Towards Proactive Context-Aware Self-Healing for 5G Networks

Muhammad Zeeshan Asghar[a,*], Paavo Nieminen[a], Seppo Hämäläinen[b], Tapani Ristaniemi[a], Muhammad Ali Imran[c], Timo Hämäläinen[a]

[a] *Department of Mathematical Information Technology*
University of Jyväskylä,
P.O. Box 35 (Agora),
FI-40014 University of Jyväskylä, Finland
[b] *Nokia Networks*
[c] *Glasgow College UESTC*
School of Engineering
University of Glasgow
James Watt Building South, Glasgow G12 8QQ

Abstract

In this paper, we suggest a new research direction and a future vision for Self-Healing (SH) in Self-Organizing Networks (SONs). The problem we wish to solve is that traditional SH solutions may not be sufficient for the future needs of cellular network management because of their reactive nature, i.e., they start recovering after detecting already occurred faults instead of preparing for possible future faults in a pre-emptive manner. The detection delays are especially problematic with regard to the zero latency requirements of 5G networks. To address this problem, existing SONs need to be upgraded from reactive to proactive response. One of the dimensions in SH research is to employ more holistic context information that includes, e.g., user location and mobility information, in addition to traditional context information mostly gathered from sources inside the network. Such extra information has already been found useful in SH. In this paper, we suggest how user context information can not only be incorporated in SH but also how future context could be predicted based on currently available information. We present a user mobility case study as an example to illustrate our idea.

Keywords: Self-Organizing Network, Self-Healing, User Context, Context Aware System, 5G Networks

1. Introduction

At the time of 1G and 2G networks deployment, mobile terminals were dumb devices and processing was done on the network side. At the advent of 2.5G, 3G, and 4G technologies, the terminals started to become more intelligent, and nowadays mobile phones are called smart phones because they have much of the processing power and intelligence that was previously believed to be done only by the network. Now mobile terminals can contribute to the network management by providing more data about the service quality, channel quality index (CQI), reference signal received power (RSRP), device location, and many other attributes. This opens new opportunities to gather data from User Equipment (UE) and to make the network better aware of the user perspective of the network coverage and services. Currently, all the data available from millions of mobile devices is not yet being fully used for network operation purposes, though. Instead, network operation and management is mostly based on only a few Key Performance Indicators (KPIs) measured from inside the network, thus using only a network perspective. Much of data available outside the network is being wasted. The requirements set by 5G technologies and the massive deployments of small cells such as micro and pico cells along with macro cells,

*Corresponding author
Email addresses: muhammad.z.asghar@jyu.fi (Muhammad Zeeshan Asghar), paavo.j.nieminen@jyu.fi (Paavo Nieminen), seppo.hamalainen@.nokia.com (Seppo Hämäläinen), tapani.ristaniemi@jyu.fi (Tapani Ristaniemi), muhammad.imran@glasgow.ac.uk (Muhammad Ali Imran), timo.t.hamalainen@jyu.fi (Timo Hämäläinen)

have made network management more challenging [1, 2]. The traditional network management tools are not enough to capture the complete behavior of the system and to propose optimal configurations.

Roughly following the definition of self-organization given in the survey [3], Self-Organizing Networks (SONs) are networks that are adaptive and autonomous and also scalable, stable, and agile enough to maintain their services in the face of all potential environmental dynamics. In future wireless networks, SON enabled operations are expected to be the default operational mode, and SON functions will have to operate in an environment with multiple operators, vendors, and radio access technologies [2, 3, 4]. The three categories of SON are Self-Configuration, Self-Optimisation, and Self-Healing (SH).

SH refers to autonomous fault management in wireless networks, including performance monitoring, detection of faults and their causes, triggering compensation and recovery actions, and evaluating the outcome. SH improves business resiliency by eliminating disruptions and ensuring network availability, reliability and retainability.

Traditional fault management based on KPI thresholds neglects user behavior and mobile phone usage patterns. Consider the situation, where many mobile users send text messages frequently and the text activity is high. Then, because of some problem in the network, the messages fail to go through. When users will experience delays or no service at all, they may silently stop using the service and may eventually shift to another operator. The operator would assume that the network is functioning well all the time. On the other hand, if user behavior was being monitored and used for anomaly detection, the problem could have been noticed and diagnosed.

The SH functions of 3G/4G are designed in such a way that they would trigger only when a problem has occurred, which makes the fault management reactive in nature. A certain time is required to observe the situation, diagnose the problem, and then trigger the compensating action. For example, a cell outage compensation function is triggered when the cell outage has been detected and user calls started to drop already. The network operator would already start losing revenue. This reactive fault management of current SONs will not be able to meet the performance requirements or the targeted quality of experience (QoE) levels of 5G network, especially the zero latency perception requirements.

Instead of detecting problems that have already occurred, an optimal SH system could also predict problems beforehand, and prevent them, thus transforming network management from reactive to proactive. Even if all problems cannot be predicted beforehand, the proactive approach could substantially reduce the intrinsic delay between the observation and compensation phases compared to current state-of-the-art SH.

Proactive fault management has been explored in the broader computer systems area, e.g., in [5]. Inspired by [5], we differentiate between root cause analysis (diagnosis) and failure prediction in communication networks as illustrated in Figure 1. The fault diagnosis mechanisms refer to the process of identifying the causes ("faults") of an already degraded network performance. On the other hand, failure prediction tries to assess the risk of a future degradation leading to a possible loss of service ("failure"). For example, in case of cell outage detection, the diagnosis mechanisms try to identify what the reason for the cell outage is, e.g., broken network element or software errors. The failure prediction refers to the assessment of whether an outage is likely to occur in the future. A possible way to achieve this goal could be the strengthened use of context built from the available user perspective and other relevant data.

In this paper, we briefly overview some very recent proposals towards SH in 5G networks. We then build upon the recent concepts by suggesting the addition of a context prediction component. For example, user behavior, such as mobility from place to place, can be modeled and used to predict future resource needs of the network to enable proactive and pre-emptive, rather than reactive, network management. Our main contributions are the following:

- A proposal of using user context and predictor models to transform SH from reactive to proactive response.
- A case study demonstrating future context prediction.

The remainder of this paper is organized as follows. In Section 2, we explain the background and the central concepts of this work, and cover some recent related works on the topics. In Section 3, we propose an approach to incorporate further context

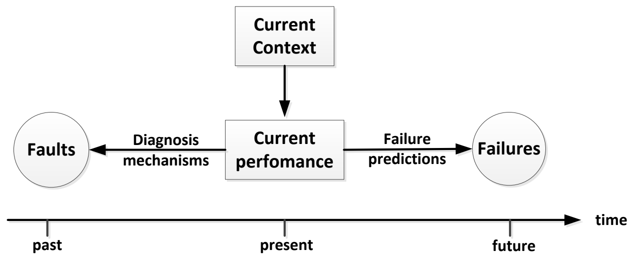

Figure 1: Difference between fault diagnosis mechanisms and failure prediction (cf. [5]).

information and especially future context prediction in SH. In Section 4, we present a practical example of context prediction. Finally, Section 5 concludes this paper.

2. Background and Related Work

2.1. Classical Approaches

Modern networks are very complex pieces of equipment [6, 7]. Besides the large variety of hardware like antennas, the backbone network, and routing components, there is also a myriad of different software stacks in these components. Furthermore, these devices are deployed in harsh environments. Hence, in practice, faults happen on a regular basis. Typical examples of network faults are software faults, broken hardware components, and inappropriate network configuration settings.

There are many performance metrics/indicators available for wireless networks that capture the network status at any given moment. These measurements are the low-level network counters and KPIs derived from them. Each KPI describes a specific aspect of the network. A KPI can be a simple average of consecutive measurements during a time period, or it can be a more advanced statistic. Typically, KPIs describe the success or non-success rates of the most important events such as handovers or dropped calls. The operator usually sets the time window for collecting network counter values before recording them as KPIs. The length of this window is a balancing act between how fast the operator can act upon a problem in the network, window size required to detect the problems, and how much data can be transferred from the base stations to the place where the SH functions are running.

When the network does not contain any (known) faults, it is possible to collect one or more KPIs and create what is called a profile of the network. This profile contains the typical values of the different indicators. The profile can be built on a per-cell basis, for each base station, or even on a wider aggregation layer (e.g., considering traffic in a cluster of base stations). Once the profile is built, continuous monitoring of the KPIs is conducted and statistically significant deviations from the profile will trigger an alarm. Often a deviation is determined by using a fixed threshold and the alarm will be triggered if the value goes beyond this bound. For example, an alarm could be raised when the call drop rate exceeds 0.1%.

Typically, the thresholds and profiles of the network are maintained in centralized Operations and Maintenance Centers (OAMs) where the KPIs and alarms are directly presented to the operator who then filters out high quality alarms manually. There can be multiple alarms generated by one fault and the same single alarm may be generated by multiple faults. It is also possible that alarms are generated without the presence of a fault. For example, any external factors, e.g., bad weather, could cause some alarms when there is no real malfunction. Sometimes it is also possible that alarm messages are not conveyed to the OAM. So alarms are not a complete/reliable source of information for fault diagnosis.

The flow of uncorrelated alarms and the big volume of alarms can be reduced by employing alarm correlation methods [8]. The alarm correlation consists of interpretation of multiple alarms, combining low level alarms to form high level alarms. The alarm correlation is an important part of SH, but alarms alone do not provide enough information to determine the root cause of the observed problems [9]. Furthermore, these methods can only reduce the quantity of alarms but not help to increase their quality. One drawback of the threshold based approaches is that they essentially quantify the KPIs into a binary space, i.e., normal and abnormal, which makes it difficult to detect performance degradations which have not yet developed into complete outages or total losses of performance.

2.2. Developments in SH Research

Earlier research on SH focused solely on automation, but in more recent efforts more focus has been given to the intelligent characterization of the network state.

Good examples of recent practical approaches to SH in real operational networks are found in [10, 11, 12, 13, 14, 15]. For example, [10] addressed the problem of verifying the effect of network configuration changes by monitoring the state of the network and determining if the changes resulted in degradations. The proposed framework consists of an anomaly detector and a diagnosis component. The anomaly detector monitors a group of cells using topic modeling. The diagnosis component, in turn, uses Markov Logic Networks (MLNs) to generate probabilistic rules that distinguish between different causes. Another anomaly detection approach using refined KPIs is presented in [12].

An incremental topic modeling approach was proposed in [15]. In that approach, the authors followed a modified version of Hierarchical Dirichlet Processes (HDP) which utilizes stochastic gradient optimization to allow the training process to evolve incrementally over time. The authors adapted that method to input all KPIs as multivariate. For the evaluation of the incremental topic modeling method, the authors used real data collected from a 3G cellular network. The incremental algorithm is run for HDP by randomly choosing timestamps from the 3G dataset and updating the model parameters accordingly. The adaptability to different cell scopes is achieved by first applying clustering to the largest scope. Then, the state of the network can be determined for subsets of the largest number of cells. The incremental approach for topic modeling will gradually update the clusters with information from the larger scope. The paper presented the initial feasibility of the incremental topic modeling approach in the context of cellular network data but the results are not mature yet.

In [16], an experimental system for comprehensive testing of different 3rd Generation Partnership Project (3GPP) Self-Optimization use cases is developed. In [17], the system is further extended to a SH framework for 3GPP Long Term Evolution (LTE) networks where detection and compensation of cell outages are evaluated in a realistic environment. The impact of SH on the KPIs such as the number of connected users and radio link failures is also shown in the paper. In [18], the authors suggested that the correlation coefficient between cell pairs can be used as a means of degradation detection in cells. In these works, the KPIs are used for detection and diagnosis of faults.

A framework for network monitoring and proactive anomaly detection is proposed in [19], using principle component analysis (PCA) for dimension reduction and kernel-based semi-supervised fuzzy clustering with an adaptive kernel parameter. The algorithms are evaluated using simulated data collected from a LTE system level simulator. The authors claim that this framework proactively detects network anomalies associated with various fault classes.

2.3. User Measurements in Traditional SH

So far, the SH research has been focused mainly on data collected from KPIs, network counters, alarms, and drive tests. In addition to these, Next Generation Mobile Networks (NGMN) and 3GPP have identified other inputs for fault management, such as direct KPI reporting in real time, UE traces, Minimization of Drive Tests (MDT) via UE reports, and location information [6].

A SH solution for 5G heterogeneous network (HetNet) architectures has been presented in [20] with separate detection methods for the control and the data plane in the split architecture of 5G (see [21, 22]) respectively. The cell outage detection is achieved using MDT with user position information. In this approach, the idea of incorporating direct reports from UEs including localization information was presented for detecting cell outages. However, the outage detection using MDT approaches is mainly done offline. Also, except user position information and received signal strength, no other information was included. The analysis was done on an elementary reference scenario using very limited examples of network failures.

The recent advances in indoor localization and UE data are utilized to provide sleeping cell detection and diagnosis solutions for 5G ultra-dense networks in [23]. An automatic root-cause analysis method using UE traces is presented in [24].

Although user measurements have been used in network management systems, the use has been limited so far, and comprehensive applications of such information have not been fully addressed.

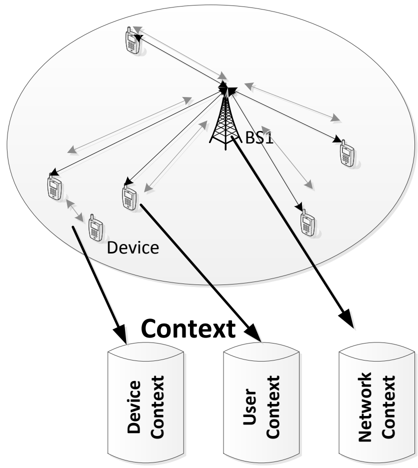

Figure 2: Three types of context: device, user, and network.

2.4. Context-Awareness

The term Context-Awareness (CA) was first introduced in the research area of pervasive computing [25]. According to the authors, CA is the ability of computing systems to acquire and reason about the context information and adapt the corresponding applications accordingly. During the last years, there has been an increasing interest in ways to share and exchange context information among remote and heterogeneous CA systems. Developments in the definition of "context" is surveyed in [26]. The early definitions, roughly to the effect of numerical state information resulting from interactions, were more primitive and limiting than the current ones, which deal more with the dynamical flow of information and knowledge within a system. One example of the use of CA in heterogeneous wireless connectivity management is [27].

There could be several types of context depending on the perspective we consider. In Figure 2, three types of context are illustrated: *Network context* consists of input from the network side, such as radio measurements, performance indicators, network configuration settings, history of configuration changes, network commissioning and planning information, etc. *User context* consists of all information about users, such as their mobility patterns, behavior, preferences, etc. *Device context* consists of the information and the influence of nearby devices which can be used in device-to-device communication (see [28]).

Traditionally, all the data for fault management was collected within the network. More recently, e.g., in [23], further kinds of data, including user context, are proposed to be considered. Context information can be broadly collected from the following three major sources:

1. UEs: location, call logs, GSM and WLAN connections, etc.

2. Cellular Network: network down for maintenance purposes, configuration changes, switching on new base stations, etc.

3. Environment: weather reports, new constructions, new buildings, railway station, events in the city or the indoor facility, etc.

A general framework for empowering SONs with big data is provided in [4]. In that paper, the authors list and categorize many possible data sources for context information applicable in SH. Here we give a few examples:

- Configuration Parameters: information on the actual configuration of network elements.
- Alarms History: messages generated by network elements when faults are detected.
- Network Counters: measurements from the network elements periodically transferred to the OAM.
- KPIs: combinations of other measurements.
- Drive Tests: field measurements related to, e.g., coverage and interference, performed in a certain area by specialized equipment such as measurement terminals and GPS.
- Mobile Traces: information from UEs.
- Call Logs: calls history information.
- Traditional context information: time, estimated UE location.

2.5. Context-Aware Self-Healing (CASH)

Recently, there has been work towards Context-Aware Self-Healing (CASH), which takes into account more of the context information. In [29], contextualized indicators for failure diagnosis are presented. The authors claim that context information can be used to support root cause analysis that provides better diagnosis results than traditional approaches. In their work, the user context was defined by location, user category and service. Recently, location-aware self-organizing methods are presented in [30].

Major challenges of small cell deployments are identified in [31]. One is "Reduced monitoring" which refers to the limited availability of troubleshooting information. Another one is "Irregular and overlapped cell areas" which makes the fault detection difficult because a fault would not create coverage holes or complete outage. Yet another one is "Performance variations" which refers to the problems occurring due to a low number of users connected to the cells. These variations generate situations where there may not be enough information about a failure for a long time. Another problem is that the fault cases usually do not deviate from the normal behavior enough to provide a significant statistical difference.

According to [31], addition of context information will help in distinguishing a fault scenario from a normal one. For example, if the user moves to a cell border, the received power will be low just as in the case of a fault. However, with context information, the cell border measurements could be separated from fault cases. For the indoor scenario, positioning information is very useful as the small cells are overlapping.

The CASH framework presented in [31] consists of 5 major blocks, i.e, *indicators' acquisition*, *context acquisition*, *context aggregation*, *inference engine* and *record update* blocks, as shown in Figure 3. For the purposes of this paper, the illustration is simplified from the original in [31]. The indicators' acquisition block collects network and user measurements and accumulates them in independent buffers. A profiling window is used to select a group of samples for statistical profiling of UEs. This block considers current measurements for generating profiles, and old samples are discarded. The context acquisition block builds the current context by combining the data obtained from different sources. The context aggregation block associates the current context with the previously recorded situations and retrieves the contextualized profile of the KPI with the same context. The inference engine block performs the detection and diagnosis of problems by comparing current KPI distributions with the contextualized profiles obtained from the context aggregation block. The record update block stores historical KPI measurements.

Incorporating user context in SH involves some challenges such as context data storing, processing, and overhead caused by transmitting extra information over the air interface. However, the feasibility of context inclusion has been already demonstrated [32].

3. Towards Proactive CASH

As observed in [5] with online failure prediction, our vision on proactive CASH can be well expressed in the words of the Greek poet C. P. Cavafy [33, p. 53]: "Ordinary people know what's happening now, the gods know future things because they alone are totally enlightened. Of what's to come the wise perceive things about to happen".

3.1. Vision of Context Prediction Applications

It is wise to predict the near-term future rather than attempting long-term prediction forecasts. In the indoor and small-cell scenarios, the near-term future is more relevant than things far ahead. The small cells are so dynamic that it does not make sense to make long-term predictions based on radio measurements and KPIs collected for small-cells. However, in these dynamic and complex indoor environments, the short-term predictions of near future are very relevant and important. In this situation, the prediction of near-future context will provide a base for forecasting the near-future network performance and the failure probabilities of the network elements.

It is known that before a cell goes into a complete outage, its performance first starts to degrade and then only after a while the cell becomes underperforming or totally dead. Finding the early signs of cell outage is very challenging because the signs may not be strong enough to be detected. In addition, it is not at all possible to detect faults that present no signs of degradation in the observed performance indicators. This is where context information comes to help by providing extra background information. In practice, the prediction of failures is not much different than early detection of the very first signs of performance degradations. By having the predicted future context, it is possible to detect those early signs of performance degradations which would lead to failures in near future.

The current CASH proposals deal with current measurements, and they are thus still reactive in nature. In what follows, we augment the SH system shown in Figure 3 to make it more proactive and pre-emptive, in order to better meet the network availability requirements of 5G.

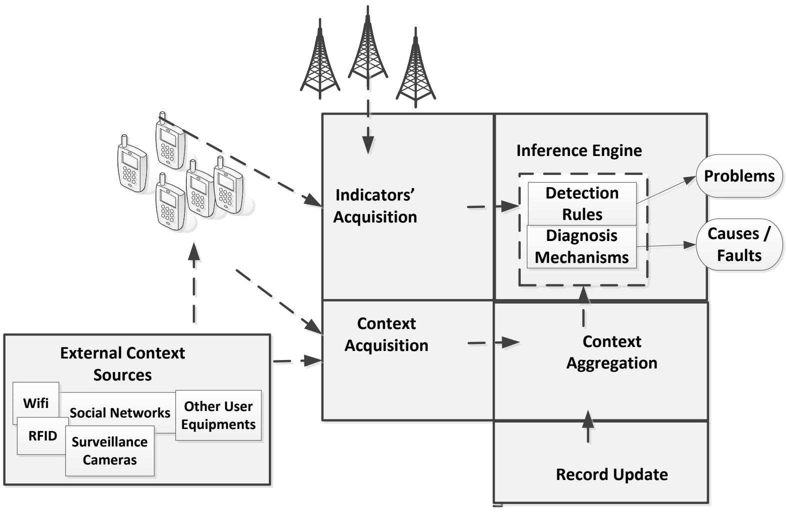

Figure 3: A simplified illustration of the CASH system presented in [31].

3.2. *Proposed Augmentation: Context Prediction Engine*

Our proposed extension to the system described in [31] is shown in Figure 4. What is added, is a Context Prediction Engine (CPE). It has been traditional to use current measurements to diagnose the root cause of a current problem. We propose that the same analysis methods could be used also to predict future failures and their causes. This scheme is illustrated in our figure that contains the same blocks as in [31] but adds the CPE component that includes a prediction model that feeds a duplicate of the inference engine with predicted future values. Also the outputs of the CPE are of the same form as in the original inference engine, but they relate to the near-future predicted situation, thus giving a *forecast of possible problems and their possible causes*. These outputs could be used to schedule preventive actions before the problems ever occur.

3.3. *Data Processing and Analysis*

Basically, any of the usual predictive methods from the machine learning vocabulary, e.g., [34] could be used. Not one method fits all purposes, so the methods should be customized and selected for each of the key attributes that are deemed worthy of inclusion in the engine. Common to these methods is that they require a comprehensive training dataset of numerical data.

The major steps required in training a predictor for one output variable usually include roughly the following:

1. Selection of the base model(s), learning algorithm(s) and the features, i.e., input measurements, to use.
2. Pre-processing and transformation of the data into a representation that is useful for the selected base algorithm(s).
3. Tuning and validating model parameters, and selecting the best-performing model for actual use.
4. Possibly combining a selected subset of the models into an ensemble that works better than any of the individual models.

4. A Case Study for Prediction of Future Context

The seeds for the idea presented in this paper were sown already some years ago when the first author of this paper participated in the Nokia Mobile Data Challenge organized by Nokia [35]. The task in the challenge was to create a user-specific predictor that learns from the user's mobility history, and predicts, based on the current user context, the next location he will visit. The next location would be considered future context, which is exactly what our proposed CPE should provide. In this paper, we use the method created for the challenge as a case study that illustrates the plausibility of the CPE.

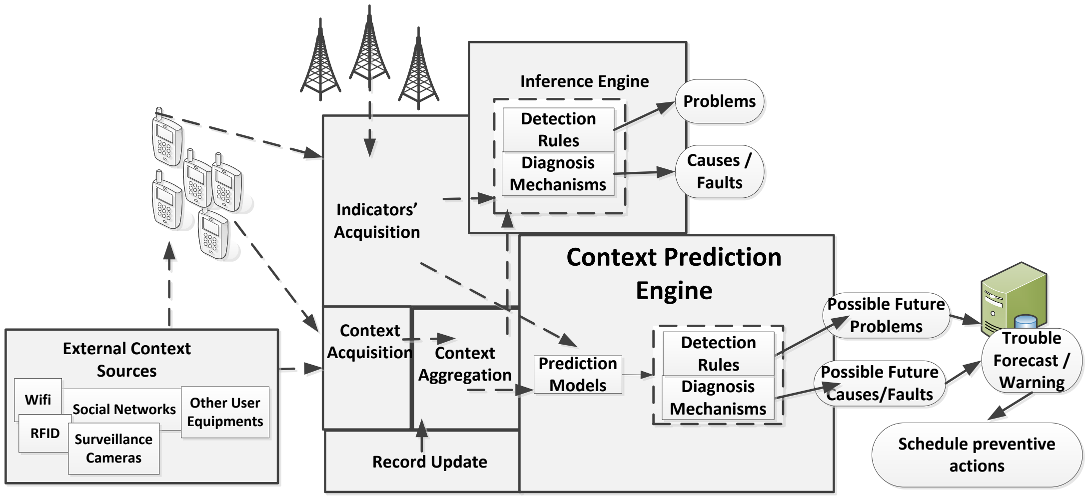

Figure 4: The CASH system of [31] augmented by a Context Prediction Engine (CPE).

The prediction of future location of a mobile user is beneficial to a network management system because of its potential application in traffic planning, radio network optimization, location-based services and also fault detection and cell outage compensation.

While, in this case study, we present the prediction of the future state of one context attribute (future user location) from the already known current context, similar methods could be used for the prediction of other attributes just as well, which could be useful in CASH.

The challenge dataset consisted of data collected from the mobile phones of 80 users, over periods of time varying from a few weeks to two years. As this was a competition, only a training dataset was given, and the final scoring was based on a testing dataset for which the true outputs were not disclosed. The sets contained smartphone data logs from disjoint time periods.

4.1. Selection of Base Models and Algorithms

The first step is to select a suitable model and useful features to use. In our scenario, the problem statement can be expressed clearly: "Given the finite set of possible locations where a mobile user can reside at a time, where will the user be next, given the current context information". This is clearly a case of classification, i.e., the prediction of the next "place ID" based on some appropriate features that can be extracted from the wealth of data obtainable from a smartphone. As usual in supervised learning tasks, the dataset was given with labels that indicate the true classes, in this case, the true destination place IDs.

From among the methods available for classification, the easiest choice during the time of the challenge was to use a specific implementation of a Multi-Layer Perceptron (MLP) that was being developed by a contemporary research group close to the competition participant. The implementation had been used earlier for continuous variable prediction in [36] and it had been found to work well also for classification tasks in other industrial projects. For further comparison and verification of the functionality of the MLP, we used also the standard, widely used, Classification and Regression Tree (CART) method available in Matlab [37].

MLPs belong to the class of feed-forward artificial neural networks [38]. They are models that comprise a number of layers of computational units, each of which performs a weighted summation and a possibly nonlinear transformation. Each unit feeds its output forward to each of the units on the next level. The structure of an MLP is illustrated in Figure 5. In this case study, only two layers were used. The knowledge acquired by an MLP is stored in the numerical values of the connection weights.

A simple MLP can be written out and computed using a compact matrix notation addressed, e.g., in [39]:

$$\mathbf{o}^0 = \mathbf{x}_i, \quad \mathbf{o}^l = \mathcal{F}^l(\mathbf{W}^l \widehat{\mathbf{o}}^{(l-1)}) \quad \text{for} \quad l = 1, \dots, L. \tag{1}$$

Here the "zero-th" output vector $\mathbf{o}^0$ is the input vector $\mathbf{x} \in \mathbb{R}^n$, i.e., the n selected numerical

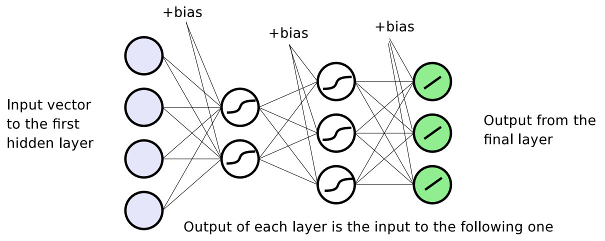

Figure 5: Schematic of a MLP neural network.

features of the current user context. For the remaining L layers, the output vector of the previous layer is always prepended with an initial element of value 1, which is denoted in the equation by a circumflex (a "hat"). The prepended vector is then multiplied by a layer-wise weight matrix, and operated element-wise by an activation function $\mathcal{F}^l$. This way the bias terms of layer l can be written as the first column of the matrix $\mathbf{W}^l$. The activation function used was the hyperbolic tangent on the inner layer and identity function on the output layer.

For compatibility with the MLP, the classes (place IDs) were encoded as binary vectors where the element with the index corresponding to the place ID is given the value 1, and the others are given the value -1. Each label c_i in a training set $\{(\mathbf{x}_i, c_i)\}_{i=1}^N$ was also thus encoded, yielding the set $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$ with target vectors $\mathbf{t}_i$. With an encoding like this, the MLP output will be decoded back to a class index by taking the index of the largest component. Numerical input features were scaled to the range $[-1, 1]$. For training the network, the conjugate gradient method was used. For further details on the formulation, we refer to [39]. The CART [37] of Matlab was used with the default parameters.

What matters in the end is that the predictions are made as accurately as possible for all the users recorded for the dataset, in average. Before comparisons of methods by their measured validation accuracies were deemed meaningful, some idea needed to be found about what is possible to achieve, e.g., by random guessing or similar crude, "baseline" methodology.

The frequencies of visits in different places differed greatly. The target class frequencies were determined by finding the total number of visits to each place during the available data collection period, which also varied between tracked individuals. Based on the class frequencies, we determined the most common place IDs, i.e., the places where the user is most likely to reside at any given time. For the baseline guess of the destination place IDs, we used the most common place always. This provided a baseline for the prediction of the next destination. By always predicting the most common place, the validation result would equal the class frequency, which turned out to be 32.5% on the average over all the different persons. *Anything above this accuracy would be an improvement* to the most naïve guess. Conversely, any method with a result worse than 32.5% would be practically useless. In this case, *even the naïve guess is better than uniform random guessing* due to the prior knowledge employed.

4.2. Selection and Generation of Features

From among the various features available, we first tried only the number of available WLAN connections and GSM cells present during particular time intervals. This was not enough information to create a classifier better than the naïve baseline guess. After experimentation, we ended up with the following features:

- Time of a visit: day of week (1-7), hour of day (0-23), and the length of the visit. It was assumed that much of human behavior can be explained by the rhythm of the society, where different things tend to happen on office days than during weekends, for example.

- The place of current visit. A person's mobility patterns could repeat themselves, as in possibly going to the supermarket directly from work every day.

- GSM and WLAN information: number of available WLAN connections and GSM cells present during the current visit. Perhaps such "connectiveness information" could give clues about the kind of location, even if it was not exactly the same as some other similar location.

- Call log information: we computed the number of calls made during a visit, number of text messages sent or received, and total duration of the calls. A person might relocate as a response to communication such as an invitation, or a certain level of communication could be indicative of some activity (e.g., work/hobbies) regardless of the current location.

- Other integrative measures of phone system information: whether the phone was charged during a visit, whether new media was noted, and whether media player had been active. These details would give further clues of what kind of activity was taking place, which might bear information about the situation preceding the next relocation.

Details including characteristics of the dataset, its partition, and the availability of different portions for the various challenge tasks are described in [35]. After the framework was built, it would have been very easy to append new, more elaborate features based on some kind of modeling of the rich smartphone data available, including, e.g., detailed data from the acceleration sensors and the actual identities of GSM cells, WLAN devices and phone numbers. Alas, there was a limited time for the competition, so a lot had to be left as future work for novel studies.

We focused on a sequence of place visits longer than 20 minutes. Besides the user identity, each entry consisted of current place ID, start and end times (normalized to hour-of-day, taking different time zones into account), and whether the visit's start, end, and transition to the next location were to be trusted (i.e., tracking data had been available between the locations). We considered only the trusted transitions in this study. Also, many place IDs occurred only once or twice in the training dataset. No classifier could have enough samples to do classification with regard to such rare occurrencies. Thus the training based on those place IDs would not be reliable for the test data, and we decided to disregard those.

4.3. Pre-processing and Transformation

We opted for a modular approach to address the problem: All users had to be modeled separately because the data was anonymous and user-specific. Also, the mobility of each user was independent of other users. A future research challenge would be to average activity patterns between models of different users of a network or a part of a larger network. After loading all available data, we pre-processed it: Anonymized user-specific strings were converted to integral numbers to make it easier to read in Matlab which was our chosen tool. Then we built a fully numeric input matrix. Each row of the input matrix represented one time period and the columns represented the features available that constituted our context during the current place being visited by the user.

4.4. Measuring of the Accuracy for Validation

We developed a prediction and validation framework to check the performance of the classifiers. The true labels can be used for evaluating validation accuracy on data rows that have not been used in training the model. A common way is to use some 70% of a given dataset for training and the remaining 30% for validation. However, in this case, for some users the data was very limited, and to make better use of all the available data, we chose to use cross-validation using 5 folds: Out of 5 randomly chosen subsets of the data, 4 subsets were used for training the model, and the remaining 1 (unseen during training) for measuring prediction accuracy. The overall accuracy, i.e., the percentage of correct predictions, was taken as the average over the 5 different divisions of the folds. Both the MLP and CART classifiers were employed in such a way. More folds were initially used, but after experimentation we found that the results were not greatly different when using only 5, which was suitable from the point-of-view of computational time.

During the training phase, a classifier looks for patterns in the training data. Here it tries to find out the patterns that connect the the mobile user's current context to the next place the user will visit. The patterns discovered may be spurious and noisy, i.e., they may be valid in training data but not valid or not strong in the test data. Validation attempts to alleviate this phenomenon.

4.5. Final Selection of the Classifier and Feature Set

Table 1 lists the cross-validation accuracies obtained with different classifiers and feature subsets. Observe that the models were generated independently for each user in the dataset, and what is shown is the average performance over all the 80 different users in the dataset. From top to bottom, the table shows the accuracies for CART, MLP and the naïve baseline guess for various selections of feature combinations:

- "all" means that all the features listed in Section 4.2 were used.

- "t&p" means that only the time and place features were selected.

Table 1: Cross-validation accuracies for method and feature selections (% correct: mean, worst, and best over all users, and average weighted by the users' occurrence in the data).

method	feat.	mean	min	max	wtd
CART	all	41.9	0	64.7	45.4
	t&p	43.1	0	62.6	46.4
MLP	all	42.1	10.0	63.1	45.9
	t&p	**45.5**	10.0	**69.4**	**49.1**
	t	43.4	10.0	65.3	46.6
	g&w	36.7	9.5	69.4	39.7
	calls	29.6	0	52.6	30.7
	other	33.4	10.0	62.6	35.8
	ens1	**46.4**	10.0	**71.2**	**49.9**
baseline		32.5	6.2	53.8	32.7

- "g&w" means that only the GSM and WLAN features were selected.
- "calls" means that only the call log feature was used.
- "other" means that the last feature set of Section 4.2 was used.

The first numerical column of the table contains the average (mean) accuracy of each method over all the users. The second column (min) gives the worst result obtained among the users. There were some very difficult cases with very few example measurements available. The third column (max) gives the best result obtained on a single user. The last column (wtd) gives an estimate that is weighted using the number of data points available for each user. For the competition, and possibly also for a real use scenario, such a measure, even if optimistic, could be more realistic, because it compensates for the difficulties posed by rare and possibly irrelevant users.

The accuracy percentages seem low at first (less than 50% when weighted with user data abundancy), but one has to understand that the data was extremely sparse and originated from a noisy real-world collection endeavour. Further feature modeling would certainly have helped. For the competition, and thus these results, there was barely enough time to create an MLP ensemble classifier that uses a weighed vote of the classifiers trained with other features. The weights were determined by trial and error.

All of the prediction results show a *clear improvement over naïve guessing*, and they could certainly have been made better with more elaborate feature extraction. From the results, we can conclude that the feature based on the number of calls and messages was bad, giving results worse than guessing. All other features, on the other hand, showed consistently better accuracy than guessing. The MLP performed better than the standard CART, and the very best results were obtained by an ensemble that combined the information from other classifiers. This, in part, supports the hypothesis that different features obtainable from the context contain different aspects of the user activity. Prediction of future situations based on the current and obtainable information is possible using machine learning methods.

5. Conclusion

In this paper we overviewed recent developments towards the inclusion of context information in Self-Healing solutions for Self-Organizing Networks. We suggested a way to make Self-Healing proactive via the prediction of near-future context, which should be especially useful in the small-cell scenarios in future 5G networks. As a technical example of plausibility, an earlier case study for predicting a user's mobility pattern was published here for the first time.

Training accurate prediction models requires more data than was available in the small case study presented here. Should obvious ethical and legal issues be resolved, long-term tracking and storage of user data would enable such models to become increasingly accurate within areas where the same users appear often.

References

[1] G. P. Fettweis, A 5G wireless communications vision, Microwave Journal 55 (12) (2012) 24–36.

[2] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, S. Geirhofer, Network densification: the dominant theme for wireless evolution into 5G, IEEE Communications Magazine 52 (2) (2014) 82–89.

[3] O. G. Aliu, A. Imran, M. A. Imran, B. Evans, A survey of self organisation in future cellular networks, IEEE Communications Surveys & Tutorials 15 (1) (2013) 336–361.

[4] A. Imran, A. Zoha, A. Abu-Dayya, Challenges in 5G: how to empower SON with big data for enabling 5G, IEEE Network 28 (6) (2014) 27–33.

[5] F. Salfner, M. Lenk, M. Malek, A survey of online failure prediction methods, ACM Computing Surveys (CSUR) 42 (3) (2010) 10.

[6] S. Hämäläinen, H. Sanneck, C. Sartori, LTE self-organising networks (SON): network management automation for operational efficiency, John Wiley & Sons, 2012.

[7] J. Ramiro, K. Hamied, Self-organizing networks (SON): self-planning, self-optimization and self-healing for GSM, UMTS and LTE, John Wiley & Sons, 2011.

[8] H. Wietgrefe, Investigation and practical assessment of alarm correlation methods for the use in GSM access networks, in: Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP, IEEE, 2002, pp. 391–403.

[9] S. Novaczki, P. Szilagyi, Radio channel degradation detection and diagnosis based on statistical analysis, in: IEEE 73rd Vehicular Technology Conference (VTC Spring), 2011, IEEE, 2011, pp. 1–2.

[10] G. Ciocarlie, U. Lindqvist, K. Nitz, S. Nováczki, H. Sanneck, On the feasibility of deploying cell anomaly detection in operational cellular networks, in: 2014 IEEE Network Operations and Management Symposium (NOMS), IEEE, 2014, pp. 1–6.

[11] G. F. Ciocarlie, C. Connolly, C.-C. Cheng, U. Lindqvist, S. Nováczki, H. Sanneck, M. Naseer-ul Islam, Anomaly detection and diagnosis for automatic radio network verification, in: International Conference on Mobile Networks and Management, Springer, 2014, pp. 163–176.

[12] S. Nováczki, An improved anomaly detection and diagnosis framework for mobile network operators, in: 9th International Conference on the Design of Reliable Communication Networks (DRCN), 2013, IEEE, 2013, pp. 234–241.

[13] G. F. Ciocarlie, U. Lindqvist, S. Nováczki, H. Sanneck, Detecting anomalies in cellular networks using an ensemble method, in: Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), IEEE, 2013, pp. 171–174.

[14] G. F. Ciocarlie, C.-C. Cheng, C. Connolly, U. Lindqvist, K. Nitz, S. Nováczki, H. Sanneck, M. Naseer-ul Islam, Demo: SONVer: SON verification for operational cellular networks, in: 2014 11th International Symposium on Wireless Communications Systems (ISWCS), IEEE, 2014, pp. 611–612.

[15] G. F. Ciocarlie, C.-C. Cheng, C. Connolly, U. Lindqvist, S. Nováczki, H. Sanneck, M. Naseer-ul Islam, Managing scope changes for cellular network-level anomaly detection, in: 2014 11th International Symposium on Wireless Communications Systems (ISWCS), IEEE, 2014, pp. 375–379.

[16] M. Z. Asghar, S. Hämäläinen, N. Meinke, Experimental system for self-optimization of LTE networks, in: Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, ACM, 2012, pp. 91–98.

[17] M. Z. Asghar, S. Hämäläinen, T. Ristaniemi, Self-healing framework for LTE networks, in: 2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, 2012, pp. 159–161.

[18] M. Z. Asghar, R. Fehlmann, T. Ristaniemi, Correlation-based cell degradation detection for operational fault detection in cellular wireless base-stations, in: International Conference on Mobile Networks and Management, Springer, 2013, pp. 83–93.

[19] Q. Liao, S. Stanczak, Network state awareness and proactive anomaly detection in self-organizing networks, in: 2015 IEEE Globecom Workshops (GC Wkshps), IEEE, 2015, pp. 1–6.

[20] O. Onireti, A. Zoha, J. Moysen, A. Imran, L. Giupponi, M. A. Imran, A. Abu-Dayya, A cell outage management framework for dense heterogeneous networks, IEEE Transactions on Vehicular Technology 65 (4) (2016) 2097–2113.

[21] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, D. Malladi, A survey on 3GPP heterogeneous networks, IEEE Wireless Communications 18 (3) (2011) 10–21.

[22] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, R. Tafazolli, Control-data separation architecture for cellular radio access networks: A survey and outlook, IEEE Communications Surveys & Tutorials 18 (1) (2015) 446–465.

[23] S. Fortes, R. Barco, A. Aguilar-Garcia, Location-based distributed sleeping cell detection and root cause analysis for 5G ultra-dense networks, EURASIP Journal on Wireless Communications and Networking 2016 (1) (2016) 1–18.

[24] A. Gomez-Andrades, R. Barco, I. Serrano, P. Delgado, P. Caro-Oliver, P. Munoz, Automatic root cause analysis based on traces for LTE self-organizing networks, IEEE Wireless Communications 23 (3) (2016) 20–28.

[25] W. N. Schilit, A system architecture for context-aware mobile computing, Ph.D. thesis, Columbia University (1995).

[26] P. Makris, D. N. Skoutas, C. Skianis, A survey on context-aware mobile and wireless networking: on networking and computing environments' integration, IEEE Communications Surveys & Tutorials 15 (1) (2013) 362–386.

[27] P. Bellavista, A. Corradi, C. Giannelli, A unifying perspective on context-aware evaluation and management of heterogeneous wireless connectivity, IEEE Communications Surveys & Tutorials 13 (3) (2011) 337–357.

[28] J. Liu, N. Kato, J. Ma, N. Kadowaki, Device-to-device communication in LTE-Advanced networks: A survey, IEEE Communications Surveys & Tutorials 17 (4) (2015) 1923–1940.

[29] S. Fortes, R. Barco, A. Aguilar-García, P. Muñoz, Contextualized indicators for online failure diagnosis in cellular networks, Computer Networks 82 (2015) 96–113.

[30] A. Aguilar-Garcia, S. Fortes, M. Molina-García, J. Calle-Sánchez, J. I. Alonso, A. Garrido, A. Fernández-Durán, R. Barco, Location-aware self-organizing methods in femtocell networks, Computer Networks 93 (2015) 125–140.

[31] S. Fortes, A. A. Garcia, J. A. Fernandez-Luque, A. Garrido, R. Barco, Context-aware self-healing: User equipment as the main source of information for small-cell indoor networks, IEEE Vehicular Technology Magazine 11 (1) (2016) 76–85.

[32] C. Baladron, J. M. Aguiar, B. Carro, L. Calavia, A. Cadenas, A. Sanchez-Esguevillas, Framework for intelligent service adaptation to user's context in next generation networks, IEEE Communications Magazine 50 (3) (2012) 18–25.

[33] C. P. Cavafy, But the wise perceive things about to happen, in: Collected Poems / C. P. Cavafy, ed. by G. Savidis. Rev. Ed. Translated by E. Keely and P. Sherrard. Princeton University Press (1992).

[34] T. M. Mitchell, Machine Learning, 1st Edition,

McGraw-Hill, Inc., New York, NY, USA, 1997.

[35] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, et al., The mobile data challenge: Big data for mobile computing research, in: Pervasive Computing, no. EPFL-CONF-192489, 2012.

[36] P. Nieminen, T. Kärkkäinen, K. Luostarinen, J. Muhonen, Neural prediction of product quality based on pilot paper machine process measurements, in: International Conference on Adaptive and Natural Computing Algorithms, Springer, 2011, pp. 240–249.

[37] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.

[38] S. Haykin, Neural Networks and Machine Learning, 3rd Edition, Pearson Education, Inc., Upper Saddle River, 2009.

[39] T. Kärkkäinen, MLP in layer-wise form with applications to weight decay, Neural Computation 14 (6) (2002) 1451–1480.