The Role of Cognitive Autonomy in “5G and Beyond” Communications Networks

Christian Mannweiler
Strategy & Technology
Nokia Solutions and Networks
GmbH & Co. KG
Munich, Germany
christian.mannweiler@nokia-bell-labs.com

Henning Sanneck
Strategy & Technology
Nokia Solutions and Networks
GmbH & Co. KG
Munich, Germany
henning.sanneck@nokia-bell-labs.com

Stephen S. Mwanje
Strategy & Technology
Nokia Solutions and Networks
GmbH & Co. KG
Munich, Germany
stephen.mwanje@nokia-bell-labs.com

Abstract—**This paper motivates the importance of cognitive autonomy in network management automation in communication networks of the fifth generation (5G) and beyond. Briefly outlining the evolution of automated network operations, the paper elaborates on the continuously increasing need for formalizing the notions of both cognition and autonomy. A taxonomy for distinguishing different levels of machine cognition and autonomy and a cognition workflow for perceiving, reasoning, and acting are presented. The application of these theoretical concepts is illustrated by two case studies, one regarding a service optimization scenario in the seaport of the City of Hamburg, the second one on anomaly detection in cellular radio access networks. Abstracting and generalizing the findings of these case studies, the most important gaps towards fully cognitive and autonomous network operations are summarized and way forward is outlined.**

Keywords—network automation, cognition, autonomy, 5G and beyond, AI and ML, case study

I. Introduction

The need for automating network management has been articulated for at least a decade [10][11]. Successive work has studied the different aspects of the Network Management Automation (NMA) challenge and different solutions for the specific challenges. This resulted in the concept of Self-Organizing Networks (SON) [12], i.e., closed-loop control mechanisms that evaluate network state and propose the appropriate (re)configurations of selected parameters. SON was largely applied to use cases with relatively simple and static workflows and parameter types. In parallel, the demand for increased levels of network automation have expanded. The introduction of new technologies in 5G has led to a combinatorial explosion of network customization possibilities and the related operational complexity has further increased. Hence, NMA needs to evolve to a more cognitive system, i.e., capable of reasoning over all possible contexts when recommending subsequent behavior. However, to systematically exploit these cognitive techniques, the cognitive process needs to be broken down into its (quasi-)orthogonal sub-processes as a means to identify (and implement) the capabilities that the available cognitive techniques can provide within this end-to-end process.

The remainder of this paper is structured as follows. Sec. II formalizes the levels of network cognition and autonomy and provides a cognitive workflow for machine-based perception, reasoning, acting, and memorizing. Sec. III and Sec. IV apply and evaluate related techniques in two concrete case studies. While Sec. V summarizes remining gaps towards achieving fully autonomous and cognitive networks, Sec. VI concludes the paper.

II. Modelling Cognition and Autonomy

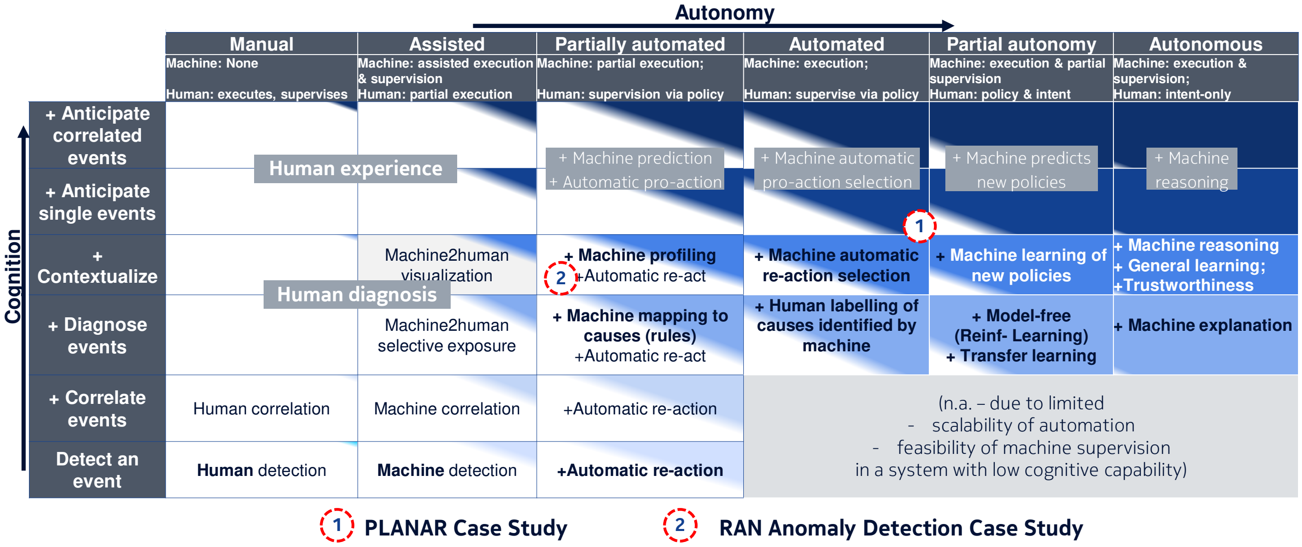
The first step in leveraging cognitive techniques comprises the formalization of the levels of cognition and autonomy of a technical system, so that input observations (and output actions) can be analyzed (and generated, respectively) at the right level of abstraction and with the right level of automation.

A. Levels of Network Cognition and Autonomy

The degree of automation in networks can be labelled with multiple terms, but they in general do not guarantee to be orthogonal. Generally, however, a network should accomplish the following capabilities listed with increasing cognitive difficulty:

1) Take a specific routine action e.g. download a file
2) Detect an event e.g., instrument a parameter with a level crossing detector on its data
3) Correlate events to match known event patterns
4) Diagnose events – distinguish among different (correlated) events via known cause-effect relationships
5) Contextualize events/data, i.e., confirm that a detection and/or correlation is only relevant under certain context(s) and untrue otherwise
6) Anticipate standalone events, i.e. find the probability of occurrence of a future event, e.g., by detection over an extrapolation of a time series, or based on a model-based (c.f. PLANAR use case in sec III)
7) Anticipate correlated events, i.e., determine the probability of occurrence of multiple future events which may require a correlation among the multiple single predictions. Relatedly, task 6 gets more complex with the length of the prediction time while, for a given time instant, task 7 gets more complex with the number of considered events.

These tasks indicate the increasing levels of cognition but do not indicate the varying levels of automation that may be



	Manual	Assisted	Partially automated	Automated	Partial autonomy	Autonomous
	Machine: None Human: executes, supervises	Machine: assisted execution & supervision Human: partial execution	Machine: partial execution; Human: supervision via policy	Machine: execution; Human: supervise via policy	Machine: execution & partial supervision Human: policy & intent	Machine: execution & supervision; Human: intent-only
+ Anticipate correlated events			+ Machine prediction + Automatic pro-action	+ Machine automatic pro-action selection	+ Machine predicts new policies	+ Machine reasoning
+ Anticipate single events						
+ Contextualize		Machine2human visualization	+ Machine profiling +Automatic re-act	+ Machine automatic re-action selection	+ Machine learning of new policies	+ Machine reasoning + General learning; +Trustworthiness
+ Diagnose events		Machine2human selective exposure	+ Machine mapping to causes (rules) +Automatic re-act	+ Human labelling of causes identified by machine	+ Model-free (Reinf- Learning) + Transfer learning	+ Machine explanation
+ Correlate events	Human correlation	Machine correlation	+Automatic re-action	(n.a. – due to limited - scalability of automation - feasibility of machine supervision in a system with low cognitive capability)		
Detect an event	**Human** detection	**Machine** detection	**+Automatic re-action**			

Fig. 1. A delineation of the levels of autonomy in networks

related with the derived decisions. As such, as illustrated by Fig. 1, characterizing the network as automated, autonomous or not would be true regardless of the combination of any such cognitive capabilities it has.

Historically networks have been managed by human operators who made all the analysis and decisions as well as the execution of the subsequent actions. Over time however the machine has been allowed to take on some of the tasks.

The most basic level of automation may be classified as machine assisted automation where the machine may be enabled to make detections and correlations which are then selectively exposed to human operators for further analysis. The operator then applies their experience to anticipate possible future events and to decide actions based thereon.

Any degree of automation implies that the machine is responsible for partial or full execution, further relieving the human operator who is then only mainly responsible for supervision of the system through the use of policies. At partial automation, the machine may automatically react with pre-set actions to event detections and correlations. With a higher level of cognition, however, the machine may add capabilities for rule-based diagnosis and profiling with which it can determine cause-effect relations based on single events or on rule-based processing of profiles of typical behavior. A highly cognitive but partially automated system may be able to predict outcomes and take pro-active decisions.

At full automation or any degree of autonomy, low levels of cognition are not applicable since full automation requires some degree of making or at least selecting actions. As such automated system do have good capabilities for detecting and correlating events. With medium cognitive capability and full automation, the machine is able to identify causes that are not a priori listed by the human operator. These causes may then only be later-on labelled by the operator. The machine may, however, also be able to select the applicable reactions for specific contexts which it then executes without requiring human approval. Moreover, given a list of potential future events, the machine may be able to predict any one or more of those likely events and select pro-active responses thereto.

Autonomy is distinguished from automation w.r.t. the supervision of the system, i.e., the human operator no longer needs to decide the policies but can control the system by only stating intents, which are the sets of desired outcomes. At partial autonomy, operator control may include combinations of intents and policies depending on the use case, but policies cease to be needed on achieving full autonomy. At partial autonomy, medium cognitive skills imply the ability to learn through model free mechanisms like Reinforcement learning and transfer learning through which the system builds its own modes of its environments and their relationships. Such models may be extended with ore data to allow the system to learn not only how to act but also to predict new policies

At full autonomy, medium cognitive skills allow the machine to provide explanations for observation or decisions. And as the level of cognition increases general learning capabilities are acquired which allow the machine to provide trustworthiness, e.g. through robustness of its decisions. Machine reasoning is the highest cognitive capability that the machine is expected to develop characterized by a "knowledge base". At this point the machine is able to create new outcomes that come from combinations of unlikely and seemingly unrelated information elements.

Our communication systems are far from reaching full cognitive autonomy, in fact the system as a whole may only asymptotically approach full cognitive autonomy. Instead, the level of cognitive autonomy should be evaluated on a use-case basis. Correspondingly, although today's 5G system may in general be considered to have achieved full automation, the reality is that for some use cases only partial automation has been achieved while for other a significant level of autonomy has already been reached. Related, looking on to future 6G networks, it should be possible to develop fully cognitive autonomous functions and applications for use case with small state spaces while other use cases will still fall somewhere within the continuum. The next subsection presents a generic workflow that such fully cognitive autonomous applications will have to implement.

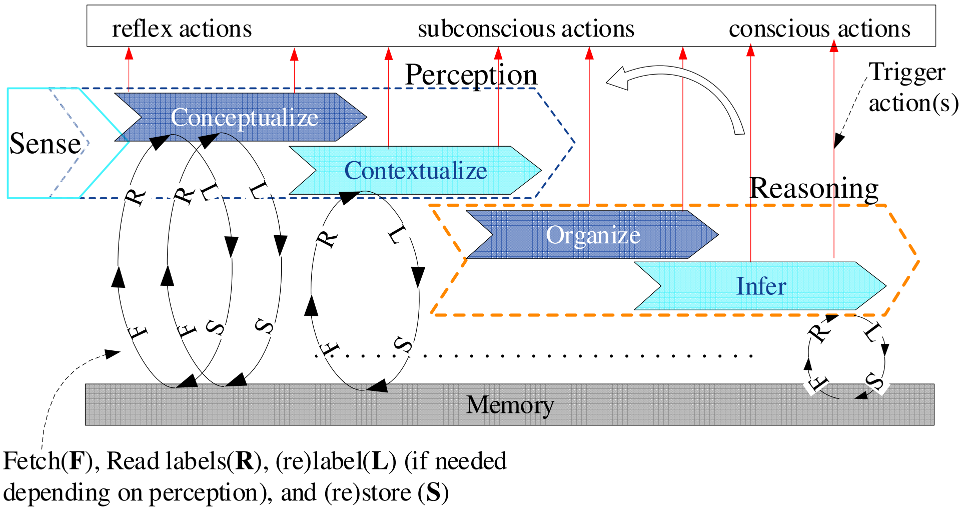


Fig. 2. Modelling cognition in network automation – a pipeline for perceiving, reasoning, acting, and memorizing [1]

B. *Overall process for perception, reasoning, acting, and memorizing*

Cognition can be described as observing a piece of data, (referred to as a data element (DE) in the following), running it through a data processing engine that then generates understanding and action [2]. For implementing this cognitive process in a technical system, a break down into the following steps is recommended, cf. Fig. 2 (loosely inspired by Bloom's taxonomy of cognitive learning objectives [3]):

1) **Perceiving**: Upon observing (sensing) the DE, it is first **conceptualized** by utilizing the detected features to match the DE with the model of known DEs. In parallel, the DE is **contextualized** by adding the situational, environmental, and further circumstances. Contextualization may therefor improve the confidence in the result of conceptualization.
2) **Reasoning**: The next step **organizes** the perceived DEs by adding attributes or relations between individual (or groups of) DEs. For **inference**, a logical analysis of the organized DEs is performed, allowing to evaluate the validity of different logical combinations (AND, OR, NOT) of two or more DEs. Similar to conceptualization, organizing and inferring rely on existing models, which may also be updated with the newly perceived DEs.
3) **Acting**: In general, actions can be triggered during any of the four stages depicted in Figure 1. However, whereas reflex and subconscious actions are rather characteristic for a human, cognitive systems for network automation typically take conscious actions in the sense that they also mandate the steps of organizing and inferring, i.e., the actual reasoning.
4) **Memorizing**: Finally, each stage of the data processing cycle can execute the memory operations cycle, which may involve up to four steps: fetch (F), read (R), (re-) label (L) and store (S). Besides enabling the comparison between observed DEs with existing models, it allows to store new or updated DE models during any step of the cognition procedure.

III. Case Study I: PLANAR – Optmizing City of Hamburg Seaport Operations

The PLANAR concept uses a Convolutional Neural Network (CNN) [4] for "Predictive Location-Aware Network Automation for Radio management" and targets smaller scale, privately-owned campus networks. In these scenarios, the creation of a digital twin of the network (e.g., for detailed simulation) is likely possible, as the network is deployed in a limited area with available detailed information about the layout of the environment (such as a digital 3D floorplan). The PLANAR concept has been studied in cooperation with the Hamburg Port Authority (HPA), implementing a fully operational 5G testbed in the seaport of Hamburg that provided different industry-relevant use cases.

The PLANAR system relies on an AI module that collects, merges, and analyzes data from different domains in order to predict mobility patterns and service degradations, up to 40 seconds into the future. This leaves enough time to the network to accommodate to expected service degradations. For the seaport study, different types of data have been collected from different sources of the testbed, which included UE-specific service-provider-level logs of GPS (Global Positioning System) coordinates, and ping, as well as radio quality measurements in the form of RSRP and RSRQ (Reference Signal Received Power and Quality, respectively) values available from the RAN. Further, cell-specific KPIs were logged, such as throughput or PRB (Physical Resource Block) utilization. The data collection took place over a time of six months with five second granularity. In total, around three million records were collected.

Using the recorded locations ("ground truth"), an MPP (Mobility Pattern Prediction) module was created using a deep CNN to predict the movement of the barges. The input to the MPP consisted of a fixed-length sequence of historical locations of one of the barges ("input sequence"), up to the most recent location. The output was a fixed-length prediction of future locations the barge will visit, cf. Fig. 3. The deep CNN was able to learn context-dependent routes around the port, such as recognizing when a barge was aligning to the shore for docking, and correctly predicting its movement even in this unusual situation. In a subsequent step, the predicted locations have been used to derive future QoS levels and determine the need for action in case of expected service degradations. Example actions include change of antenna beam configuration and transmission power settings in the cells.

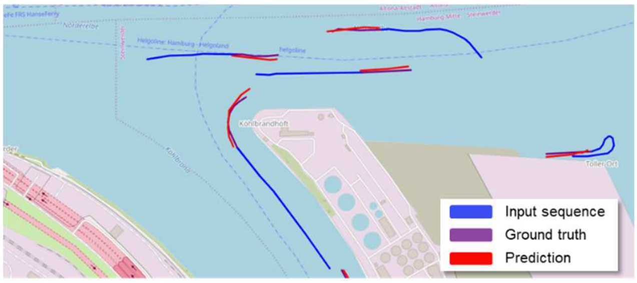


Fig. 3. Principle of CNN-based mobility pattern prediction

For large scale evaluation, a set of the collected barge location traces were fed to a network simulator, which recreated the real network conditions for these traces. The PLANAR system was then tasked to work in this simulated environment, by predicting and avoiding service degradation for the barges. Out of the 493 RLF (Radio Link Failure) events, only 12 were not detected using the farthest (i.e., most future-looking) predictions, achieving a 97.6% success rate. Note: as the barges got closer to the problematic areas, the incorrect predictions were also corrected, still before the actual RLF would have materialized (but with an effectively reduced prediction horizon, i.e., leaving less time to the network to take necessary action).

Fig. 4 shows an example of a prevented RLF. The dark blue lines depict the actual measurements, while the light blue lines depict the predicted values as forecasted 40 seconds earlier. In this example, one of the barges was on a trajectory that led behind a steep riverbank, which could have caused a complete radio link failure. The ship's route was immediately accurately predicted by the MPP module. Using this prediction, the RLF was avoided by increasing the serving beam's power and down tilting the beam to better target the ship.

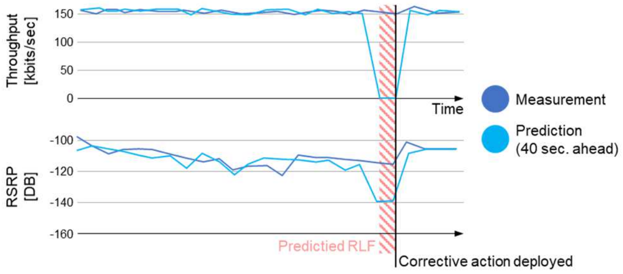


Fig. 4. Prevention of RLF by predicting location and QoS

IV. Case Study II: Anomaly Detection in Radio Access Networks

The Radio Access Network (RAN) is comprised of many network elements that are physically distributed in the serving area with relatively limited resources and less redundancy. Detecting and addressing the potential faults in these elements is not trivial. There are at least five types of faults in the RAN, including: Software faults like memory leaks, Hardware failures such as a faulty power adapter or amplifier, Misconfiguration by human operators or automation functions, Environmental impacts like shadowing effects e.g. due to new buildings; as well as Unexpected exceptional traffic, e.g., from special events.

In many networks, Anomaly Detection is achieved via expert-designed threshold monitors that process key performance indicator (KPI) metrics, generating alarms that are processed by operations personnel. However, because radio network elements operate in diverse environments, static, rule-based methods may perform sub-optimally, resulting in unnecessarily increase in workloads and missing real alarms because of the noise. Utilizing machine learning based anomaly detection methods to augment the traditional Fault Management (FM) provides practicable solutions: threshold monitors can still detect well-known problems, while anomaly detection reveals complex or previously unforeseen problems.

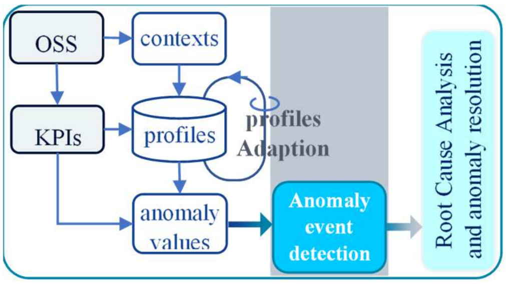


Fig. 5. Anomaly detection process (Adapted from [1])

Fig. 5 depicts an overview of a RAN anomaly detection and diagnosis function. The anomaly detection process typically includes four processes as described in the next subsections – profiling the normal behavior; adapting the profile to changes; defining the anomaly-level and detecting an anomaly.

A. *Profiling the Normal Behaviour*

A profile is the model that captures the learned normal behavior. The profiles may be statistical models of normal distributions with fitted set of parameters [6] or may consist of cluster centroids in an encoded feature space [7]. The choice of profiling-algorithm, features and context depends on application specific design choices. Among these are, e.g., the deployment location of the algorithm, the availability of labelled training data, the context and granularity of detection of the faults, the kind of available resources (computational, memory, ..) and the need for profiles to be intuitively understandable to human operators.

Fig. 6 shows examples of profiles for a RAN cell. Fig. 6a is a simple diurnal profile for modelling the behavior of a KPI time series, like traffic-dependent KPIs. Here, it models the number of Radio Resource Control (RRC) releases, with the lines showing for each hour the 1 and 2.5 standard deviations from the mean and the parameterizable boundary (the thick line) for anomaly detection. Such profiles are only meaningful for KPIs and other features that exhibit a clear time dependency. A more generic way of profiling correlations is Fig. 6b showing the normal correlation of a pair of KPIs, here the average number of RRC connected UEs and utilization ratio of 3 OFDM Packet Data Control Channel (PDCCH) symbols. A nonlinear dependency pattern can be observed, shown as the enclosed area of a hysteresis curve. The ellipse curves represent quanta in the profile to which bivariate normal distributions have been fitted respectively. The further a point is from a centroid of any cluster, the more anomalous it is. Note that since normal distributions are fitted only locally within a cluster, such profile is able to represent also more complex distributions [1].

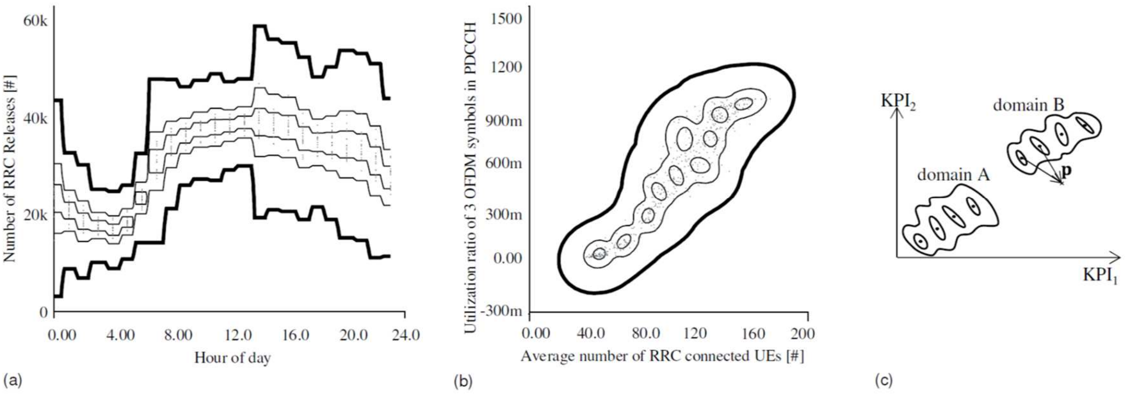


Fig. 6. (a) Diurnal, (b) cross-correlational, and (c) compound profiles [Adapted from [1]]

B. *The New Normal – Adapting to Changes*

Besides profiling the instantaneous normal behavior of RAN components, profiles should be adapted to with both seasonal and trend changes in the network traffic. RAN data continuously increases over time, so if the absolute total amount of traffic is profiled on a network level and not adapted after training, the detection function will soon indicate higher than normal traffic, yet it is the normal that has changed.

Similarly, a holiday destination in the alps exhibits a clear increase in network traffic during the skiing season when all ski resorts are full. As such, seasonal variations should also be considered normal and the profiles adapted accordingly. All these traffic variations heavily depend on the location of the RAN site and so profiles should be learned and adapted individually for each cell or site.

Adapting to seasonal and trend variations typically implies smoothing the data to make it stationary, e.g. using simple moving average or exponential smoothing methods, such as the Holt-Winters double exponential smoothing, ARIMA or SARIMA [8]. Using the simple moving average [6] e.g., the re-learning of profiles is performed in a sliding window manner, controlled by two parameters: the *span* and the *slide interval*. The *span* defines the amount of data used for each profile while the *slide interval* controls the time difference between the successive profiles. *Together*, the two parameters define the smoothness of transitions via the amount of overlap between the data used consecutive profiles.

C. *Anomaly-Level Calculation*

The anomalousness of a given observation can be calculated against the profiles. This is done for each profiled feature to get the *anomaly level* of the feature in that observation. If the feature is a single KPI that is normally distributed and profiled with a diurnal pattern as shown in the example in Fig. 6a, the anomaly level can be simply the z-score of the KPI value in that hour of day. However, many RAN KPIs have more complicated distributions, e.g. with several peaks in the probability density function. This complexity may be handles by clustering the multi-dimensional space into smaller parts, within which a multi-variate normal distribution can better approximate the data as illustrated by the two-dimensional example in Fig. 6c for the cross-correlation profile of two KPIs. The mean values of the observations in the training data are depicted as the bullets and the covariance matrices as the ellipses having axis lengths equal to the standard deviations of the marginals, i.e., the diagonals of the covariance matrix [5].

D. *Anomaly Event Detection*

Given the anomaly levels, an anomaly classification or anomaly event detection process determines the observations to be classified as anomalous and if certain anomalous observations belong together, i.e. may have the same root cause. The simplest approach thereof is to set a fixed threshold for the classifying based on the feature's anomaly level. However, this is typically inadequate. For example, is it classified as anomalous if one feature in an observation with multiple KPIs and features is above the threshold? Instead, models capable of capturing diverse anomaly classes are required. One example, illustrated by Fig. 7, is an algorithm based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN), that charts the anomaly level of the KPI in time. DBSCAN clusters the data points and calculates the anomaly value for a point in time *t* by integrating the area created by the anomaly level curve within t-ε and t+ε, a window of length ε. If this area is larger than a specified threshold *MinPts*, the observation at time *t* is labelled as anomalous. By adjusting the attributes ε and *MinPts*, the compromise between detection sensitivity for longer lasting but less severe anomalies can by adjusted. Additionally, a simple threshold can be used to detect instantaneous anomalous KPI anomaly levels [9].

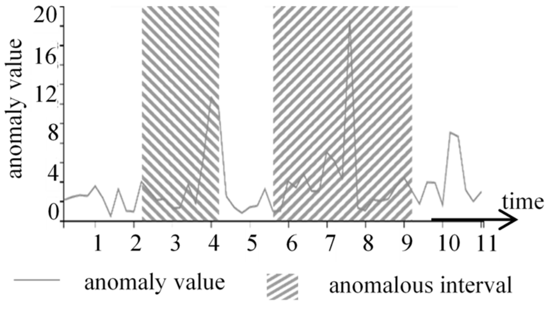


Fig. 7. Anomaly value development in time [1]

V. Remaining Gaps and Way Forward for CAN

As stated above, today's communication systems are far from reaching full cognitive autonomy. However, the case studies presented in previous sections have demonstrated, in an exemplary manner, to what extent state-of-the-art NMA systems can already realize cognitive and autonomous behavior for selected use cases. The path towards fully cognitive and autonomous networks will comprise an evolution to the point where the human operator ceases to be involved in the very large majority of the operational activities, and only be consulted in very rare and exceptionally ambiguous problems in network operations.

In general, this development will follow the different levels of automation introduced in Sec. II.A and comprise the following two major phases:

1) NMA systems of this phase incorporate advanced cognitive capabilities (as described in Sec. II.B). Examples include recent work that proposes learning functions that assume (from the human operator) the responsibility of configuring the NMA functions. In such a cognitive-automated network, the NMA system matches the prevailing contexts in the network to manually defined context model, for which the individual NMA functions can learn the best configuration parameter values. However, this remains inadequate since the operator cannot enumerate all the possible values for each context dimension.
2) Fully cognitive and autonomous networks exhibit comprehensive control over the network. This final phase requires NMA systems to implement "Cognition by Design", incorporating the learning capabilities of AI/ML techniques into each part of the functionality. The CAN is characterized by being able to learn appropriate actions for specific contexts in the network, to perceive and reason over observed contexts as well as to automatically adjust the context models during operations. I.e., the CAN also learns new context dimensions and their granularities and corresponding best actions.

VI. Conclusions

This paper has motivated the need for increased levels of cognition and autonomy in network automation and presented selected fundamental concepts enabling systems to conceptualize observed input, analyze and reason over basic and more complex data elements and their multi-dimensional relations to each other, and generate recommendations for action. Two case studies have illustrated and evaluated the capability of network management automation systems to demonstrate cognitive and autonomous behavior in specific network settings. Nonetheless, considerable work remains to be done to exploit state-of-the-art AI/ML techniques to realize the vision of generally Cognitive Autonomous Networks.

Acknowledgments

The authors would like to thank Janne Ali-Tolppa and Marton Kajo for their contributions related to the City of Hamburg seaport case study. The respective contents in this paper are based on their work on applying Convolutional Neural Networks to mobility pattern prediction and learning according network (re-) configuration actions.

Moreover, the authors would like to thank all co-authors of our book "Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond" [1], a work that has significantly guided our subsequent research in the area.

References

[1] S. Mwanje, C. Mannweiler (eds.), "Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond", Wiley, 2020.

[2] S. Mwanje, C. Mannweiler, "Towards Cognitive Autonomous Networks in 5G", Proceedings of ITU Kaleidoscope: Machine learning for a 5G future (K-2018), 2018

[3] Nancy E. Adams, "Bloom's taxonomy of cognitive learning objectives", in Journal of the Medical Library Association: JMLA103.3: 152, 2015.

[4] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, K.Q. Weinberger (eds.), Curran Associates, Inc., pp. 1097-1105, 2012.

[5] J. Ali-Tolppa, B. Schultz, S. Kocsis, L. Bodrog and M. Kajo, "Self-Healing and Resilience in Future 5G Cognitive Autonomous Networks," in ITU Kaleidoscope, 2018.

[6] L. Bodrog, M. Kajo, S. K. a. B. Schultz, "A robust algorithm for anomaly detection in mobile networks," in IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, 2016.

[7] C. Aytekin, X. Ni, F. Cricri, E. Aksu, "Clustering and Unsupervised Anomaly Detection with L2 Normalized Deep Auto-Encoder Representations," in IJCNN, 2018.

[8] G. Ciocarlie, S. Novaczki, H. Sanneck, "Detecting Anomalies in Cellular Networks Using an Ensemble Method," in CNSM, 2013.

[9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Kdd, vol. 96, no. 34, pp. 226-231, 1996.

[10] Next Generation Mobile Networks Alliance, "Use cases related to self organising network, overall description," May 2007. [Online]. Available: http://www.ngmn.org

[11] L. Schmelz, J. V. D. Berg, R. Litjens, A. M. Amirijoo, O. Linnell, C. Blondia, T. Kuerner, N. Scully, and J. Oszmianski, "Self-configuration, -optimisation and -healing in wireless networks," in *WWRF*, December 2008, pp. 4477-4479. [Online]. Available: http://link.aip.org/link/?RSI/72/4477/1

[12] S. Hamalainen, H. Sanneck and C. Sartori, Eds., "LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency", Wiley, 2012.