Diagnosis Cloud: Sharing Knowledge Across Cellular Networks

Gabriela F. Ciocarlie*, Cherita Corbett*, Eric Yeh*, Christopher Connolly*,
Henning Sanneck†, Muhammad Naseer-Ul-Islam†, Borislava Gajic†, Szabolcs Nováczki† and Kimmo Hatonen†
*SRI International
Email: {gabriela.ciocarlie,cherita.corbett,eric.yeh,christopher.connolly}@sri.com
†Nokia Networks Research
Email: {henning.sanneck,muhammad.naseer-ul-islam,borislava.gajic,szabolcs.novaczki,kimmo.hatonen}@nokia.com

***Abstract*—Diagnosis functionality as a key component for automated Network Management (NM) systems allows rapid, machine-level interpretation of acquired data. In existing work, network diagnosis has focused on building "point solutions" using configuration and performance management, alarm, and topology information from one network. While the use of automated anomaly detection and diagnosis techniques within a single network improves operational efficiency, the knowledge learned by running these techniques across different networks that are managed by the same operator can be further maximized when that knowledge is shared. This paper presents a novel diagnosis cloud framework that enables the extraction and transfer of knowledge from one network to another. It also presents use cases and requirements. We present the implementation details of the diagnosis cloud framework for two specific types of models: topic models and Markov Logic Networks (MLNs). For each, we describe methods for assessing the quality of the local model, ranking models, adapting models to a new network, and performing detection and diagnosis. We performed experiments for the diagnosis cloud framework using real cellular network datasets. Our experiments demonstrate the feasibility of sharing topic models and MLNs.**

I. Introduction

To maintain good customer experience in mobile network environments, operators need to configure a multitude of parameters to optimize various network elements. These efforts result in an increased network management complexity that requires automated procedures for operations. Automation can not only improve network-monitoring capabilities by applying machine-level network performance data analysis, but also provide "close-the-loop" procedures, in which network management systems autonomously handle certain types of events.

While automated network management and operation techniques within a single network can improve operational efficiency, sharing the knowledge learned by these techniques across different networks could further maximize their usefulness. Learning from data is an expensive process due to the complexity and growing network size. The derivation of models and operational information from several hundred different variables and events that are monitored and recorded for analysis requires a range of actions and processes.

So far, network diagnosis has focused on building "point solutions" using Configuration Management (CM), Performance Management (PM), alarm and topology information from one network. The capability to collect and share expert knowledge among different networks does not exist today. Therefore, the scope of this paper is the design of a diagnosis cloud framework that enables the extraction and transfer of knowledge across networks managed by the same operator.

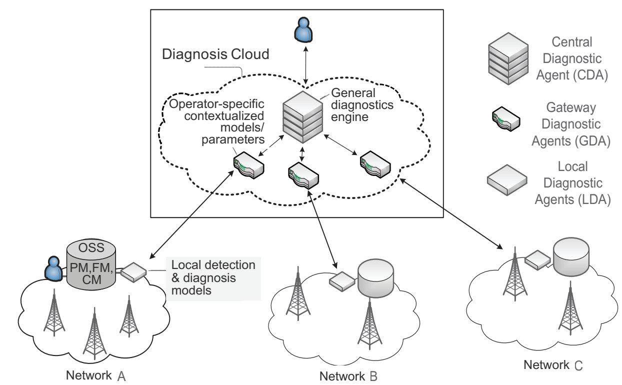


Fig. 1: A conceptual representation of the diagnosis cloud. The framework provides the ability to transfer diagnosis information from a local network to the cloud and vice-versa.

Contributions. This paper addresses the need for sharing networks' operational models across different networks. This is particularly important when new networks are deployed and they can already take advantage of existing knowledge from other networks, or when they encounter new behavior that other networks have already encountered and addressed. Main contributions include a framework and methods to:

- share the local operational knowledge with the knowledge cloud in a network-agnostic representation;
- derive new knowledge from the input of all networks;
- re-apply cloud knowledge to individual networks as needed.

II. Diagnosis Cloud Overview

An operator has to wait to accumulate experience each time a new network is deployed or a new issue is encountered. Problem resolution time is significant, and the user experience is potentially degraded until the network is sufficiently mature to provide information which an operator can use to automatically resolve problems and avoid upcoming issues. However, an NM system can benefit from diagnosis information from the

978-3-901882-85-2

operator's other networks. Networks witness similar internal (e.g., new network equipment, new software) or external (e.g., changing deployment environment, operator policy) factors that can lead to similar diagnosis processes and preventive mechanisms. While these similarities can be leveraged, cellular networks also have known technical differences that must be considered to make knowledge sharing a feasible task. For instance, an operator may deploy different radio access technologies (e.g., GSM, LTE) that have different key performance indicators (KPIs), CM, PM, and alarm information that may not be comparable to each other. Different vendor equipment (for the same technology) may use different syntax for network parameters with the same semantics. Cellular networks can also have different deployment configurations (rural vs. urban) to accommodate various needs such as QoS, coverage, voice and/or data usage patterns, etc.

To address these differences, a diagnosis cloud must be able to extract knowledge learned by the local diagnosis process of individual networks, generalize this knowledge to a network-agnostic representation, fuse and derive new diagnosis knowledge from all networks, and re-apply and parameterize knowledge relevant to the target network. Hence, the diagnosis cloud must address the need for sharing detection models and diagnosis information across different networks for the fully automated, joint operation of a set of networks. Figure 1 presents the overall diagnosis cloud concept.

The Local Diagnostic Agent (LDA) initiates a diagnosis request to the cloud when it encounters a new unknown problem. As newly deployed networks mature, they can contribute diagnostic models as the local diagnosis processes reach acceptable accuracy. LDA function blocks include:

- *Local diagnosis models*: capture anomalous behavior for multiple events models are generated by the local network using a variety of approaches, e.g., machine learning, rule-based methods, or human-driven.
- *Model Assessment*: assess the quality of a (local or cloud) model based on a local criteria.
- *Local Database*: contains locally used models that were retrieved from the cloud or derived locally.

The Gateway Diagnostic Agent (GDA) provides the data translation as knowledge is transferred to/from the cloud. The main function block of the GDA is:

- *Data Translation/Data Generalization*: maps local parameters to a network-agnostic form. When transferring knowledge from the cloud to a local network, this process maps the network-agnostic general representation into network-specific parameters. To the extent possible, these mapping operations should be lossless and privacy-preserving (so that operators can share data without revealing sensitive information). To accommodate multiple genres of networks, multiple general representations are needed to account for the distinctions between networks, such as different technologies (e.g., 3G, LTE), vendors, or usage characteristics (e.g., urban vs. rural).

The Central Diagnostic Agent (CDA) manages the global knowledge base to which all networks contribute reports and models. The CDA receives requests for model retrieval, ranks models in the cloud, and forwards relevant cloud models to the target LDA via the GDA. CDA function blocks include:

- *Model Ranking*: ranks models in the cloud based on a global (cloud) criteria or a local criteria (specified by a local network).
- *Model Similarity Check*: checks the similarity of models based on a global (cloud) criteria or a local criteria (specified by a local network).
- *Model DB update*: updates the model database when new models are available.
- *Model Database*: contains the cloud models.
- *Global knowledge*: contains the global knowledge.
- *Global Knowledge Extraction/Analytics*: operate on the model database and analyze its performance, such as extracting statistics on model usage, determining the accuracy of models and the best model, determining the best feature set for models, or combining models into one unifying model, when possible.

III. Experimental Evaluation

To evaluate our framework, we consider the case where new cells are added to an operator's existing network deployment. However, existing local models lack the proper diagnosis information to identify the network state of the new cells and determine if the anomalous behavior was caused by weather events. Our exemplary framework implementation uses a topic modeling approach for capturing the network state and Markov Logic Networks (MLNs) to provide the most likely explanation for the observed network condition.

Topic modeling [1] is a statistical modeling approach that uses the maximum likelihood of parameters occurring in a specific pattern to discover groupings (topics). We apply topic modeling to KPIs measured at the cell level to identify normal and abnormal states (topics) of a cellular network. Each topic is characterized by its distribution of KPI values. A topic is labeled abnormal if its distributions exceed operator-defined thresholds.

MLNs [13] are graphical models that allow first order probabilistic inference over a domain. MLNs allow us to specify rules (without training data) to express relationships between observations, such as KPIs, and performance of the network at different levels of abstraction. We approach diagnosis by expressing multiple hypotheses within the MLN rule set, running the inference engine, and querying it for the most likely explanations for the observed conditions.

A. Network State using Topic Modeling

We applied the topic modeling approach to two real 3G datasets, called Net1 and Net2. For both, we used 5 KPIs related to cell availability and session setup/success rates. To emulate the use case scenario described above, we divided our datasets to represent: 1) an existing network of cells (*Net1-SliceB**); 2) an addition of new cells to the existing network (*Net1-SliceB*); and 3) a set of networks (*Net1-SliceA*, *Net1-SliceC*, *Net2-Slice1*, and *Net2-Slice2*) that had submitted their models to the cloud. Finally, we use all 1032 cells (existing plus new cells) in *Net1-SliceB All* as "ground truth" to generate

Role	Network segment	No. of timestamps	No. of cells	No. of normal topics	No. of abnormal topics
New Cells	Net1-Slice B	1000	853	-	-
Initial Local Model	Net1-Slice B*	1000	179	2	1
Cloud Model	Net1-Slice A	785	894	6	12
Cloud Model	Net1-Slice C	1000	986	12	20
Cloud Model	Net2-Slice 1	1000	2169	2	8
Cloud Model	Net2-Slice 2	1000	2167	4	7
Baseline Model	Net1-Slice B All	1000	1032	10	25

TABLE I: Summary of network slices and their role in the diagnosis cloud analysis.

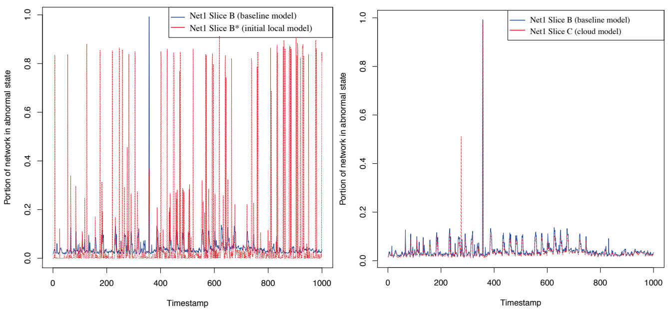


Fig. 2: **Left**: Detection outcome of *Net1-SliceB* using the existing local model compared with the baseline model. **Right:** Detection outcome of *Net1-SliceB* using the cloud model compared with the baseline model.

a Baseline model for which all analysis is compared to assess performance. Table I summarizes the different network slices and their role in the diagnosis cloud analysis. While Net1 and Net2 are both 3G networks, they were deployed in different countries and thus are contextually different (e.g., configuration settings, user requirements, etc). The use of different slices of Net1 is analogous to using different segments of a larger network with similar context and use of Net2 slices is analogous to using models from a completely different network with different context. The diversity of the models can be gleaned from the variability in the number of normal and abnormal topics across the different slices (Table I).

Performance Before Applying Diagnosis Cloud. We evaluated the abnormal state of new cells using the initial local model (*Net1-SliceB**). We also evaluated the new cells using the baseline (*Net1-SliceB All*) model for comparison to help illustrate the improvements of applying the cloud approach. Figure 2 Left shows the portion of the new segment in abnormal state using the initial local model versus the baseline model. Clearly, we observe that the initial local model assesses a network state that is much different and erroneously more anomalous than that measured by the baseline model.

Performance using Diagnosis Cloud. When new cells are added, the LDA must determine the quality of its local topic model. For a given test sample of KPI measurements observed from the new cells, the local model computes the log-likelihood values using its respective topics. We measure the mean of these values over a sliding window and extract the distance of each value from the mean of its window. When the distance exceeds an operator-defined threshold, the LDA initiates a model retrieval request to the CDA. The CDA sends all available models in the cloud to the LDA via the GDA. For a larger library of cloud models it may be more advantageous to rank the models and only forward the most relevant models to the LDA. In our evaluation, network-specific KPIs represented in the topic models were the same across different candidate networks. As a result, the data translation procedure performed by the GDA was a one-to-one mapping. When network-specific parameters are not the same (different vendor or Radio Access Technology), a more elaborated scheme is needed for mapping parameters.

The LDA locally assesses the detection capability of the cloud models it received from the CDA. Given a sliding window of KPI data samples from the local network segment, each topic model computes the log-likelihood of the sample using its respective topics. Next, we compute the moving average of the log-likelihood values for each respective model. Finally, we measure the distance $d_{m_i}(x) = ||LLL_{m_i}(x) - mean(LLL_{m_i})||$, where x is the current data sample from the new network segment, LLL is the log-likelihood of current sample, the mean is measured over previous window of samples, and m_i is the i-th model from the cloud. For each sliding window, we select the diagnosis of model m_i with the minimum distance d_{m_i}. Models are ranked based on how often they are selected. We illustrate the overall results for all models in the cloud in Table II, as the percentage of times a model is selected in the ranking process. The higher the value, the better it is: the Net1-SliceC model had the highest rank.

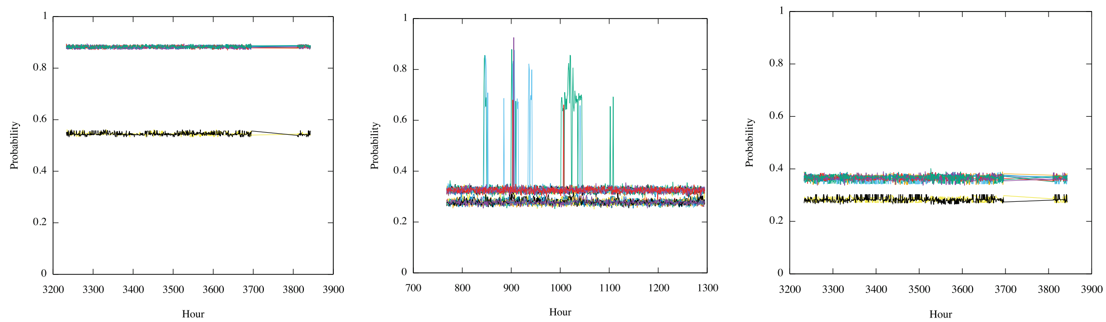
Figure 2 Right presents the detection outcome for the best-ranked cloud model *Net1-SliceC* compared with the baseline model, when tested on the network segment with the newly added cells. We note that the cloud model followed much closer to the baseline detection outcome than the local model (Figure 2 Left). These results illustrate the benefit of the diagnosis cloud and its ability to select the best available cloud model.

B. Network Diagnosis using MLNs

Building on prior work for KPI-based analysis of network performance [2], we developed MLN models, which determine whether anomalies detected by topic modeling are caused by weather-related events. Our MLN generator operates by accessing KPI measurements, cell metadata, topic model posterior probabilities, along with weather event information. The MLN reasons over groups of cells, rather than individual cells. We rely on topic modeling to perform this grouping in an automated way. Reasoning is performed at the group level, but weather conditions within the group are weighted by the

Model	Window size 10	Window size 50	Window size 100
Net1-Slice C	38.89%	44.00%	45.56%
Net1-Slice A	37.88%	31.79%	30.89%
Net2-Slice 1	12.12%	11.89%	11.11%
Net2-Slice 2	11.11%	12.32%	12.44%

TABLE II: The percentage of data samples a model is selected in the ranking process. The higher the value, the better it is (Net1 Slice C is the favorite model).



(a) (b) (c)

Fig. 3: (a) Net2 local MLN probabilities; (b) Net3 local MLN probabilities; (c) Net2 cloud MLN probabilities.

add [G] (precip(G, FREEZING-RAIN) and anomaly(G)) implies weather_event(G) 5.0;
add [G] (precip(G, LIGHT-SNOW) and anomaly(G)) implies weather_event(G) 2.0;
add [G] (precip(G, SNOW) and anomaly(G)) implies weather_event(G) 12.0;
add [G] (precip(G, HEAVY-SNOW) and anomaly(G)) implies weather_event(G) 15.0;
add [g] ~weather_event(g) 0.1;
add [G] (precip(G, FREEZING-RAIN) and anomaly(G)) implies weather_event(G) 5.0;
add [G] (precip(G, LIGHT-SNOW) and anomaly(G)) implies weather_event(G) 2.0;
add [G] (precip(G, SNOW) and anomaly(G)) implies weather_event(G) 12.0;
add [G] (precip(G, HEAVY-SNOW) and anomaly(G)) implies weather_event(G) 15.0;
add [g] ~weather_event(g) 1.0;
add [g] ~anomaly(g) 1.0;
add [g,y] ~precip(g,y) 0.5;

TABLE III: Excerpts from local (top) and cloud (bottom) MLN rules. Differences are in bold; the cloud MLN has defaulting rules and improved weight settings.

number of cells whose geographic location is nearest to a corresponding weather station.

The MLNs were constructed and evaluated on an hourly basis for Net2 and Net3 (another network containing information for 9242 cells) networks, for which we had access to weather information. A local MLN was used initially for Net2. This MLN represents a simple, default set of rules that might be applied generically to any given network (Table III). The results of this rule set are shown in Figure 3 (a). Cell groups that exhibit relatively high weather event probabilities are in fact anomaly groups as determined by topic modeling, but the lack of proper defaulting rules in the local MLN causes those groups to attain an elevated weather event probability uniformly across the time period. In contrast, Figure 3 (b) shows weather event probabilities for the Net3 dataset, using an MLN that has been modified to provide good resolution for those events. This is a case where the core MLN for Net3 becomes available in the cloud, and can be evaluated on other networks to determine whether the local MLN should be replaced. Figure 3 (c) shows the results of applying the Net3 MLN directly to the Net2 network. Using the mean entropy measure of MLN quality (lower entropy means higher quality), the local MLN for Net2 resulted in a mean entropy of 0.666. When the cloud MLN was applied to the same time series, it resulted in a mean entropy value of 0.5, which would mean that the cloud MLN serves as a better choice.

IV. Related Work

To the best of our knowledge, the concept of "diagnosis cloud," where network models for cellular network automation are shared, has not been addressed before. The concept of model sharing was applied previously, for example, in a cyber security context [4], [3], where models are shared for the purpose of detecting attacks across multiple cyber networks. Wang et al. [15] propose a knowledge transfer scheme for femtocell networks that takes historical network measurements from remote cells to address the challenges of data scarcity at an individual local cell. Based on the environment's similarity, the local cell leverages remote measurements to derive the local diagnostic model. A knowledge transfer scheme has been proposed for cognitive radio networks [17] to optimize radio channel selection for improving spectrum efficiency among densely populated, multi-hop base stations. The target base station combines its local channel selection table with tables from neighboring base stations. The location of a neighboring base station is used to decide if its table contributes to the local decision-making process to achieve optimal frequency reuse in a certain area.

V. Conclusions

This paper proposed a novel framework for sharing diagnosis knowledge across cellular networks. Our exemplary implementation that used topic modeling and MLNs on real datasets illustrated the benefits of sharing models across networks. While the proposed scheme is applicable to single network operators and across different operators, sharing can present risk in disclosing sensitive info. We operate on sensitive info (KPIs, cell metadata) locally, while the shared topic models consist of labeled topics associated with KPIs. The shared MLN models have rules associating weather events with anomalous behavior. To protect the KPIs and type of event, one could anonymize this info before or after processing. Secondarily, we support sharing of context (e.g., location). To preserve privacy, a user could forego context sharing. A consequence may be additional processing time spent considering a model that would have been eliminated, had context been considered. Even still, an operator may not want to divulge that an anomalous event occurred, a desire that would require stronger privacy preserving techniques that are beyond the scope of this work. Next steps include extensions to the framework to accommodate more models to increase diagnosis capabilities and preserve privacy when sharing across different operators.

REFERENCES

[1] D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," IEEE Signal Processing Magazine, 2010.
[2] G. F. Ciocarlie, C. Connolly, C. Cheng, U. Lindqvist, S. Novaczi, H. Sanneck and M. Naseer-ul-Islam, "Anomaly detection and diagnosis for automatic radio network verification," 6th International Conference on Mobile Networks and Management (MONAMI), September 2014.
[3] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, "Casting out demons: sanitizing training data for anomaly sensors," IEEE Symposium on Security & Privacy, 2008.
[4] G. F. Cretu, J. J. Parekh, K. Wang, S. J. Stolfo, "Intrusion and anomaly detection model exchange for mobile ad-hoc networks," IEEE Consumer Communications and Networking Conference, 2005.
[5] F. Forbes, G. Fort, "Combining Monte Carlo and mean-field-like methods for inference in hidden Markov random fields," IEEE Transactions on Image Processing,vol. 16, no.3, pp.824-837, March 2007
[6] N. Friedman, D Geiger, M Goldszmidt, "Bayesian network classifiers," Machine Learning, vol. 29, no. 2, pp.. 131-163, 1997.
[7] D. Heckerman, "A tutorial on learning with Bayesian networks," In Learning in Graphical Models, Ed. M. Jordan, Cambridge, MA: MIT Press, 1999.
[8] D. Hsu, "Algorithms for active learning", doctoral dissertation, 2010, retrieved from http://www.cs.columbia.edu/ djhsu/papers/dissertation.pdf.
[9] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi Software," PloS ONE, 2014.
[10] I. Katzela and M. Schwartz, "Schemes for fault identification in communication networks," IEEE/ACM Trans. Netw. vol. 3, no. 6, December 1995, pp. 753-764.
[11] Q. Pleple, "Perplexity to evaluate topic models", retrieved from http://qpleple.com/perplexity-to-evaluate-topic-models/
[12] T. Papai, S. Ghosh, and H. Kautz, "Combining subjective probabilities and data in training Markov logic networks," In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2012.
[13] M. Richardson and P. Domingos, "Markov logic networks," in Machine Learning, vol. 62, no. 1-2, 2006, pp. 107-136.
[14] E. Riloff, R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," In Proceedings of Sixteenth National Conference on Artificial Intelligence and eleventh Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI), 1999.
[15] W. Wang, J. Zhang, and Q. Zhang, "Transfer learning based diagnosis for configuration troubleshooting in self-organizing femtocell networks," IEEE Global Telecommunications Conference (GLOBECOM), Dec 2011, pp. 1-5.
[16] C.-H. Wu and P.C. Doerschuk, "Cluster expansions for the deterministic computation of Bayesian estimators based on Markov random fields," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 3, pp. 275-293, Mar 1995.
[17] Q. Zhao, D. Grace, "Agent transfer learning for cognitive resource management on multi-hop backhaul networks," Future Network and Mobile Summit (FutureNetworkSummit), 2013 , pp.1-10, 3-5 July 2013.