Combining ontological modelling and probabilistic reasoning for network management

Kasper Apajalahti [a], Eero Hyvönen [a], Juha Niiranen [b], and Vilho Räisänen [c]
[a] *Aalto University, Dept. of Computer Science, first.last@aalto.fi*
[b] *University of Helsinki, Dept. of Mathematics and Statistics, first.last@helsinki.fi*
[c] *Nokia Bell Labs, first.last@nokia-bell-labs.com*

Abstract. Advanced automation is needed in future mobile networks to provide adequate service quality economically and with high reliability. In this paper, a system is presented that takes into account the network context, analyses uncertain information, and infers network configurations by means of probabilistic reasoning. The system introduced in this paper is an experimental platform integrating a mobile network simulator, a Markov Logic Network (MLN) model, and an OWL 2 ontology into a runtime environment that can be monitored via a Resource Description Framework (RDF) -based user interface. In this approach, the OWL ontology contains a semantic representation of the relevant concepts, and the MLN model evaluates elements of uncertain information. Experiments based on a prototype implementation demonstrate the value of semantic modelling and probabilistic reasoning in network status characterization, optimization, and visualization.

Keywords: Network management, 5G, semantic modelling, probabilistic reasoning, ontologies, Markov Logic Networks

1. Introduction

Mobile networks are a part of a critical infrastructure, facilitating wireless access to Internet with all its services. The number of users, devices, and applications is expected to continue to grow [9]. A dramatic increase in the number of Internet of Things (IoT) endpoints is expected [24]. The advent of IoT is one of the most important drivers for fifth-generation (5G) mobile networks. Consequently, 5G networks need to cater for an increase in data volume and massive growth in the number of terminals served. Furthermore, 5G networks also need to support services with requirements of a new kind, such as ultra-low latency and high reliability [24].

Networks need to be configured optimally to provide customers with high service quality at a reasonable price. Manual network configurations employed in the past make per-cell optimization unfeasible in practice. Given the growth in the complexity of networks, more automation is needed from Operations and Support Systems (OSS) [19], a collective term for capabilities used for managing mobile networks.

State-of-the art operability in fourth-generation networks (4G) is based on the concept of Self-Organizing Networks (SON), which amounts to a set of closed-loop agent systems reacting to measurements, typically by means of a fixed model [19]. Each agent is an implementation of an operability use case. The challenge with this approach is creating and maintaining up-to-date models in view of geographic and temporal variety across cells in radio access networks.

The application of autonomic computing to various areas has been envisioned to the mobile networks with new architectures and software components [14,15,19]. These capabilities are based on architectures which support learning and adaptation to context-specific situations by means of Machine Learning (ML) algorithms, knowledge representation, and reasoning with knowledge bases [14,15,19].

1876-1364/17/$35.00

Machine Learning is expected to increase the level of automation in the OSS area, for example, by improving the analysis of traffic patterns and cell-related data to learn statistical correlations. Another application of ML is SON verification [8], where learning capability is used for identifying effective solutions. The output of a stand-alone ML system can be characterized as hypothetical in contrast to the deterministic results of a traditional rule-based system. This paper will argue that an effective approach to utilizing ML in the complex environment of future mobile networks involves both classical and probabilistic reasoning.

In this paper, a new approach is proposed to automate mobile network management by using statistical relational learning with a Markov Logic Network model (MLN) [32] for handling uncertainty in mobile network analysis. An OWL 2 ontology is used to complement the MLN model by providing global meaning and a semantic description of the system. The ontology is currently used as a semantic storage with a SPARQL interface for the MLN model and for an RDF-based faceted graphical user interface (GUI). In summary, the practical reason for combining semantic technologies with probabilistic reasoning is the decrease the complexity in monitoring of the MLN model and to making it dynamically modifiable.

To enable a human user to monitor and understand complex network management operations, intelligent storage and presentation of data is needed [22]. In this work, the OWL 2 model targets this issue with a semantic representation of the network state and the related automated configurations. The use of a formal knowledge model which supports automated reasoning reduces the need for case specific software design and implementation, and provides a mechanism for assessing the degree of consistency of the relevant models. Suitably chosen probabilistic reasoning accommodates partial and conflicting data, which would be challenging to address using subsets of First Order Logic (FOL) alone. Additionally, it has been argued that knowledge models bring benefits as a basis of future telecommunication systems both in view of systems design and from the perspective of value networks [30].

The paper is structured as follows: First, the approach is presented in Section 2 and an overview of the system is given in Section 3. After that, Section 4 presents the concept of MLNs and how they are applied to our implementation. Section 5 describes the OWL 2 model and Section 6 presents the RDF-based GUI used to monitor the MLN reasoning. Section 7 presents experiments with the system, such as the evaluation of the MLN reasoning, the affection of the MLN model size to execution times (MLN reasoning and SPARQL queries), and use case examples of the GUI. Finally, Section 8 discusses related work and Section 9 concludes the paper and presents ideas for future work.

2. Approach

The radio access network of a mobile network domain is composed of cells. In terms of hardware, cells have base stations having dedicated transmitter/receiver (TRX) units, each serving a sector of the cell. Each base station serves a number of terminal devices, known as User Equipment (UE) in industry terminology. Terminal devices may roam across cells and their traffic characteristics (uplink/downlink) traffic may vary depending on the services used.

Mobility and traffic characteristics exhibit diurnal and weekday variations. Additionally, the patterns vary according to the location of the cell in question; a suburban cell may be expected to have the highest loading outside of business hours, whereas the opposite is true for a downtown cell in a business district. There are longer term trends relating to varying usage patterns of services and residential/business user densities. There are also short-term variations due to special events like concerts or sports events.

Given the complexity and variability of network loading, it is difficult to try to optimize network status as one entity. In 4G networks, SON agents execute specific use cases in a limited scope (one or a few cells), performing automated configurations based on measurements. A run-of-the-mill implementation of SON agent utilizes a fixed model. This works well when the model reflects the situation of the cell, but becomes a bottleneck in tailoring the behaviour of the agent on cell level when cell characteristics vary.

Automated adaptation is the next step beyond the fixed model. An approach utilizing Case-Based Reasoning (CBR) has been utilized [33,7]. Interpolation across cases is an enhancement, but is nevertheless limited by the case base.

We study the use of probabilistic reasoning in coping with complexity and adaptability. Instead of using a rule or case base, this system is driven by formulae for reasoning performed over facts about system status. In addition to raising the abstraction level of implementations, a suitably chosen method also supports incomplete and conflicting data, which classical reasoning cannot address.

3. System Overview

3.1. Components

The architecture view of the system used in this article is depicted in Figure 1. The bottom component is the data source, e.g. cellular network or a network simulator. In this paper, we demonstrate the system with a Long Term Evolution (LTE) simulator.

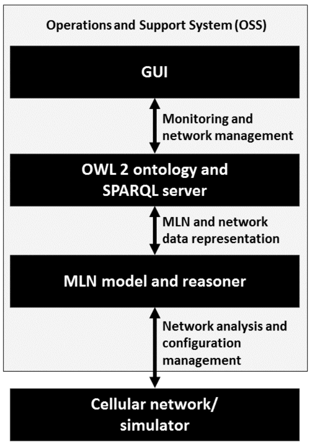

Operations and Support System (OSS)
GUI
Monitoring and network management
OWL 2 ontology and SPARQL server
MLN and network data representation
MLN model and reasoner
Network analysis and configuration management
Cellular network/ simulator

Fig. 1. System architecture.

The simulator interacts with the MLN model which analyses the network data and provides data for decision support in the configuration management. The OWL 2 ontology and reasoner are used to create a semantic representation of the network and MLN data and to describe the data as a graph. The ontology is stored in a SPARQL server providing access to the ontology. The GUI is used to monitor and manage the system. In a real network, the three topmost components would belong to an OSS system which manages the network.

3.2. Scenario

The simulation scenario consists of a small urban area (a diameter of 5 km) with 2000 terminals and 32 sectors. In the context of LTE, term "cell" is also used as a synonym for a sector of a base station. The simulator sends performance data that contains measurements from key performance indicators (KPIs) for various cases. KPIs utilized in the system are the number of connected terminals (CUE) per cell, the channel quality indicator (CQI) distribution vector for measuring the signal quality of a cell, and the radio link failures (RLF) for measuring the amount of connection failures per cell. The simulator receives configuration data that contain possible changes in the transmission power (TXP) and angle (remote electrical tilt, RET) of a cell antenna. The data are sent periodically in 15 minute intervals in simulation time.

The MLN model analyses the CUE, CQI, and RLF measurements and infers posterior marginal probabilities for potential network configuration changes in order to optimize the CQI and RLF metrics. A fit of model parameters of the MLN reasoner is performed to historical performance data and past executed configuration actions. The OWL 2 ontology is constructed by transforming the MLN reasoner model into a semantic representation that can be utilized as an RDF graph in a SPARQL endpoint. An operator interface for managing the system and the underlying mobile network is implemented on top of the SPARQL endpoint with HTML5 based GUI.

3.3. Data sequence

Figure 2 depicts the data sequence between system components starting with a measurement report from the LTE simulator on the left. The simulator sends the MLN reasoner performance management (PM) data (1) periodically. The reasoner processes data into its evidence (classified KPI values, such as low, moderate, and high CQI) and infers the probabilities for network actions (2) from them. Then, the reasoner sends a set of action proposals (configurations with high probabilities) to the simulator (3). In addition, the MLN reasoner sends further data, including evidence, formulae, and action proposals to the ontology processor (4), which processes the data into RDF and populates the ontology with new network- and MLN-related instances (5). The RDF graph is then uploaded into a SPARQL server based on Fuseki[1] (6). The server dynamically updates facets from the ontology with SPARQL update scripts (7). The user can monitor the system via the GUI that interacts with the SPARQL endpoint in order to retrieve network- and MLN-related data from the ontology (8) and to update MLN formulae (9). Similarly, the MLN model queries the SPARQL endpoint in order to retrieve an updated model (10).

[1] https://jena.apache.org/documentation/serving_data/

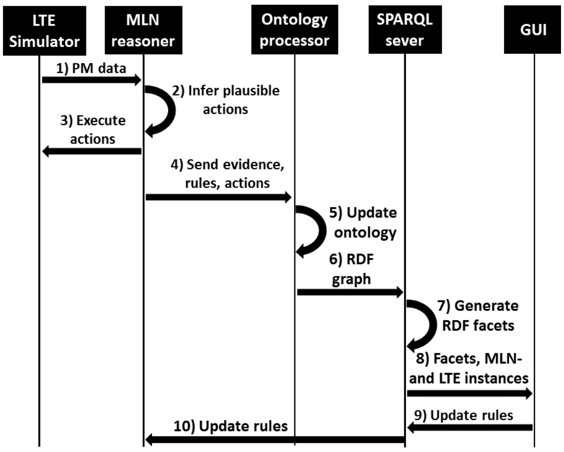


Fig. 2. Data sequence diagram for managing a mobile network (simulator) with system components.

4. Markov Logic Network model

4.1. Definition

MLNs allow uncertain and contradictory knowledge in a first-order logic (FOL) subset model by introducing a weight parameter for each formula in the FOL subset knowledge base. The weighted set of formulae defines a template for a Markov network, where the features and feature weights are determined by the formulae and formula weights.

Definition 4.1 *A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a first-order formula and w_i is a real-valued weight parameter. Together with a set C of constant terms, over which the formulae in L are applied, it defines a Markov network $M_{L,C}$ with a binary variable for each possible grounding of each predicate appearing in L and a feature for each possible grounding of each formula in L. The value of the feature corresponding to a grounding of formula F_i is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is w_i, the weight associated with F_i in L.*

Each state of the variables in a Markov network $M_{L,C}$ represents a possible world, for example a truth assignment for each of the ground atoms for (L, C). The probability distribution over possible worlds $x \in \mathcal{X}$ specified by $M_{L,C}$ is defined in Eq. (1).

$$P_{M_{L,C}}(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right), \qquad (1)$$

where $n_i(x)$ is the number of true groundings of F_i in x and Z is a partition function given by $Z = \sum_{x\in\mathcal{X}} \exp\left(\sum_i w_i n_i(x)\right)$. Intuitively this means that the weights of the true ground formulae give the logarithmized factors of the distribution function. If two worlds differ only on a single ground formula, then the weight of the formula gives the logarithmic odds of choosing one world over the other.

4.2. Inference

Given an MLN L and a set of constants C, the most likely state of the world can be inferred, $\hat{x} \in \mathcal{X}$, according to the joint probability given in Eq. (1), such as the most likely truth assignment for the variables in $M_{L,C}$. Also, the marginal probability for each of the variables can be inferred.

However, a typical inference task is to deduce the most likely state or marginal distributions for a subset of the variables, called query variables, using the values of (some of) the rest of variables as evidence. Given the values x_E of a set of evidence variables $X_E \subset X$, the most likely state $\hat{x}_Q$ for query variables $X_Q \subset X$ is inferred according to Eq. (2).

$$\begin{aligned} &P(X_Q = x_Q | X_E = x_E) \\ &\quad = \frac{P(X_Q = x_Q, X_E = x_E)}{P(X_E = x_E)} \\ &\quad = \frac{\sum_{x\in\mathcal{X}_Q\cap\mathcal{X}_E} P_{M_{L,C}(X=x)}}{\sum_{x\in\mathcal{X}_E} P_{M_{L,C}}(X = x)}, \end{aligned} \qquad (2)$$

where $\mathcal{X}_\varphi$ is the set of worlds where $X_\varphi = x_\varphi$ holds for $\varphi = E$ and $\varphi = Q$.

As Richardson and Domingos show [32], exact inference over an MLN model is infeasible in practice. Instead, they introduce an efficient approximation algorithm for this problem using stochastic simulation.

4.3. Weight learning

The weights of an MLN model can be learned from one or more databases. A database is effectively a Boolean vector stating the observed truth values for each ground predicate. Given a database (or databases), the most likely weight values are those that maximize the probability of the database given in Eq. (1). The standard method for maximum likelihood estimation (MLE) is the gradient descent method. However, computing the gradient of the probability in Eq. (1) requires computing the estimated number of true groundings for each formula, which is infeasible, as shown by Richardson and Domingos [32]. They propose a solution using gradient descent for optimizing the pseudo-likelihood [4] of the ground Markov network.

Typically, in optimizing the weights of an MLN, it is known in advance which of the predicates in the model will be used for querying and which will be used as evidence. In that situation, Lowd and Domingos [23] propose a discriminative learning solution where they use a gradient descent algorithm to optimize the likelihood of the conditional probability, as given in Eq. (2), instead of the full joint probability, given in Eq. (1). They show that this solution outperforms the solution based on pseudo-likelihood.

4.4. Application in OSS setting

4.4.1. Structure of the model

The MLN model of the system is defined in terms of three types of predicates:

- *Context predicates* reflect the current status of the relevant network scope and its environment. A context predicate can indicate, for example, that a KPI value for a cell is currently below the acceptable level, or that two cells are neighbors in the network topology.
- *Objective predicates* indicate required changes to KPI values to achieve performance targets defined by the operator. For example, an objective predicate can indicate that a particular KPI value for some cell is too low and needs to be increased.
- *Action predicates* indicate changes to network configuration parameter values.

Each predicate represents an attribute of a cell in the network or a relation among the cells. The domain of a predicate can be either the set of cells or an n-ary Cartesian product of the set of cells.

The MLN model is composed of formulae with these predicates. The formulae are intended to describe a correlation between a set of $Objectives$ and a set of $Actions$ in a certain $Context$. A typical inference task is to query for appropriate actions using the current context data and objective requirements as evidence. Therefore, the formula format is defined as

$$\mathcal{C} \Rightarrow (\mathcal{O} \Leftrightarrow \mathcal{A}). \tag{3}$$

Above, $\mathcal{C}$, $\mathcal{O}$ and $\mathcal{A}$ are sets of one or more context, objective and action predicates, respectively.

The values of the context predicates are computed from the PM data measured from the LTE network. For numerical data, the measurements need to be first classified, for example to low, moderate, and high classes. When values for the action predicates are inferred, the values of the objective predicates are derived from the classified PM data according to performance requirements set by the network operator. For example, if the CQI value measured for a cell is classified as "low", the derived objective could be to increase the CQI for that cell. On the other hand, when the weights of the MLN model are learned from a database of historical PM and configuration data, the objective predicate values are computed by analyzing realized changes in PM values. In a similar way, the values of the action predicates are computed from changes in configuration data.

After the weights are defined for the MLN model (either learned iteratively or set by an expert), the most likely state of the action predicates can be inferred. Furthermore, the network can be configured according to the inferred action predicates values, because they give the best possible solution with respect to the model. The model weights can still be adjusted by configuring the network with respect to the marginal probabilities of actions and monitoring performance changes. In this manner, the model is dynamically adapted to the network situation.

4.4.2. Inference example

To illustrate the usage of the MLN model, the inference is demonstrated with an example.

Example Let L be a simple MLN model consisting of the weighted formulae defined in Table 1.

w_i	F_i
2.4	$I(c,Rlf,High) \Rightarrow (O(c,Rlf,Dec) \Leftrightarrow A(c,Txp,Inc))$
0.5	$I(c,Rlf,High) \Rightarrow (O(c,Rlf,Dec) \Leftrightarrow A(c,Ret,Dec))$
0.9	$(N(c,d) \wedge I(c,Cqi,Low)) \Rightarrow (O(c,Cqi,Inc)$ $\Leftrightarrow (A(c,Ret,Dec) \wedge A(d,Txp,Inc)))$

Table 1

Examples of weighted formulae in the MLN model.

Here the variables c and d denote cells in the mobile network. I is a context predicate indicating KPI category, N is a context predicate indicating the neighborship of two cells, O is an objective predicate indicating a change in a KPI value and A is an action predicate indicating change in a parameter value. Suppose that there is a mobile network with two neighbor cells named $C1$ and $C2$ and that low CQI value for cell $C1$

and a high RLF value for $C2$ is measured. This information is used to infer proper configuration actions to get the CQI and RLF values to a normal level. The MLN reasoner is used to query the MLN model L for marginal probability distributions for action proposals $A(c,Txp,Inc)$, $A(c,Txp,Dec)$, $A(c,Ret,Inc)$, and $A(c,Ret,Dec)$ for each cell c given the evidence:

$$\begin{aligned}E = \{&N(C1,C2), I(C1,Cqi,Low),\\ &I(C2,Rlf,High), O(C1,Cqi,Inc),\\ &\qquad\qquad O(C2,Rlf,Dec)\}\end{aligned}$$

An example of the reasoning output is shown in Table 2, which shows inferred marginal probabilities for cell configurations given the model L and the evidence E. The output indicates that decreasing RET for cell $C1$ and increasing TXP for cell $C2$ are the most likely actions to achieve the objectives according to the model.

Action	$P(Action \mid L,E)$
$A(C1,Txp,Inc)$	0.35
$A(C1,Txp,Dec)$	0.30
$A(C2,Txp,Inc)$	0.70
$A(C2,Txp,Dec)$	0.14
$A(C1,Ret,Inc)$	0.28
$A(C1,Ret,Dec)$	0.41
$A(C2,Ret,Inc)$	0.29
$A(C2,Ret,Dec)$	0.41

Table 2
Marginal probabilities for cell configurations.

4.4.3. Generating formulae for the model

The CQI, CUE, and RLF KPIs are measured for each cell. RLF and CQI characterize the performance and CUE is an indicator of the cell load. To facilitate usage across varying load levels, RLF value is normalized by dividing it with the CUE value. Furthermore, to get a scalar value from the CQI distribution vector, the average channel efficiency is computed using the CQI vector values as weights for the channel efficiency values of each CQI class as defined in the technical specification [13]. For simplicity, these two aggregate metrics are referred to as RLF and CQI, respectively.

For the MLN model these three metrics, CUE, RLF and CQI, were classified as low, moderate, or high. The context predicates of the model describe classified values for these metrics for a cell variable. Furthermore, another context predicate was introduced describing the neighborship status of pairs of cells.

Objective predicates were introduced for the RLF and CQI metrics. For inference, the values of the predicates were computed according to operator goals from the context predicate values so that if the RLF value was high, the objective was to decrease the value, and if the CQI value was low, the objective was to increase the value. For weight learning the objective predicate values were computed from the realized changes in the numeric RLF and CQI values.

For the network configuration parameters, RET and TXP, action predicates are introduced to indicate a fixed-size decrease or increase of the value.

Algorithm 1 shows how weighted formulae are generated. The pseudocode illustrates updating the set of formulae; after a combinatorial generation of formulae, all unachievable formulae are removed from the model. A formula is unachievable, if a KPI is included in the objective (e.g. decrease RLF) but not in the context (e.g. high RLF). A formula is also unachievable, if it has a KPI value in the context that violates an operator goal (e.g. high RLF), but has not a corresponding objective to change that value (e.g. decrease RLF).

Algorithm 1. Generating weighted formulae for the model

```
Create all combinations of formulae with given
KPIs, operator goals, and configuration parameters
for formula in model do
    If KPI in objective not contained in context
        Then Remove formula from the model
    If KPI in context does not achieve operator goal
    and corresponding objective does not exist
        Then Remove formula from the model
end for
```

In addition to creating weighted formulae, some constraints are added to the model, such as: 1) each KPI must have exactly one value, 2) a KPI value can not be both increased and decreased, 3) a parameter value can not be further decreased (increased) from its minimum (maximum) level.

5. OWL 2 Model

The OWL2 ontology[2] has been designed to support GUI based on reasoning. Because of this, the ontology is minimal and use case driven rather than an all-encompassing ontology of the entire mobile network

[2] https://www.w3.org/TR/owl2-overview/

domain. The ontology contains mobile network concepts such as KPIs and cells on the one hand, and MLN model concepts like rules, actions, and parameters on the other. The MLN model supports learning of the most effective action to achieve a particular goal in a specific network context. The results of learning and as well as evidence (input data used by MLN) are reflected in the ABox of the OWL2 ontology.

The MLN model concepts are linked to mobile network concepts, which helps exploration. Furthermore, this link clarifies the interrelations between the MLN model on the one hand and network state and MLN action inference on the other. An added benefit is that the governance of the MLN model is easier as the network-related metadata is defined for it.

5.1. Evidence and action proposals

In Figure 3, mobile network concepts are described together with MLN evidence and action proposals. The *Cell* is the most fundamental class in the model and has properties *hasKPI* to its performance metrics (instances of the class *CellKPI*) and *hasParameter* to its configuration parameters (instances of the class *CellParameter*). *CellKPI* has a crisp description for its value (*KPIValue*), such as low, moderate, or high, which is defined in the MLN model. Also, according to the MLN model, a KPI might have an objective defined using the class *KPIObjective*. *KPIObjective* in turn has some *EventImpact* defining the direction of the change of an impact (increase or decrease). The *CellParameter* can have an *ActionProposal*, if the parameter needs to be adjusted with respect to the MLN inference. The *ActionProposal* also has a relation to an instance of the class *EventImpact* to describe its impact direction.

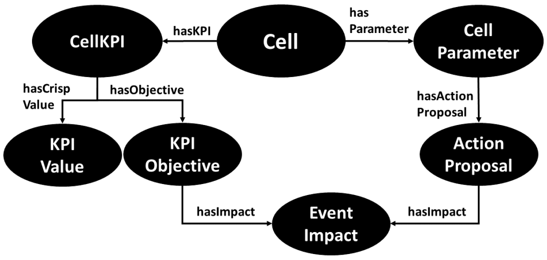


Fig. 3. MLN evidence, action proposals, and their cell-related concepts in the ontology TBox.

5.2. Formulae

In addition to the evidence and action proposals, the weighted formulae of the MLN model are represented with concepts and mapped to mobile network concepts in the OWL 2 TBox, as shown in Figure 4. In the OWL 2 model, the "rule" term is used as a synonym for a formula. The *MLNRule* class defines a formula, which has a numerical value *hasRuleWeight* defining its weight and relations to formula classes *RuleContext*, *RuleObjective*, and *RuleAction*. The figure also depicts that the formula classes are bound to network classes *CellParameter* and *CellKPI*. A *RuleAction* has a relation to an instance of a *CellParameter* (such as *Txp*) and that is bound to an instance of *EventImpact* class (such as *Increase*). Similarly, *RuleObjective* and *RuleContext* have relations to *CellKPI* instances, which in turn contain instances of an *EventImpact* (in case of *RuleObjective*) and *KPIValue* (in case of *RuleContext*).

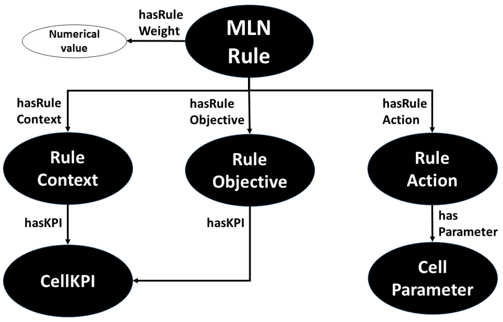


Fig. 4. Rules (same as formulae in the MLN model) and their cell-related concepts in the ontology TBox.

5.3. Ontology query

To demonstrate the ontology, Listing 1 shows RDF triples representing data regarding the cell with ID 1. The cell has information about the neighboring cells, amount of neighbors, KPI instances (:hasCqi, :hasCue, and :hasRLF are subproperties of :hasKPI), parameter instances (:hasRet and :hasTxp are subproperties of :hasParameter) and facet values that are generated on-the-fly in the SPARQL server. The facets designed for the cell and MLN formula specific data are explained in more detail in the next section.

```
:Cell1 a :Cell,
  owl:NamedIndividual ;
:hasCellId 1 ;
:hasAmountOfNeighbors 3 ;
```

```
:hasNeighbor :Cell25,
  :Cell2, :Cell3 ;
:hasCqi :Cqi1 ;
:hasCue :Cue1 ;
:hasRlf :Rlf1 ;
:hasRet  :Ret1 ;
:hasTxp :Txp1 ;
:hasFacetCqi :LowCqi ;
:hasFacetCue :LowCue ;
:hasFacetNeighbors :Few ;
:hasFacetRlf :HighRlf .
```

Listing 1: RDF representation of Cell 1 and its related data.

The ontology can be queried with a SPARQL query pattern shown in Listing 2. The query returns cell-specific data for cells fulfilling the given facet selections (high RLF is chosen in this example). The return variables of the query are cell URI (?*c*), link property (?*link*) to a related node (link to a KPI or parameter instance of a cell), column name in the GUI (?*col*), or column value (?*val*).

The query has three union parts, which find column properties (:hasColumnProperty and :hasColumnDataProperty) that specify instances represented in the tabular view of the GUI which is explained in more detail in the next section. The first and second union parts find column properties directly related to the cell, such as :hasFacetRlf and :hasAmountOfNeighbors. The last union part finds instances from the related nodes, such as an objective to increase a KPI value of a cell. The third part also contains a filtering condition that prevents search results from neighboring cell instances.

```
SELECT DISTINCT ?c ?link ?col ?val
{
  ?c a :Cell .
  ?c :hasFacetRlf :HighRlf .
{
  ?col rdfs:subPropertyOf
    :hasColumnProperty .
    ?c ?col ?val .
} UNION {
  ?col rdfs:subPropertyOf
    :hasColumnDataProperty .
  ?c ?col ?val .
} UNION
{
  ?c ?link ?k_or_p .
  FILTER NOT EXISTS
        {?k_or_p a :Cell}
  ?k_or_p ?col ?obj_or_act .
  ?obj_or_act ?has_impact ?val .
  ?has_impact rdfs:subPropertyOf
    :hasColumnProperty .
} }
ORDER BY ?c
```

Listing 2: SPARQL query for cell-specific data with facet selection high RLF.

The SPARQL query returns all cells having a high RLF value. Cell 1 is fulfilling this condition and it is shown as an example in the table 3. Each column depicts a SPARQL return variable (cell URI (?*c*), link property (?*link*), column name (?*col*), or column value (?*val*)).

Cell (?*c*)	Link (?*link*)	Column (?*col*)	Value (?*val*)
:Cell1		:hasAmountOf Neighbors	3
:Cell 1		:hasCellId	1
:Cell1		:hasFacetCqi	:LowCqi
:Cell1		:hasFacetCue	:LowCue
:Cell1		:hasFacetRlf	:HighRlf
:Cell1	:hasCqi	:hasObjective	:Increase
:Cell1	:hasRlf	:hasObjective	:Decrease

Table 3

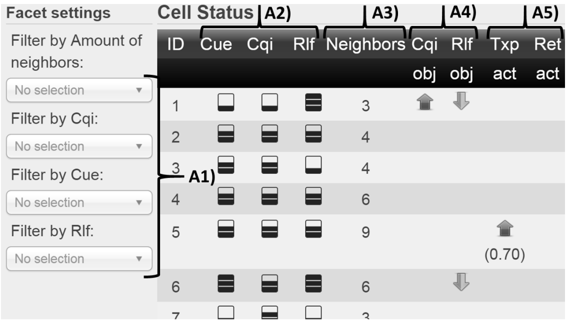
A snippet of SPARQL query results for the example query. Cell 1 fulfils the facet condition high RLF.

The SPARQL query pattern shown in Listing 2 can be used with slight modifications to retrieve MLN formula-specific data. MLN formulae have own column properties defining retrieved values in a different tabular view and facets to filter search results.

6. Graphical User Interface

The system presented in this paper is monitored via a faceted RDF-based GUI [1] that visualizes the ontology instances processed using the MLN model and supports exploration. The purpose of the GUI is to provide the end-user with informative and interactive tools for evaluating the MLN functionality. Thus, views are implemented to present MLN reasoner-related cell states and MLN formulae.

Figures 5 and 6 show cell states in two alternative visualizations: tabular and graph-oriented. With these views, the user examines how MLN evidence (KPI values and KPI objectives) affects the MLN reasoning outcome (action proposals). In the tabular visualization (Figure 5), the rows depict cell instances and the columns their attributes, such as classified KPI values (A2), KPI objectives (A3), amount of neighbors (A4), and action proposals (A5). The data describe the current states of cells and thus are based on the latest PM report from the simulator. In the graph visualization (Figure 6), positioning of cells corresponds to their Cartesian coordinates in the simulator. The arcs in the graph depict neighborship relations between cells and the size of the node depicts the classified value for some KPI in the evidence (CUE, CQI, or RLF). In the figure, RLF value has been selected as the node size from the settings (B2). The color of a node indicates the desired impact of a KPI objective or an action proposal (in the regarding grayscale figure, light gray indicates no action and dark gray an increase). In the figure, TXP action has been selected for the node color and it can be dynamically changed between objectives and action proposals from the settings (B3). Both in the tabular and graph visualization user can interactively browse cells with similar states by selecting facet values (A1 in Figure 5 and B1 in Figure 6), such as the amount of neighbors and classified KPI values.



Facet settings

Filter by Amount of neighbors: No selection ▾

Filter by Cqi: No selection ▾

Filter by Cue: No selection ▾

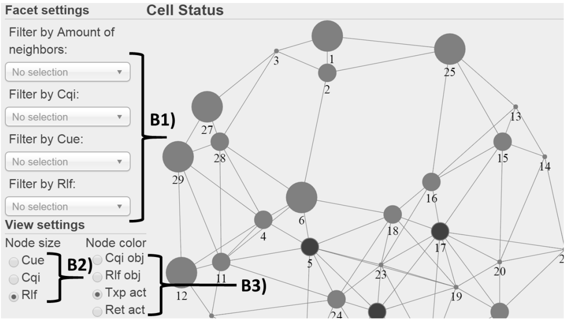
Filter by Rlf: No selection ▾

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⬆	⬇		
2	▄	▄	▄	4				
3	▄	▄	▁	4				
4	▄	▄	▄	6				
5	▄	▄	▄	9			⬆ (0.70)	
6	█	▄	█	6		⬇		
7	□	▄	□	3				

Fig. 5. A faceted view for a cell-specific tabular visualization

Figure 7 depicts the weighted formulae in a tabular view by dividing each formula into a formula weight (C5) and the formula classes defined earlier: context (C2), objective (C3), and action (C4). The user examines this view to learn the contents of the formulae and may modify or create formulae in order to change the behaviour of the MLN reasoner. For example, modification can be done by removing a formula or by changing its weight (C6). Facets (C1) in this view are generated as a combination of formula classes (contexts, objectives, and actions) and their objects (CQI, RLF, CUE, TXP, and RET).



Facet settings

Filter by Amount of neighbors: No selection ▾

Filter by Cqi: No selection ▾

Filter by Cue: No selection ▾

Filter by Rlf: No selection ▾

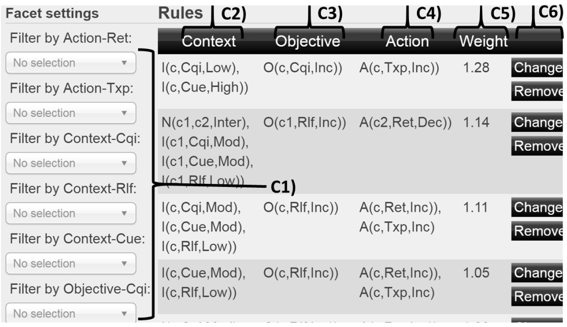
View settings

Node size: ○ Cue ○ Cqi ● Rlf

Node color: ○ Cqi obj ○ Rlf obj ● Txp act ○ Ret act

Cell Status

Fig. 6. A faceted view for a cell-specific graph visualization.



Facet settings

Filter by Action-Ret: No selection ▾

Filter by Action-Txp: No selection ▾

Filter by Context-Cqi: No selection ▾

Filter by Context-Rlf: No selection ▾

Filter by Context-Cue: No selection ▾

Filter by Objective-Cqi: No selection ▾

Rules

Context	Objective	Action	Weight	
I(c,Cqi,Low), I(c,Cue,High))	O(c,Cqi,Inc))	A(c,Txp,Inc))	1.28	Change Remove
N(c1,c2,Inter), I(c1,Cqi,Mod), I(c1,Cue,Mod), I(c1,Rlf,Low))	O(c1,Rlf,Inc))	A(c2,Ret,Dec))	1.14	Change Remove
I(c,Cqi,Mod), I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.11	Change Remove
I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.05	Change Remove

Fig. 7. Faceted view for MLN formulae.

7. Experiments

7.1. Statistical evaluation of the MLN reasoner

The MLN model is used to optimize the performance of a simulated LTE network by adjusting the parameters of each cell simultaneously with respect to PM data retrieved from the simulator in 15 minute intervals (simulation time). Initially, all formula weights of the MLN model were set to zero. The weights were updated after every 48 measurement rounds using the measurements as training data. The Alchemy 2.0 software package [21] is used for the MLN inference and weight learning. The Alchemy implements the marginal inference and discriminative weight learning algorithms described in Sections 4.2 and 4.3.

Figure 8 shows the changes in the total number of RLFs in the LTE network as the model is updated. From the figure can be seen that in the beginning the number increases, as the model weights have not yet been learned. After the weights are first updated, the number begins to rapidly decline and stabilizes after a number of iterations.

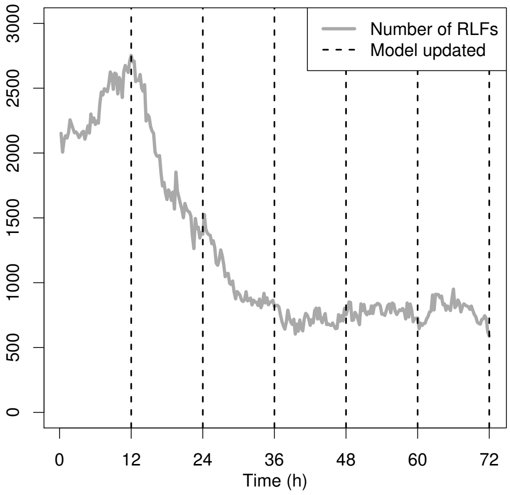


Fig. 8. Total number of radio link failures in a 15 minute interval. The vertical dotted lines indicate an update of the model weights.

7.2. Rule model statistics

Table 4 describes the results of some experiments on the impact of model sizes on MLN reasoning and SPARQL query times. In the table, *model 1* is constructed with a combinatorial generation of formulae from which unachievable formulae are removed, as defined in Algorithm 1. *Model 2* presents the model after formulae with zero weights are removed from the *model 1*. *Model 3* is created by removing formulae having weights less than $|0.1|$ from *model 2*. Apparent from the results, the number of formulae decreases drastically when the model is refined, and finally the model size is only 32 % compared with the original *m*odel 1. Also, a decrease in the model size has an impact on the reasoning and SPARQL query times. The reasoning time has decreased from 12.1s to 8.9s (from *model 1* to *model 3*. This is a slight improvement to the system, although 12.1s is already feasible as the reasoning is run only once every 15 minutes in simulation time (and would also be run every 15 minutes in a real network management system). The SPARQL query time has dropped from 1.6s to 0.8s which enhances the usability of the GUI as the page contents need to be frequently updated in a faceted GUI.

	Number of formulae	Reasoning time	Query time
model 1	1920	12,1s	1,6s
model 2	1063	11,3s	1,0s
model 3	614	8,9s	0,8s

Table 4

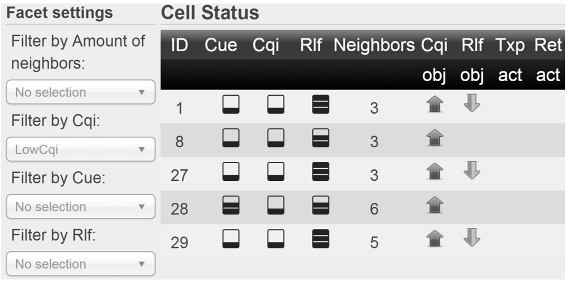
Statistics of model sizes and their impact on MLN reasoning and SPARQL query times.

7.3. Browsing activities in the GUI

To demonstrate the usage of the GUI, a random simulation round is taken from the system and the MLN reasoner and its model is explored via the GUI. First, two use cases are shown for 1) browsing cell states in a table and 2) graph visualization. Then, an example is shown on how modifying the model affects the reasoning outcome.

7.3.1. Observing cell states in the table view

The cell-specific MLN reasoning data of the given simulation round are presented in Figure 5. This view can be further examined interactively by selecting facet values. Figure 9 shows that cells having low CQI values (selected as a facet value) also have objectives to increase these values. In the same manner, from Figure 10 it can be concluded that high RLF values imply objectives to decrease the RLF values. These discoveries relate to the data preprocessing logic of the MLN model (operator goals that need to be achieved), as some classified KPI values (such as low CQI and high RLF in our scenario) trigger KPI objectives into the evidence.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: LowCqi

Filter by Cue: No selection

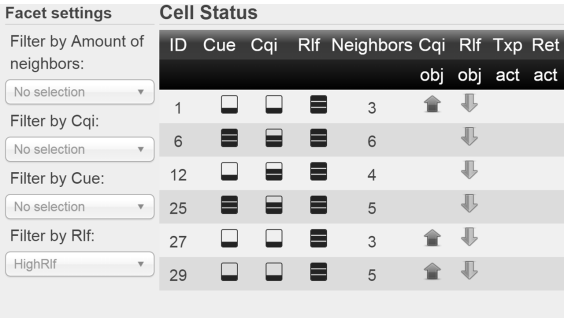
Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⬆	⬇		
8	▁	▁	▄	3	⬆			
27	▁	▁	█	3	⬆	⬇		
28	▄	▁	▄	6	⬆			
29	▁	▁	█	5	⬆	⬇		

Fig. 9. Cells having low CQI.

Another finding can be done from the table by selecting many neighbors (9 or more) as a facet value. Figure 11 shows the results for this selection and as it can be seen, all the three cells having many neighbors also have high probabilities (more than 0.7) for increasing their TXPs. As explained in Section 4, the



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: No selection

Filter by Cue: No selection

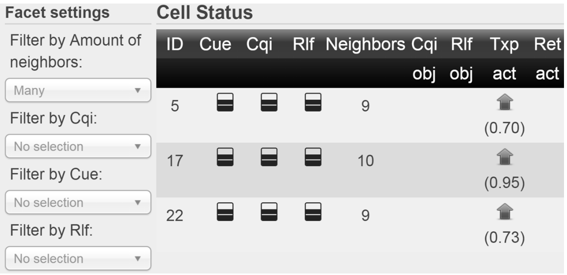
Filter by Rlf: HighRlf

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	↑	↓		
6	█	▄	█	6		↓		
12	▁	▄	█	4		↓		
25	█	▄	█	5		↓		
27	▁	▁	█	3	↑	↓		
29	▁	▁	█	5	↑	↓		

Fig. 10. Cells having high RLF.

MLN model and its formulae consider the states of neighboring cells as well, which might imply that the number of neighbors affects the probabilities of the action proposals in this scenario. Another aspect to observe from this figure is that these cells have neither an objective to increase CQI nor decrease RLF although actions are proposed to them. This also indicates how dependent cells are on their neighbors.



Facet settings

Filter by Amount of neighbors: Many

Filter by Cqi: No selection

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
5	▄	▄	▄	9			↑ (0.70)	
17	▄	▄	▄	10			↑ (0.95)	
22	▄	▄	▄	9			↑ (0.73)	

Fig. 11. Cells having many neighbors.

7.3.2. *Observing cell states in the graph view*

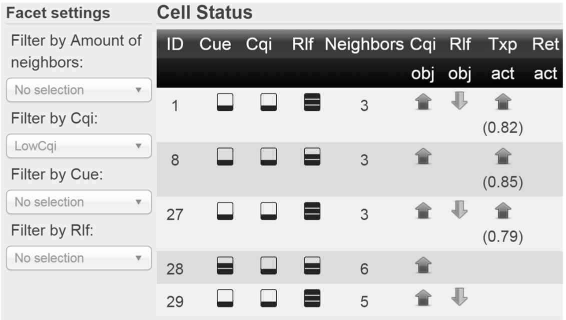
Figure 6 showed cell states in a graph visualization, in which node sizes indicate RLF values and node colors action proposals for TXP parameter. Observing this graph more closely, it can be seen that four of the six cells having high RLF values (cells 1, 25, 27, and 29) are located at the edge of the network and have only a few neighbors (five or less). High RLF values of these cells are most likely caused by terminals located at the edge of the network coverage area. Although the whole simulation area is populated by terminals, cells cannot cover the whole area with high signal strength from current cell locations. Thus, these cells most probably have high RLF values regardless of their configuration (even if coverage areas are expanded, new terminals will occur at the new edge of the coverage area). Moreover, the figure shows that MLN reasoner neither proposes configuration changes to these cells nor to their neighbors. This might indicate that the model has learned that the performance is not improved with any action for cells with given characteristics.

7.3.3. *Examining and modifying the MLN model*

For demonstration purposes, a weight of the following formula in the MLN model is modified:

$$I(c, Cqi, Low) \Rightarrow (O(c, Cqi, Inc) \iff A(c, Txp, Inc) \tag{4}$$

The weight of this formula is changed from 0.41 to 4.0. The Figure 12 shows cell states for cells having low CQI values after MLN reasoner has recalculated the action probabilities. Compared with Figure 9, new action proposals have been generated for cells 1, 8, and 27, which implies that the weight update had an effect on the functionality of the reasoner.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: LowCqi

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	↑	↓	↑ (0.82)	
8	▁	▁	▄	3	↑		↑ (0.85)	
27	▁	▁	█	3	↑	↓	↑ (0.79)	
28	▄	▁	▄	6	↑			
29	▁	▁	█	5	↑	↓		

Fig. 12. Updated states (actions re-inferred after model update) for cells having low CQI.

8. Related Work

The use of probabilistic reasoning in telecommunications has been studied previously. For example, Bayesian networks (BN) have been investigated in automatic network fault management [3,18] and in configuration evaluation [11]. MLNs have been used earlier to diagnose anomalous cells [8] in the network. Ontological modelling has been used together with BNs to evaluate network management activities in [10], which proposes using an ontology to describe domain-specific knowledge which is then utilized to dynamically generate a BN for a context-specific probabilistic evaluation task. To the best of our knowledge, statistical relational models (such as MLNs or a combination of an ontology and BNs) have not been applied earlier for cell configuration tasks specifically. Also, MLNs provide a template from which multiple Markov networks can be analysed. This approach provides more flexibility than a basic Bayesian or Markovian network model.

In the fields of pervasive computing and ambient intelligence, probabilistic reasoning and ontologies have been studied in several works. For example, MLNs are applied to context-aware decision processes in smart home environments [6]. The ontology is used to interpret and recognize situations from incoming data streams and an MLN model is dynamically constructed with respect to ontological knowledge. The MLN model is then used for decision making (smart home activities) from incomplete information [6]. Another project ([16,17]) examined human activity recognition from sensor data; first with MLN [16] and later with a combination of an ontology and a log-linear DL model (a model integrating description logics with probabilistic log-linear models [26]) [17]. Human activity recognition is also considered in another project [31] that uses an ontology to recognize potential human activities and statistical analysis to examine their confidence level [31].

In other problem domains, experiments in combining an ontological approach with probabilistic methods have been investigated. For example, BN-specific projects have use cases for medical decision support [34], financial fraud detection [5], and instance matching in a geological domain [27]. MLNs have been applied with semantic technologies in problem domains for ontology matching [25] and for natural language processing, where ontological concepts are extracted from text [12,28]. Aforementioned studies are not addressing cellular networks or network management tasks and thus they are not fully comparable. Yet, they provide similar technologies and address tasks, such as decision making support and graph analysis, that could be adapted and examined also in our work.

Another statistical relational model used for similar problems as MLN is the Probabilistic Soft Logic (PSL) [20]. It is based on First Order Logic, and provides weighted formulae and probabilistic inference. One application for a combination of ontologies and PSL is in analysing semantic similarities between natural language sentences. [2] Another project utilizes these techniques to extract a knowledge graph from text by using ontologies for representing domain-specific constraints and PSL to infer the most probable meaning for the text as a graph. [29] It has been argued that PSL is more efficient than MLN [29]. Our experiments indicate that MLN analysis can be applied to mobile networks by analysing a limited scope. The analysis of an entire network would in any case be futile for a system having the complexity and dynamicity of a mobile network domain. The comparison of PSL and MLN should be examined more thoroughly in order to conclude their difference in our case.

9. Conclusion and Future Work

This paper presented an experimental network management platform that uses MLN to analyse uncertain information from the LTE simulator and to infer suitable actions for the simulator. Along with the MLN model, OWL 2 ontology is used to semantically represent relevant network and MLN concepts. The ontology is utilised with SPARQL queries by the operator (via the faceted GUI) and by the MLN model.

Experiments of the platform in the Section 7 show that the MLN model works well in practice after it is trained to the current simulation context. The average number of RLFs reduced significantly during the weight learning phase. Also, experiments show the importance of monitoring the model; the combinatorial generation of formulae leads to a high amount of insignificant formulae. Removing these formulae with pruning makes both the MLN inference (the calculation of action probabilities) and the SPARQL query executions more robust. A potential future direction is investigating the scalability of the MLN model in larger networks and performing pruning of low-weight formulae in run time.

Some use cases of the GUI were shown to clarify how the MLN reasoning and network status can be monitored in view of this platform. The semantic representation of the underlying data and SPARQL queries provide a versatile access to the data and enable flexible information exploration with faceted browsing activities. Moreover, the SPARQL interface and the GUI provide an easy modification in order to investigate alternative reasoning results.

A further potential future research topic is the enhancement of the OWL-MLN interaction so that the MLN model settings can be dynamically modified by a human or by a description logic (DL) reasoner. Model settings include the selection of measurement variables, their threshold values, and formulae to be generated from the set of variables. Model settings could even include some initial formula weights with respect to prior knowledge. Moreover, the system will be enhanced by creating high-level goals which the user can use to modify the behaviour of the reasoning. For example, high-level goals could be mapped to corresponding MLN model settings.

The reason for combining semantic technologies with probabilistic reasoning was the decrease in the complexity in monitoring of the MLN model and to making the model dynamically modifiable. Our current implementation gives promising results to continue this work in order to enhance the system towards autonomic computing and towards adapting these technologies in more complex scenarios in the field of network management.

References

[1] Kasper Apajalahti, Eero Hyvönen, Juha Niiranen, and Vilho Räisänen. StaRe: Statistical reasoning tool for 5g network management. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenič, Sören Auer, and Christoph Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events*. Springer-Verlag, May 2016.

[2] Islam Beltagy, Katrin Erk, and Raymond Mooney. Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1210–1219, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[3] Leila Bennacer, Laurent Ciavaglia, Abdelghani Chibani, Yacine Amirat, and Abdelhamid Mellouk. Optimization of fault diagnosis based on the combination of Bayesian Networks and Case-Based Reasoning. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 619–622. IEEE, 2012.

[4] Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195, 1975.

[5] Rommel Carvalho, Shou Matsumoto, Kathryn Laskey, Paulo Costa, Marcelo Ladeira, and Laécio Santos. Probabilistic Ontology and Knowledge Fusion for Procurement Fraud Detection in Brazil. In Fernando Bobillo, PauloC.G. Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web II*, volume 7123 of *Lecture Notes in Computer Science*, pages 19–40. Springer Berlin Heidelberg, 2013.

[6] Pedro Chahuara, François Portet, and Michel Vacher. *Making Context Aware Decision from Uncertain Information in a Smart Home: A Markov Logic Network Approach*, pages 78–93. Springer International Publishing, Cham, 2013.

[7] Fedor Chernogorov, Ilmari Repo, Vilho Räisänen, Timo Nihtilä, and Janne Kurjenniemi. Cognitive self-healing system for future mobile networks. In *IWCMC*, 2015.

[8] Gabriela Ciocarlie, Christopher Connolly, Chih-Chieh Cheng, Ulf Lindqvist, Szabolcs Nováczki, Henning Sanneck, and Muhammad Naseer-ul Islam. Anomaly Detection and Diagnosis for Automatic Radio Network Verification. In Ramón Agüero, Thomas Zinner, Rossitza Goleva, Andreas Timm-Giel, and Phuoc Tran-Gia, editors, *Mobile Networks and Management*, volume 141 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 163–176. Springer International Publishing, 2015.

[9] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper. Technical report, Cisco, 02 2015. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[10] Ann Devitt, Boris Danev, and Katarina Matusikova. Ontology-driven automatic construction of bayesian networks for telecommunication network management. In *2nd Int. Workshop: Formal Ontologies Meet Industry (FOMI 2006)*, 2006.

[11] George Dimitrakopoulos, Kostas Tsagkaris, K Demestichas, E Adamopoulou, and Panagiotis Demestichas. A management scheme for distributed cross-layer reconfigurations in the context of cognitive {B3G} infrastructures. *Computer Communications*, 30(18):3807 – 3822, 2007. Optical Networking: Systems and Protocols.

[12] Lucas Drumond and Rosario Girardi. Extracting Ontology Concept Hierarchies from Text Using Markov Logic. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1354–1358, New York, NY, USA, 2010. ACM.

[13] European Telecommunications Standards Institute (ETSI). LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, 8 2016. (3GPP TS 36.213 version 13.2.0 Release 13).

[14] Carolina Fortuna and Mihael Mohorcic. Trends in the development of communication networks: Cognitive networks. *Computer Networks*, 53(9):1354 – 1376, 2009.

[15] Liljana Gavrilovska, Vladimir Atanasovski, Irene Macaluso, and Luiz A DaSilva. Learning and reasoning in cognitive radio networks. *IEEE Communications Surveys Tutorials*, 15(4):1761–1777, Fourth 2013.

[16] Rim Helaoui, Mathias Niepert, and Heiner Stuckenschmidt. Recognizing interleaved and concurrent activities: A statistical-relational approach. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 1–9, March 2011.

[17] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.

[18] Carole Hounkonnou and Eric Fabre. Empowering Self-diagnosis with Self-modeling. In *Proceedings of the 8th International Conference on Network and Service Management*, pages 364–370. International Federation for Information Processing, 2012.

[19] Seppo Hämäläinen, Henning Sanneck, and Cinzia Sartori. *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley Online Library, 1st edition, 2012.

[20] Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages

1–4, 2012.

[21] Stanley Kok, Parag Singla, Matthew Richarson, and Pedro Domingos. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2007.

[22] Sihyung Lee, Kyriaki Levanti, and Hyong S. Kim. Network monitoring: Present and future. *Computer Networks*, 65:84 – 98, 2014.

[23] Daniel Lowd and Pedro Domingos. Efficient Weight Learning for Markov Logic Networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 200–211, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] NGMN Alliance. 5G White Paper. Technical report, Feb 2015. https://www.ngmn.org/5g-white-paper.

[25] Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 1413–1418. AAAI Press, 2010.

[26] Mathias Niepert, Jan Noessner, and Heiner Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158, 2011.

[27] David Poole, Clinton Smyth, and Rita Sharma. Semantic Science: Ontologies, Data and Probabilistic Theories. In PauloCesarG. da Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, pages 26–40. Springer Berlin Heidelberg, 2008.

[28] Hoifung Poon and Pedro Domingos. Unsupervised Ontology Induction from Text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[29] Jay Pujara, Hui Miao, Lise Getoor, and William W Cohen. Using semantics and statistics to turn data into knowledge. *AI Magazine*, 36(1):65–74, 2015.

[30] Vilho Räisänen. Information and knowledge in telecommunications: a service-dominant view. In *SRII Global Conference (SRII), 2012 Annual*, pages 211–218. IEEE, 2012.

[31] Daniele Riboni and Claudio Bettini. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[32] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, February 2006.

[33] Péter Szilágyi and Szabolcs Nováczki. An automatic detection and diagnosis framework for mobile communication systems. *An automatic detection and diagnosis framework for mobile communication systems*, 9:184 ff., 2012.

[34] Hai-Tao Zheng, Bo-Yeong Kang, and Hong-Gee Kim. An Ontology-Based Bayesian Network Approach for Representing Uncertainty in Clinical Practice Guidelines. In PauloCesarG. da Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, pages 161–173. Springer Berlin Heidelberg, 2008.