not punish it, we have a rather different opinion. If a cell shows good performance before the CM change is applied and the performance remains the same after the CM change, the scoring function should take this into account and award the CM change with positive scores. However, if a cell shows poor performance and the CM change that was applied does not improve the performance, it should be labeled as an ineffective and unnecessary change, and be punished by receiving negative scores. The reasons why we need to limit the number of unnecessary changes have been outlined in [4,5].

In Fig. 2(b) we show how scores are actually given. As long as B_{cl} is in the acceptable range between 0.7 and 1 (greed domain), positive scores are returned. However, if the cell shows poor performance and B_{cl} drops below 0.7 (red domain), we punish the CM change with negative scores.

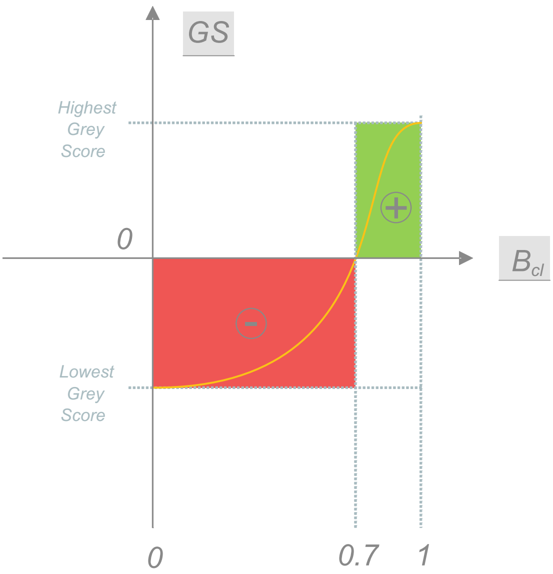
3 Assessment of Cell Performance

The conceptual diagram of the performance assessment is shown in Fig. 3. This can be considered as an aggregation pyramid with low level, cell wise KPIs on the left and the cell level in the middle and the accumulate score value on the right. Note that in this paper we use the term KPI, however, we do not want to limit the proposed method only to KPIs but to any data source that reflects some aspect of the system's performance.

The raw KPI values are converted to KPI level values, which show how far individual cell KPIs are actually from the expectations when we compare them to the corresponding profiles. More precisely, the profiles determine the accepted domain of the raw KPI values, while the KPI level measures the distance of the



(a) Zones of the scoring function



(b) Gray zone definition of the scoring function

Fig. 2. Properties of the scoring based assessment strategy (Color figure online)

actual raw KPI values from the profiles. The cell level is the aggregation of KPI levels and represent a cell's overall performance that reflects the overall behavior of the individual KPI level values. The more KPI levels are getting degraded, i.e., moving out of the acceptable domain, the more the cell level is getting degraded.

3.1 Profiling and KPI Level

The profile of a KPI is a mathematical model, which determines the acceptable domain of the KPI's values. Typical KPIs are the number of radio link failures, Call Setup Success Rate (CSSR), the Handover Success Rate (HOSR), the Channel Quality Indicator (CQI) and so on.

The profile is required to compute the *KPI level*, a value that depicts the deviation of a KPI from its expectation. To do so, the profile includes a training phase during which we collect samples $X_1 \ldots X_t$ for each KPI (t marks a training period). During this particular phase the network has to show an expected behavior. Furthermore, the duration of a training period depends on the granularity for gathering PM data from the network. For instance, it can correspond to an hour if KPIs are exported on hourly basis as presented in [7].

Then, we standardize the gathered data by computing the z-score of each data point $X_1 \ldots X_t, X_{t+1}$. Here, X_{t+1} corresponds to the current sample that we are going to observe. The level of a KPI corresponds to the z-score of X_{t+1}. It should be noted here that the KPI level can be considered as an anomaly value as well.

Let us give an example of how this may look like when we observe the CSSR for a given cell. Suppose that a cell has reported a success rate of 99.9 %, 99.9 %, 99.1 %, 99.7 %, 99.8 %, and 99.6 % during the training phase. Moreover, let us assume 90.2 % is the result from the current sampling period. The normalized result of all four samples would be 0.44, 0.44, 0.21, 0.38, 0.41, 0.35, and −2.26. The CSSR level equals to −2.26, which is the z-score of the current sampling period.

3.2 Cell Level

The cell level function creates an overall performance metric of individual cells. The output is the sum of the weighted KPI levels which we have named the *cell level*. The ability to change those weighting factors allows us to test a cell for different anomaly types. For example, we may take only handover related KPI levels into consideration when we assess the adjustments made by the Mobility Load Balancing (MLB) function.

The cell level has to fall within an acceptable range defined by two constants: c_{min} and c_{max}. Typical values for those two constants are −2.0 and 2.0, respectively. Any data-point that has a z-score higher than 2.0 or lower than −2.0 is an outlier, and likely to be an anomaly.

In addition to that, a detector is attached to this output to generate alarms if the cell levels get below a threshold. Moreover, a diagnosis module is activated by

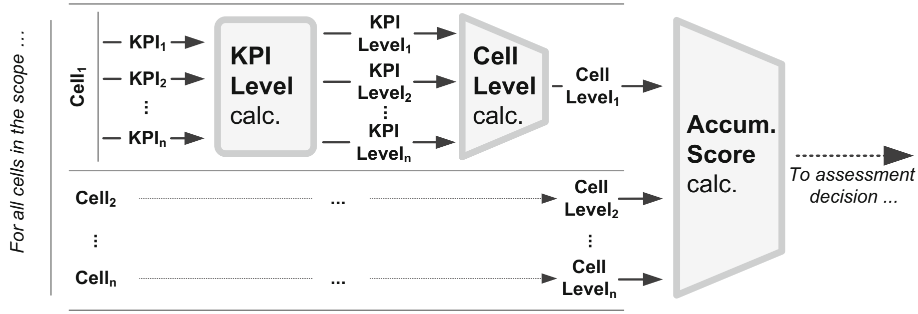


Fig. 3. Performance data aggregation overview

the detector and is used to generate detailed reports about the detected degradation. A report contains information about the worst performing cells (e.g., those having the lowest 10 % cell level values), the worst performing KPIs (i.e., the KPI levels aggregated in the observed scope, or among the worst performing cells), degradation patterns among the worst performing cells (i.e., identify groups of cells with a significantly different set of degraded KPIs).

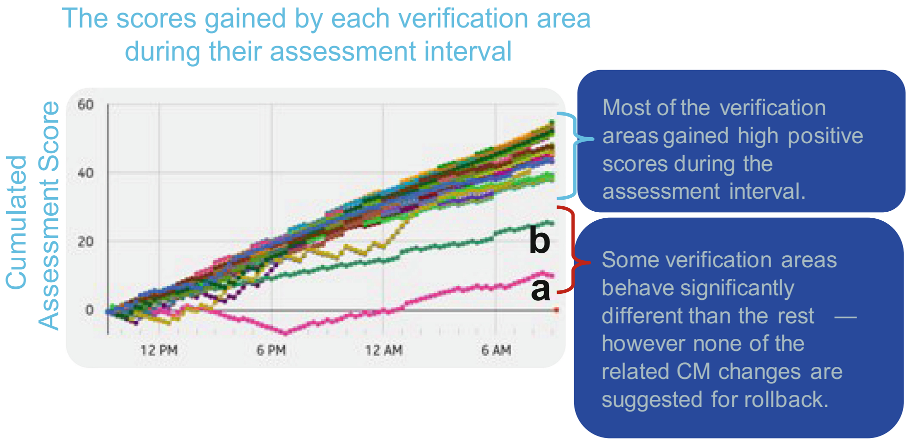
4 Evaluation

As the next step of SON verification research work the assessment algorithm has been put in action to automatically verify the performance effect of antenna tilt optimizations in a WCDMA network.

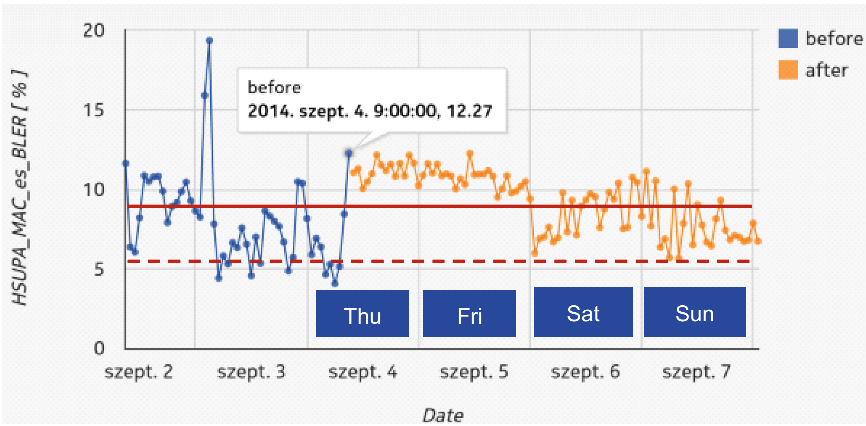
4.1 Context and Environment

The main objective was to test the performance of the CCO capabilities of the network. Based on KPI analysis results the CCO algorithm identified some cells to be optimized by either increasing or decreasing the tilt angle of the serving antenna, i.e., down-tilting or up-tilting respectively. The recommended tilt changes have been accepted and deployed in the network by the network operations personnel on 4th of September 2014 between 09:00 and 09:15. The optimization impacted 21 WCDMA cells, 11 of which were down-tilted and 10 up-tilted. In order to be able to study the effects of these tilt changes on the performance of the corresponding part of the network (referred to as optimization area), a performance measurement dataset was created. The dataset contained the hourly resolution values of 90 KPIs (covering the most important performance aspects of a WCDMA network) collected in the optimized cells as well as in their neighbors (400 cells in total) between 5th of August 2014 and 7th of September, 2014. Note that the indicated time period contains data, which shows performance of the optimization area before as well as after the tilt changes were applied. This dataset was used as the input of the verification study.

The tilt changes are deployed by using Remote Electrical Tilt (RET) modules connected to antennas. However, there are usually more than one antenna



(a) The CM assessment result



(b) The block error rate in the HSUPA MAC layer of cell W43072

Fig. 4. Impact of the CCO changes and their verification

connected to a single RET module, also called a shared RET module. In case of shared RET modules it is not possible to change the tilt angle of the connected cells independently. Consequently, in the scope of this study if the CCO algorithm suggested to change the tilt angle of a cell, which shares a single RET module with other cells, the tilt angel of all the other cells connected to the same RET module is changed. As a result we had to take 11 addition cells into account and verify the performance effects of the corresponding "forced" tilt changes. Finally, 32 tilt change events were taken into account during the study: 21 suggested by the CCO algorithm and 11 forced by the shared RET constraint.

4.2 Profiling Setup

In this study, a profile is set up to capture the usual daily fluctuation of a KPI of every cell with hourly granularity. More precisely, one profile records the mean and standard deviation of the values of the associated KPI in the corresponding cell. In addition, we distinguish between weekdays and weekends, which as a consequence leads each KPI to have two profiles. The first represents a typical weekday and other one describes the behavior of the KPI during the weekend. Furthermore, the profiles are computed on the part of the KPI dataset, which was collected before the tilt changes were applied (between 5th of August 2014 and 3rd of September 2014), thus, capturing the non-optimized performance of the optimization area.

4.3 Cell Level Computation

KPI levels are directly fed into the scoring algorithm, i.e., cell levels are not computed. This enables the system to drill down to the lowest possible level when explaining CM assessment decisions, e.g., a certain CM change is rejected by the algorithm due assessment scores crossing a certain negative threshold.

4.4 Results

In line with the post-action performance observations the automatic CM change assessment algorithm accepted all changes as no major PM degradation was observed directly after that the tilt changes were applied. However, as visible in Fig. 4(a), there were two verification areas (signified with letter a and b), gaining slightly lower scores than the others. The detailed investigation showed that while in case a the degradation started before the tilt changes were applied, thus not interesting from the verification point of view, case b turned out to be an interesting case study. Here we give some details about this, as shown in Fig. 4(b). The corresponding verification area was generated by cell `W29598`, which was up-tilted by two degrees (from 4 to 2). Looking one level deeper the analysis showed that the assessment scores of the verification area were pulled down by degradation of performance in one cell (`W43072`), which is the handover neighbor of `W29598`. We have analyzed the KPIs of this cell and observed that `W43072` shows increased values of KPI measuring the block error rate in the HSUPA MAC layer (`HSUPA_MAC_es_BLER`) after cell `W28598` was up-tilted as shown on the bottom left of Fig. 4(b). `W28598`'s up-tilting was, however, not suggested by the CCO algorithm but was forced as `W28598` is on the same RET module as cell `W13107`, which was selected by the CCO algorithm as candidate for a 2 degree up-tilt. Consequently, the CM assessment algorithm could reveal the hidden effect of an unintended tilt modification due to shared RET constraint, which was not taken into account by the CCO algorithm.

5 Related Work

While this paper is focused on applying our CM scoring method, SON verification itself is relevant from a broader use case point of view. One example is network acceptance [6] where typically fixed performance thresholds, fixed scopes (the area around a network element), and simple actions like an alarm to the network operator can be generated.

Another example is the deployment of CM undo actions. In [5], the problem of scheduling of conflicting CM undo actions has been introduced. According to the authors, a SON coordinator does not have the knowledge to resolve them and may, therefore, prevent them from being deployed. The presented approach of scheduling such undo actions makes use of minimum graph coloring in order to identify the sets of cells whose configuration can be safely rolled back. The network is partitioned in verification areas which are used as nodes during the coloring process. An edge between two nodes is added only when the two represented areas share anomalous cells that have not been reconfigured, also called a verification collision. The nodes getting the most frequently used color are marked for a CM undo.

Within the SOCRATES project [9] an idea has been introduced of how undesired network behavior can be potentially detected and resolved in a SON. The presented approach is realized by two functions: a Guard and an Alignment function. The purpose of the first one is to detect unusual network behavior like CM parameter oscillations or unexpected KPI combinations like a high Random Access Channel (RACH) rate and a low amount of carried traffic. The second one is responsible for taking countermeasures like undoing configuration changes assembled by a SON function and even suggesting SON function parameter adjustment.

In literature, several approaches have been presented of how to apply anomaly detection in mobile communication networks. In [10] topic modeling is applied to the PM data from all cells within the scope leading to the computation of topic model clusters that can be seen as indicators of the network state. Depending on the semantic interpretation, those clusters are classified as either normal or abnormal. In [11] a technique is presented that is based on an extended version of the incremental clustering algorithm Growing Neural Gas (GNG). This algorithm is known as Merge Growing Neural Gas (MGNG) and is focused on the capturing of input data behavior by taking the history data into account. Furthermore, it allows the learning of common sequences of input data and the prediction of their future values.

6 Conclusion

The existing pre-action Self-Organizing Network (SON) coordination scheme can be seen as a pessimistic approach where existing engineering knowledge is encoded into simple rules to avoid known issues in concurrently executing SON function instances. The trade-off is here that using a rule-based pro-active approach is relatively simple to implement into legacy systems, yet the knowledge implemented in the rules is rather simple. Therefore, the system may miss some relevant conditions and may enforce some coordination actions which may not be required for the specific condition. In a complementary way, SON verification is an optimistic (i.e., post-action) approach which evaluates the performance after each "round" of SON-induced actions being deployed to the network. The technical approach to realize SON verification is a type of anomaly detection and diagnosis tailored to the specific requirements. The trade-off is here that the verification is only getting active for really relevant conditions and can react also to previously *unknown* conditions which deviate significantly from the normal state. However, identifying those conditions reactively and even diagnosing their root causes is significantly more complex than just executing a set of rules.

In this paper, we addressed the problem of identifying such unknown conditions and finding at the same time the cause for them to occur. Our approach observes deployed Configuration Management (CM) changes, either computed by a SON function or manually set by the human operator. At first, we partition the network in so-called verification areas, also referred to as observation areas or scopes of verification. Then, we create an overall performance metric for each cell within a given scope, which we have named the cell level. This particular value is computed by aggregating a cell's Key Performance Indicator (KPI) levels which depict how far the KPI values actually are from their optimal domains. The computed values are used during the CM assessment interval which consists of one or more assessment cycles. During those cycles we reward or punish CM changes by giving them positive or negative scores. A key feature here, however, is the accumulation of the scores over time and punish unnecessary CM changes that did improve the network performance in case of a low cell level. The output of our approach is a recommendation to either accept or reject, i.e., undo the given configuration changes. We managed to evaluate it on real Wideband Code Division Multiple Access (WCDMA) data and outline a Coverage and Capacity Optimization (CCO) change that has led a certain part of the network to experience an anomalous behavior. This could not be foreseen by the CCO algorithm due to the shared Remote Electrical Tilt (RET) module.

References

1. Hämäläinen, S., Sanneck, H., Sartori, C. (eds.): LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency. Wiley, Chichester (2011)
2. Bandh, T.: Coordination of autonomic function execution in Self-Organizing Networks. Ph.D. thesis, Technische Universität München, April 2013
3. Romeikat, R., Sanneck, H., Bandh, T.: Efficient, dynamic coordination of request batches in C-SON systems. In: IEEE Vehicular Technology Conference (VTC Spring 2013), Dresden, Germany, June 2013
4. Tsvetkov, T., Nováczki, S., Sanneck, H., Carle, G.: A post-action verification approach for automatic configuration parameter changes in self-organizing networks. In: Agüero, R., Zinner, T., Goleva, R., Timm-Giel, A., Tran-Gia, P. (eds.) MONAMI 2014. LNICST, vol. 141, pp. 135–148. Springer, Heidelberg (2015)
5. Tsvetkov, T., Sanneck, H., Carle, G.: A graph coloring approach for scheduling undo actions in self-organizing networks. In: IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), Ottawa, Canada, May 2015

6. Ericsson: Transparent Network-Performance Verification for LTE Rollouts. White Paper, 284 23–3179 Uen, September 2012
7. Tsvetkov, T., Nováczki, S., Sanneck, H., Carle, G.: A configuration management assessment method for SON verification. In: International Workshop on Self-Organizing Networks (IWSON 2014), Barcelona, Spain, August 2014
8. Ciocarlie, G., Lindqvist, U., Nitz, K., Nováczki, S., Sanneck, H.: On the feasibility of deploying cell anomaly detection in operational cellular networks. In: IEEE/IFIP Network Operations and Management Symposium (NOMS 2014), Krakow, Poland, May 2014
9. Kürner, T., Amirijoo, M., Balan, I., van den Berg, H., Eisenblätter, A., et al.: Final Report on Self-Organisation and its Implications in Wireless Access Networks. Deliverable d5.9, Self-Optimisation and self-ConfiguRATion in wirelEss networkS (SOCRATES), January 2010
10. Ciocarlie, G.F., Connolly, C., Cheng, C.-C., Lindqvist, U., Nováczki, S., Sanneck, H., Naseer-ul-Islam, M.: Anomaly detection and diagnosis for automatic radio network verification. In: Agüero, R., Zinner, T., Goleva, R., Timm-Giel, A., Tran-Gia, P. (eds.) MONAMI 2014. LNICST, vol. 141, pp. 163–176. Springer, Heidelberg (2015)
11. Gajic, B., Nováczki, S., Mwanje, S.: An improved anomaly detection in mobile networks by using incremental time-aware clustering. In: IFIP/IEEE Workshop on Cognitive Network and Service Management (CogMan 2015), Ottawa, Canada, May 2015