On Introducing Knowledge Discovery Capabilities in Cloud-Enabled Small Cells

Jordi Pérez-Romero[1], Juan Sánchez-González[1], Oriol Sallent[1], Alan Whitehead[2]

[1] Universitat Politècnica de Catalunya (UPC), c/ Jordi Girona, 1-3, 08034 Barcelona, Spain
[2] ip.access, Cambridge, UK
jorperez@tsc.upc.edu

Abstract. The application of Artificial Intelligence (AI)-based knowledge discovery mechanisms for supporting the automation of wireless network operations is envisaged to fertilize in future Fifth Generation (5G) systems due to the stringent requirements of these systems and to the advent of big data analytics. This paper intends to elaborate on the demonstration of knowledge discovery capabilities in the context of the architecture proposed by the Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME) project that deals with multi-operator cloud-enabled small cells. Specifically, the paper presents the considered demonstration framework and particularizes it for supporting an energy saving functionality through the classification of cells depending on whether they can be switched off during certain times. The framework is illustrated with some results obtained from real small cell deployments.

Keywords: Knowledge discovery, Small Cells, Classification, Energy saving.

1 Introduction

As a next step in the evolution of cellular communication systems, industry and academia are focused on the development of the 5th Generation (5G) of mobile systems that targets a time horizon beyond 2020. 5G intends to provide solutions to the continuously increasing demand for mobile broadband services associated with the massive penetration of wireless equipment such as smartphones, tablets, the tremendous expected increase in the demand for wireless Machine To Machine communications and the proliferation of bandwidth-intensive applications including high definition video, 3D, virtual reality, etc. Requirements of future 5G system have been already identified and discussed at different fora [1][2].

It is expected that 5G networks will also be fueled by the advent of big data and big data analytics [3]. The volume, variety and velocity of big data are simply overwhelming. Nowadays, there are already tools and platforms readily available to efficiently handle this big amount of data and turn it into value by gaining insight and understanding data structures and relationships, extracting exploitable knowledge and deriving successful decision-making. While applications of big data and big data analytics are already present in different sectors (e.g. entertainment, financial services industry, automotive industry, logistics, etc.), it is envisaged that they will play a key role in 5G to extract the most possible value of the huge amount of available data generated by mobile networks and for efficiently delivering mobile services.

In this context, this paper supports the idea that Artificial Intelligence (AI) mechanisms, which intend to develop intelligent systems able to perceive and analyse the environment and take the appropriate actions, will fully fertilize in the 5G ecosystem. In [4] the authors presented a general framework for the application of AI-based knowledge discovery mechanisms relying on machine learning as a means to extract models that reflect the user and network behaviours. The paper identified different candidate tools and discussed the applicability in the development of Self-Organizing Network (SON) functionalities, also known as Self-X functionalities, for automating the operation of a cellular network [5]. In turn, a particularization of this general framework was presented in [6] focusing on extracting knowledge from the time domain traffic patterns of the different cells in a network. Two applicability use cases were elaborated, dealing with energy saving and spectrum management. Similarly, [7] focused on the identification of user mobility patterns in cellular networks by means of clustering techniques and on its applicability in the context of SON.

Relying on these prior works, this paper intends to further elaborate on the demonstration of the knowledge discovery capabilities in the context of the architecture proposed by the Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME) project [8] that deals with multi-operator cloud-enabled small cells. The proposed framework is particularized for supporting an energy saving Self-X functionality through the classification of cells depending on whether they can be switched off during certain times. To illustrate the operation of the process, the paper presents some results obtained from real small cell deployments.

The rest of the paper is organized as follows. Section 2 summarizes the architecture of the SESAME project, while Section 3 presents the considered demonstration framework for introducing knowledge discovery capabilities in this architecture. Then, Section 4 particularizes the framework for the energy saving use case and discusses the implementation of the building blocks for classifying the different cells. This is followed by Section 5, which provides some illustrative results of the proposed framework. Finally, conclusions are summarized in Section 6.

2 SESAME architecture

The SESAME project [8] focuses on the provision of Small Cell as a Service (SCaaS) under multi-tenancy, exploiting the benefits of Network Function Virtualisation (NFV) and Mobile Edge Computing (MEC). For that purpose, it proposes the Cloud-Enabled Small Cell (CESC) concept, a new multi-operator enabled Small Cell (SC) that integrates a virtualized execution platform for executing novel applications and services inside the access network infrastructure. In general terms, SESAME scenarios assume a certain venue (e.g. a mall, a stadium, an enterprise, etc.) where a Small Cell Network Operator (SCNO) is the SCaaS provider that has deployed a number of

CESCs that provide wireless access to end users of different Virtual Small Cell Network Operators (VSCNOs), according to specific Service Level Agreements (SLAs).

The SESAME architecture is presented in Fig. 1 [9]. The CESC consists of a Small Cell Physical Network Function (SC PNF) unit, where a subset of the SC functionality is implemented via tightly coupled software and hardware, and a micro server that supports the execution of Virtualised Network Functions (VNFs), which provide the rest of the SC functionality together with other added-value services. The CESCs support the Multi-Operator Core Network (MOCN) sharing model of 3GPP [10], which allows them to offer access over shared radio channels to multiple operators' core networks. Accordingly, each CESC is connected with the Evolved Packet Core (EPC) of each VSCNO through an S1 interface.

The physical aggregation of a set of CESCs, denoted as a CESC cluster, gives the possibility to jointly operate the computational, storage and networking resources of the micro servers as a single virtualised execution infrastructure, denoted as Light Data Centre (Light DC).

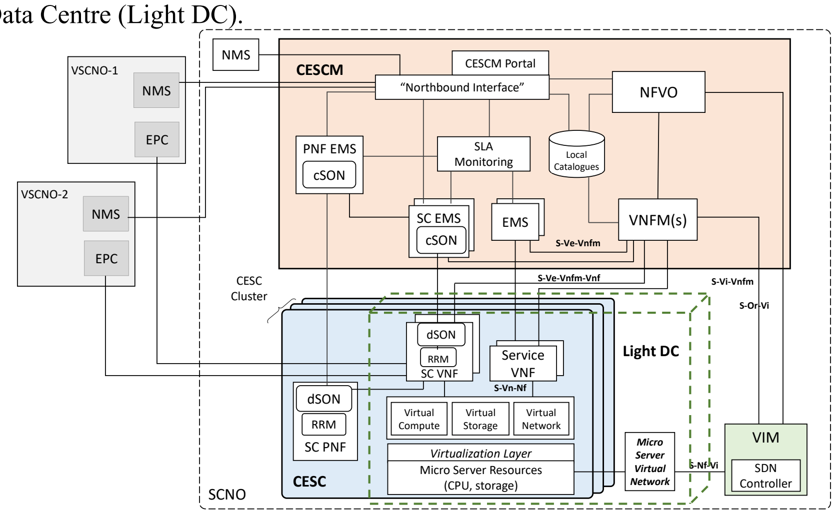

Fig. 1. SESAME architecture

The CESC Manager (CESCM) is the central service management component in the architecture that integrates the traditional 3GPP network management elements and the novel functional blocks of the NFV-MANO (Network Function Virtualization Management and Orchestration) framework. Configuration, Fault and Performance management of the SC PNFs and VNFs is performed through the Element Management System (EMS). In turn, the lifecycle management of the VNFs is carried out by the VNF Manager (VNFM), while the Network Functions Virtualization Orchestrator (NFVO) composes service chains constituted by one or more VNFs running in one or several CESCs and manages the deployment of VNFs over the Light DC with the support of the Virtualized Infrastructure Manager (VIM).

The CESCM is connected to the Network Management System (NMS) of the SCNO and the VSCNOs. Besides, it includes a portal that constitutes the main graphical frontend to access the SESAME platform for both SCNO and VSCNOs.

The SESAME architecture supports Self-X functions to tune global operational settings of the SC (e.g., transmit power, channel bandwidth, electrical antenna tilt) as well as specific parameters corresponding to Radio Resource Management (RRM) functions (e.g., admission control threshold, handover offsets, packet scheduling weights, etc.). Self-X functions can be centralised (cSON) at the EMS, distributed (dSON) at the CESCs or hybrid if they include both centralised and distributed components.

3 Implementing Knowledge Discovery capabilities as part of the SESAME demonstration framework

The introduction of knowledge discovery capabilities in a wireless network provides the ability to smartly process input data from the environment and come up with knowledge that can be formalized in terms of models and/or structured metrics that represent the network behaviour. This allows gaining in-depth and detailed knowledge about the network, understanding hidden patterns, data structures and relationships, and using them for a making smart network planning and optimisation decisions. In this way, the extracted knowledge models can be used to drive the decision-making of the actions associated to different Self-X functionalities.

Knowledge discovery is supported by machine learning tools to perform the mining of the data. Extracted knowledge models can be defined at different levels: cell level (contains the characterisation of the conditions on a per cell basis), cell cluster level (characterisation of groups of cells built according to their similarities) and user level (contains the characterisation of the conditions experienced by individual users).

Based on the above, Fig. 2 presents the considered framework for demonstrating the introduction of knowledge discovery capabilities in the context of SESAME. It is associated to the EMS, which encompasses both the PNF EMS and the SC EMS modules of the architecture shown in Fig. 1. The different elements of the considered framework are discussed in the following.

3.1 Network Orchestration System (NOS)

The SESAME EMS is based on the EMS of small cell vendor ip.access. The ip.access Network Orchestration System (NOS) provides configuration, fault and performance management features for small cells and related network elements. In SESAME, these elements include the SESAME CESC and the VNFs hosted by the CESC plus a collection of virtual network operators (VNOs), their individual virtual cells and associated SLA data.

The configuration and fault management aspects of the NOS are based on the ITU X.730 series of recommendations [11]-[14]. It represents these elements and their functions as managed objects [11]. Elements and optional functions are provisioned by creating managed objects and defining the values of their configurable attributes [12]. Once in service, the element or function represented by a managed object is able to report its state to the NOS [13]. Similarly, when a network element or function encounters a fault condition an alarm is raised on the managed object that represents it in a manner consistent with [14]. The procedural aspects of performance management reports are based on the concepts set out in [15] and the file format used conforms to [16].

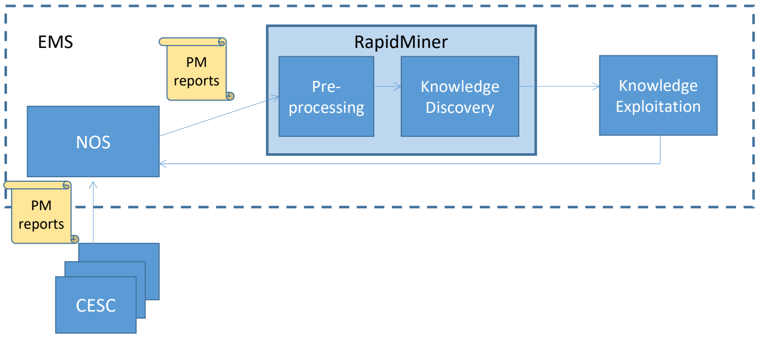

Fig. 2. Demonstration of knowledge discovery in SESAME

Managed objects are organised in a tree structure with the relationship between objects being captured by containment. In the SESAME context, there are two sub-trees of managed objects that have special significance.

- The CESC managed object is a collection object beneath which CESC objects are provisioned. Each CESC has a similar sub-tree that comprises (see Fig. 3): (i) Exactly one SC-PNF object. The PNF represents the configuration of the physical cell that is associated in a one-to-one relationship with the CESC. In SESAME, the PNF function is provided by the ip.access E40 LTE AP. (ii) Exactly one Small Cell Common VNF (SC-C-VNF) object. The function represented by this managed object interfaces with the physical cell to split the control plane into separate, per virtual network, slices. (iii) From one to six SC-VNF objects. Each such managed represents the control plane processing associated with a single virtual network slice and maintains a dedicated S1 connection into the associated EPC. As each of the above are discrete managed entities, there is also an associated Connection object for each function that represents the management link between the PNF or VNF and the NOS.
- The VSCNO managed object is a collection object beneath which VSCNO objects are provisioned. Each VSCNO object represents the data corresponding to a specific virtual network operator and is a sub-tree comprising (see Fig. 3): (i) A single VSCNO object. Each such object captures the key properties of a specific virtual network operator such as their name, Public Land Mobile Network IDentifier (PLMN ID) and access credentials for the NOS. This object also acts as a container for the child objects described below that provide details of the operator's virtual cells and SLAs. (ii) A single Virtual Cells (vCells) object. This is a collection object beneath which Virtual Cell objects are created in order to provision a new virtual cell. (iii) Zero or more Virtual Cell objects. Each such object represents the parameters a single virtual cell and contains a link to the CESC that hosts it. (iv) A

single Mobility Management Entity (MME) Pools. (v) A single SLA (vSlas) object beneath which SLA objects are created. (vi) At least one Provisioned SLA object. Each such object represents the details of a “network slice” that is applied when provisioning a new virtual cell. The network slice defines parameters such as the maximum number of UEs that are supported by the virtual cells and the maximum uplink and downlink bandwidth available to these UEs. (vii) Zero or more Monitored SLA objects. Each such object represents a set of criteria that are used to assess the performance of a set of virtual cells and the action to take when any of these criteria are not met.

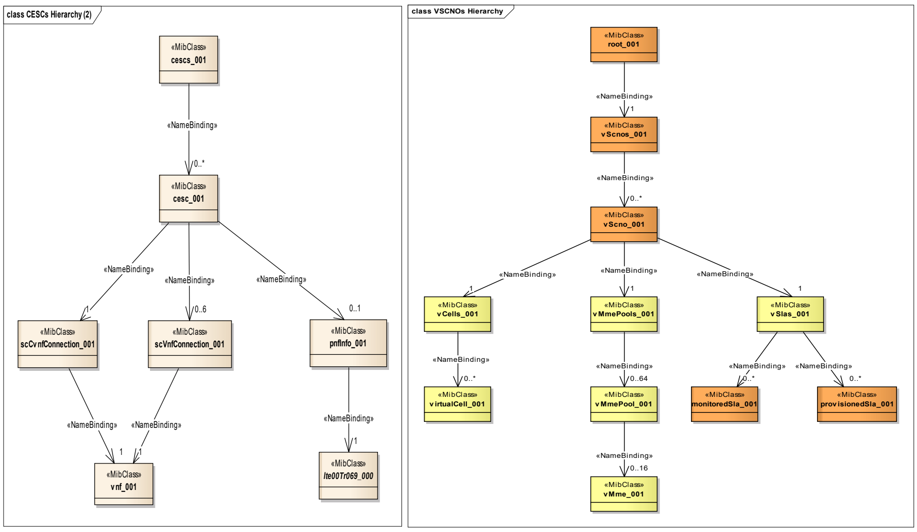

Fig. 3. Managed object sub-trees for the CESC (left) and the VSCNO (right).

3.2 Performance Management (PM) reports

Performance Management reports are XML files conforming the format described in [16]. They are produced according to a configured Reporting Interval and each file contains one or more Granularity Periods. The Granularity Period defines the time frame across which measurements are collected and aggregated and is typically defined to be in the range of 5 minutes to 24 hours. The granularity period of all the sample data in this study was set to one hour. The Reporting Interval is typically a multiple of the Granularity period. In the sample data used for this study, the Reporting Interval was set to either one hour or 24 hours.

Within each file, performance measurements are organised into groups of related items known as packages. Each file may contain up to 128 performance counters organised in to 27 packages. Some example packages are: (i) Access Control and Admission Control Packages that record the number of attempted and failed attempt to access the cell and establish radio bearers. (ii) Hand-in and Hand-out packages that record the number of attempted and successful handover procedures. (iii) A GTP-U

Usage Package that records the number of uplink and downlink GTP-U packets sent and received, the number of packets lost plus the total number of octets sent and received. (iv) A User Plane Package that records the number of call attempts by Radio Access Bearer (RAB) type, the maximum and mean number of simultaneous calls, uplink and downlink bandwidth utilisation by RAB type.

3.3 Pre-processing, knowledge discovery and knowledge exploitation

The pre-processing stage takes as input the PM files generated by the NOS and extracts the relevant metrics to be used by the knowledge discovery depending on the use case in hand. For that purpose, this stage can combine multiple PM files associated to different cells and/or time periods. Then, the knowledge discovery stage includes the machine learning algorithms to carry out the mining of the input data and extract the knowledge models.

Both the pre-processing and the knowledge discovery stages are implemented by means of the RapidMiner Studio Basic tool [17]. It is a powerful visual design environment for rapidly building complete predictive analytic workflows and incorporates multiple pre-defined data preparation and machine learning algorithms.

Finally, the knowledge exploitation stage applies the obtained knowledge models to drive the decision-making associated to different Self-X functionalities. As shown in Fig. 2 this stage can interact with the NOS to configure specific SC parameters.

4 Use case: Energy saving

The considered use case to illustrate the operation of the proposed framework is the energy saving Self-X functionality, which intends to reduce the overall energy consumption associated to the small cells deployed by the SCNO. In this case, the energy reduction is achieved by switching off the cells that carry very little traffic at certain periods of the day (e.g. at night) and making the necessary adjustments in the neighbour cells so that the existing traffic can be served through another cell. In this context, the knowledge discovery framework applies a classification methodology for identifying candidate cells to be switched-off based on their time domain traffic patterns. The automation of this procedure based on expert criteria captured in a training set becomes particularly useful considering that networks in the envisaged ultra-dense scenarios for future 5G systems can comprise several tens of thousands of cells. Therefore, it is not practical that a human expert can make this classification manually.

Based on the above, the classification categorizes the cells in the following classes:

- Class A: Candidate cell to be switched off
- Class B: Cell that cannot be switched off.

It is worth mentioning that the final decision on whether or not to switch off a cell will make use of this classification as well as other possible inputs which are out of the scope of this paper (e.g. the neighbour cell lists to ensure that traffic generated in a cell that has been switched-off can be served through another cell).

The following sub-sections illustrate the different steps of the proposed classification approach, whose mathematical details are presented in [6].

4.1 Pre-processing stage

The PM files generated by the NOS system include a number of XML files corresponding to different cells and periods of time. Each XML file includes metrics associated to different time instants. In turn, the considered classification process is based on the time domain traffic pattern of the different cells. For that purpose, the pre-processing stage is responsible for extracting the relevant metric to be used in the classification and presenting them in a format that is understandable by the classifier. Specifically, the selected metric considered in this work is the number of RAB admissions that have been accepted in a cell. Then, the pre-processing stage builds, for each cell, a time series $\mathbf{X_i}$=(x_i(t),, x_i(t-1), ..., x_i(t-(M-1))) composed of M samples of the number of RAB admissions in the cell i at different times t with a certain granularity.

Fig. 4 illustrates the building blocks of the pre-processing stage implementation using RapidMiner. The first block (Loop XML Files) reads each of the input XML files, while the subsequent blocks perform different operations to merge multiple XML files, to select from each one the hourly samples of the number of RAB admissions, and to build the table with the pre-processed data.

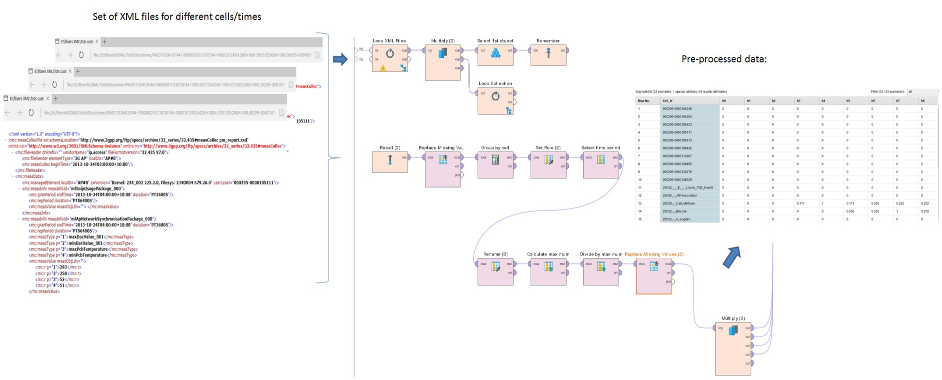

Fig. 4. Pre-processing stage

4.2 Classification stage

The classification stage performs the association between the input time series $\mathbf{X_i}$ of the i-th cell and the class C($\mathbf{X_i}$)∈{A,B} of the cell. The internal structure of the classifier is given by the specific classification tool being used and its settings are automatically configured through a supervised learning process executed during an initial training stage. This training uses as input a training set composed by S time series $\mathbf{X_j}$ j=1,...,S of some cells whose associated classes C($\mathbf{X}_j$) are pre-defined by an expert. The supervised learning process will analyse this training set to determine the appropriate configuration of the classification tool. In this way, the resulting classifier after the training stage can be used for classifying other cells whose class is unknown.

Fig. 5 illustrates the RapidMiner modules implemented for the performing the classification stage. The first module is the training stage that reads the cells from the training set and injects them to each classification in order to build the classification model (i.e. this is done in the first module shown inside a classification algorithm). Then, the last module of each classification algorithm takes as input the small cells to be classified from the pre-processing stage (output of Fig. 4) and applies the obtained classification model.

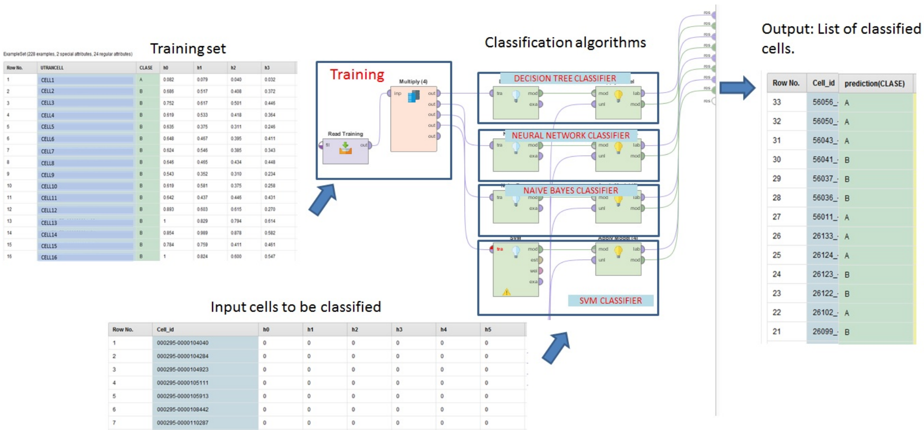

Fig. 5. Classification stage

As shown in Fig. 5, the following classifiers are implemented [18]:

- Decision tree induction: The classification is done by means of a decision tree, which is a flow-chart structure where each node denotes a test on a feature value, i.e. a component of vector $\mathbf{X}_i$, each branch represents an outcome of the test, and tree leaves represent the classes. The tree structure is built during the supervised learning stage through a top-down recursive divide-and-conquer manner.
- Naive Bayes classifier: The classifier evaluates the probability Prob($C(\mathbf{X}_i)| \mathbf{X}_i$) that a given cell $\mathbf{X}_i$ belongs to a class $C(\mathbf{X}_i)$ based on the values of the components of $\mathbf{X}_i$. The resulting class is the one with the highest probability. The computation of this probability is done using Bayes' theorem under the assumption of class conditional independence. The different terms in the computation of the Bayes' theorem are obtained from the analysis of the training set.
- Support Vector Machine (SVM): A SVM is a classification algorithm based on obtaining, during the training stage, the optimal boundary that separates the vectors $\mathbf{X}_j$ of the training set in their corresponding classes $C(\mathbf{X}_j)$. This boundary is used to perform the classification of any other input vector $\mathbf{X}_i$. The optimal boundary is found by means of a nonlinear mapping to transform the original training data into a higher dimension so that the optimal boundary becomes a hyperplane.
- Neural Network: The classification is done by means of a feed-forward neural network that consists of an input layer, one or more hidden layers and an output

layer. Each layer is made up of processing units called neurons. The inputs to the classifier, i.e. each of the components of vector $\mathbf{X_i}$, are fed simultaneously into the neurons making up the input layer. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer. The process is repeated until reaching the output layer, whose neurons provide the selected class C($\mathbf{X_i}$). The weights of the connections between neurons are learnt during the training phase using a back propagation algorithm.

5 Results

5.1 Scenario description

The considered scenario considers three different small cell deployments. The PM files of the first deployment include 9 different small cells belonging to an operator providing service on an island in the Pacific Ocean. The cells were deployed mainly in office blocks, hotels and the residences of VIPs. Whilst hand-in and hand-out to the macro network was possible, the small cells did not perform handovers to other small cells. The second deployment includes one small cell belonging to a national operator in a central European country. It is deployed as stand-alone cell in a shop belonging to the operator and, typically, did not perform hand-overs to any other cells. The third deployment includes 23 small cells. They belong to an operator providing service on an island in Northern Europe and were used to provide service mainly to users in their homes, in public houses and restaurants. Whilst hand-in and hand-out to the macro network was possible, the small cells did not perform handovers to other small cells.

5.2 Classification results

The available PM files for the considered small cells include the metrics for a total of one day. Then, the pre-processing stage shown in Fig. 4 builds, for each cell, a time series $\mathbf{X_i}$ composed of M=24 samples with the hourly values of the traffic in the cell. In turn, the classification stage of Fig. 5 applies the four considered classifiers. As for the training set, it consists of a total of S =228 cells from the deployment of [6].

As a first result, Fig. 6 shows the time domain pattern of two small cells classified as A and B by the decision-tree classifier. This appears as an adequate decision because the cell classified as A exhibits relatively long periods at night serving no traffic at all while the cell classified as B exhibits traffic during most of the time.

Table 1 summarizes the results obtained with the considered classifiers. In addition, the table also includes as a reference the “Expert classification”, which indicates the result of the classification process if it was made by the expert. Results show that, in general the number of small cells classified as A or B is very similar for the decision-tree, Bayes and SVM classifiers, while there are some more discrepancies for the Neural Network classifier. To further analyse this result, Table 2 assesses the different classification tools by presenting the percentage of coincidences between every pair of tools. For example, the table shows that 90.91% of the cells (i.e. 30 out of 33 cells) have been classified equally by the SVM and the Neural Network. The table also presents the percentage of coincidences with respect to the classification made by the expert. It can be observed that the largest percentages of coincidences are obtained with SVM and the decision-tree.

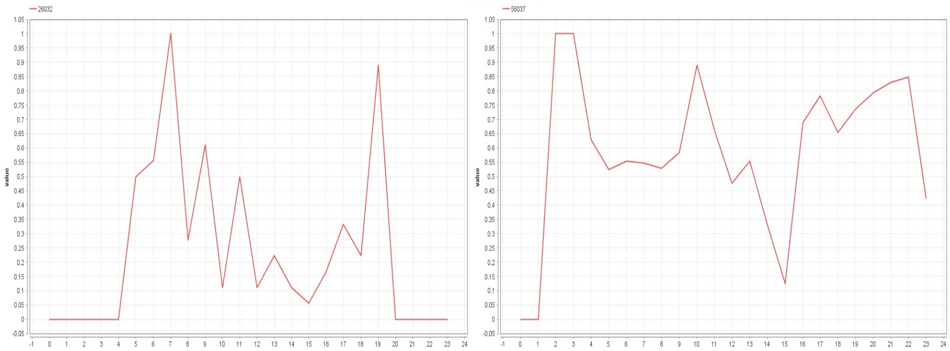

Fig. 6. Example of the time domain pattern of a cell classified as A (Left) and B (Right)

Table 1. Classification results.

Algorithm	Number of cells classified as A	Number of cells classified as B
Decision-tree	26	7
Bayes classifier	27	6
SVM	26	7
Neural network	23	10
Expert classification	27	6

Table 2. Percentage of total coincidences by every pair of classification tools.

Algorithm	SVM	Neural Network	Bayes classifier	Decision-tree	Expert
SVM	--	90.91%	96.97%	93.94%	96.97%
Neural Network	90.91%	--	87.88%	84.85%	87.88%
Bayes classifier	96.97%	87.88%	--	90.91%	93.94%
Decision-tree	93.94%	84.85%	90.91%	--	96.97%

6 Conclusions

This paper has presented a framework for demonstrating the use of knowledge discovery capabilities in the context of the architecture of the SESAME project. The proposed approach is based on pre-processing the PM files generated by a Network Orchestration System to extract the relevant metrics that will be used by the knowledge discovery to obtain the adequate knowledge models making use of machine learning tools. The framework has been particularized for supporting an energy saving Self-X functionality through the classification of cells depending on whether they can be switched off during certain times. To illustrate the operation of the process, the paper has presented some results obtained from real small cell deployments, comparing the behaviour of different classification algorithms.

Acknowledgements. This work has been supported by the EU funded H2020 5G-PPP project SESAME under the grant agreement 671596 and by the Spanish Research Council and FEDER funds under RAMSES grant (ref. TEC2013-41698-R).

References

1. M. Fallgren, B. Timus, (editors), "Scenarios, requirements and KPIs for 5G mobile and wireless system", Deliverable D1.1. of the METIS project, May, 2013.
2. R. El Hattachi, J. Erfanian (editors) "NGMN 5G White Paper", NGMN Alliance, February, 2015
3. Ericsson, "Big Data Analytics", White paper, August, 2013.
4. J. Pérez-Romero, O. Sallent, R. Ferrús, R. Agustí, "Knowledge-based 5G Radio Access Network Planning and Optimization", In: 13th International Symposium on Wireless Communication Systems (ISWCS-2016), Poznan, Poland, September, 2016.
5. J. Ramiro, K. Hamied, Self-Organizing Networks. Self-planning, self-optimization and self-healing for GSM, UMTS and LTE, John Wiley & Sons, 2012.
6. J. Pérez-Romero, J. Sánchez-González, O. Sallent, R. Agustí, "On Learning and Exploiting Time Domain Traffic Patterns in Cellular Radio Access Networks", In: 12th Int. Conf. on Machine Learning and Data Mining (MLDM), pp. 501-515, Springer, New York (2016).
7. J. Sánchez-González, J. Pérez-Romero, R. Agustí, O. Sallent, "On Learning Mobility Patterns in Cellular Networks", 1st Workshop on 5G – Putting Intelligence to the Network Edge (5G-PINE 2016), Thessaloniki, Greece, September, 2016.
8. Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME), http://www.sesame-h2020-5g-ppp.eu/
9. I. Giannoulakis (editor), "SESAME Final Architecture and PoC Assessment KPIs", Deliverable D2.5 of SESAME, December, 2016.
10. 3GPP TS 23.251 v13.1.0, "Network Sharing; Architecture and functional description (Release 13)", March, 2015.
11. ITU-T X.730 "Open Systems Interconnection – Systems Management: Object management function", January, 1992.
12. ITU-T X.732 "Open Systems Interconnection – Systems Management: Attributes for representing relationships", January, 1992.
13. ITU-T X.731 "Open Systems Interconnection – Systems Management: State Management function", January, 1992.
14. ITU-T X.733 "Open Systems Interconnection – Systems Management: Alarm reporting function", January, 1992.
15. 3GPP TS 32.401 v14.0.0, "Performance Management (PM). Concept and requirements (Release 14)", March, 2017.
16. 3GPP TS 32.435 v14.0.0, "Performance measurement; eXtensible Markup Language (XML) file format definition (Release 14)", April, 2017.
17. RapidMiner Studio, http://www.rapidminer.com
18. R.A.Wilson, F.C.Keil, The MIT Encyclopedia of the Cognitive Sciences, MIT Press, 1999.