entropy

MDPI

Article

An Approach to Data Analysis in 5G Networks

Lorena Isabel Barona López [†], Jorge Maestre Vidal [†] and Luis Javier García Villalba [*,†]

Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, Madrid 28040, Spain; lorebaro@ucm.es (L.I.B.L.); jmaestre@ucm.es (J.M.V.)

[*] Correspondence: javiergv@fdi.ucm.es; Tel.:+34-91-394-7638

[†] These authors contributed equally to this work.

Academic Editor: Kevin H. Knuth

Received: 16 January 2017; Accepted: 14 February 2017; Published: 16 February 2017

Abstract: 5G networks expect to provide significant advances in network management compared to traditional mobile infrastructures by leveraging intelligence capabilities such as data analysis, prediction, pattern recognition and artificial intelligence. The key idea behind these actions is to facilitate the decision-making process in order to solve or mitigate common network problems in a dynamic and proactive way. In this context, this paper presents the design of Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET) Analyzer Module, which main objective is to identify suspicious or unexpected situations based on metrics provided by different network components and sensors. The SELFNET Analyzer Module provides a modular architecture driven by use cases where analytic functions can be easily extended. This paper also proposes the data specification to define the data inputs to be taking into account in diagnosis process. This data specification has been implemented with different use cases within SELFNET Project, proving its effectiveness.

Keywords: 5G; data analysis; network function virtualization; situational awareness; software defined networking

1. Introduction

5G networks expect to provide a secure, reliable and high-performance environment with minimal disruptions in the provisioning of advanced network services, regardless the device location or when the service is required [1]. This new network generation will be able to deliver ultra-high capacity, low latency and better Quality of Service (QoS) compared with current Long Term Evolution (LTE) networks [2]. In order to provide these capabilities, 5G proposes the combination of advanced technologies such as Software Defined Networking (SDN) [3,4], Network Function Virtualization (NFV) [5,6], Cloud Computing [7], Self-organized Networks (SON) [8,9], Artificial Intelligence, Big Data [10–12], Device to Device Communications (D2D), among others [13–17]. In particular, 5G will be able to face unexpected changes or network problems through the identification of specific situations and taking into account the user needs and the Service level Agreements (SLAs).

Nowadays, the main telecommunication operators and community research are working in strategies to facilitate the decision-making process when specific events or situations compromises the health in 5G Networks [18,19]. Meanwhile, the concept of situational awareness (SA) and incident management models applied to 5G Networks are also an emerge topic [20,21]. In this context, Self-Organized Network Management in Virtualized and Software Defined Networks Project (SELFNET) [22] combines SDN, NFV and SON concepts to provide a smart autonomic management framework, analysing and resolving network problems and improving the QoS and Quality of

Experience (QoE) of end users. In order to facilitate the decision-making process, SELFNET proposes an analysis phase to diagnosis and predict possible problems in 5G Networks.

This paper presents the design of SELFNET Analyzer Module, which main objective is to diagnose the network state and infer data from monitored low level and aggregated metrics in order to facilitate proactive responses. The contributions of this proposal include: (i) the description of diagnosis and prediction capabilities in 5G environments and how it is being applied in current research and projects; (ii) the introduction of SELFNET Analyzer Architecture, its design principles and requirements and (iii) the definition of data specification, as well examples, to obtain the initial parameters to diagnostic purpose. This document is organized into eight sections, being the first of them the present introduction. Section 2 describes 5G requirements, related works, the main characteristics and analytic capabilities of SELFNET Project. Section 3 outlines the design principles, requirements and the architecture of Analyzer Module as a whole. Then, Section 4 shows this module as a black, emphasizing the data inputs and outputs. Section 5 formally defines how the data must be specified. Section 6 illustrates examples of the data specification and their workflows. Section 7 discusses the main contributions of this proposal. Finally, conclusion and future work are presented in Section 8.

2. Background

This section describes how analysis and intelligent capabilities can address 5G requirements, the related work and research projects, emphasizing the main features of SELFNET Project.

2.1. Diagnosis Capabilities in 5G Networks

A 5G network envisages an architecture able to cover three main domains [1]: (i) enhancement of radio capabilities to enable the spectrum optimization, the interference coordination and cost-effective dense deployments; (ii) provisioning of an effective network management environment to create and deploy a common core to support several use cases in a cost-effective manner; and (iii) simplification of the system operations by means of automated procedures, where the introduction of new capabilities or network functions should not imply increased complexity on operations and management tasks. In order to tackle these requirements, 5G networks take advantage of the separation between data and control plane (network programmability) offered by SDN architectures [23], the deployment of virtualized network functions, the scalability and flexibility in the service provisioning based on cloud environments, enabling high capacity and massive communications (cognitive radio, carrier aggregation, Machine to Machine Communication), spectrum and resource optimization (millimeter wave and massive Multiple Input Multiple Output (MIMO)) and intelligent capabilities provided by artificial intelligence or self-organization concepts [24,25].

In particular, the introduction of analysis and intelligent capabilities [8,19] could be applied to several domains such as autonomic network maintenance, automation in the provisioning of services, prediction and remediation of congestion or queue utilization, detection of security threats, improving network efficiency, multi cell coordination, provisioning of high QoS and QoE for services, etc. For this purpose, analysis and intelligent capabilities allow to response to network problems based on pattern recognition, the dynamic smart selection of the best location where the services can be deployed or migrated, sharing and releasing of resources based on forecasting methods, building of context awareness models based on real time information from the network, its devices and applications. In order to provide intelligence and facilitate the decision-making process, some tasks must be performed. On one hand, analysis stage is intended to perform the identification of network situations and events. These situations do not necessarily imply (a priori) a harmful nature. On the other hand, the decision-making task determines if a specific situation is a risk for the network health, or its components, and then it performs the respective countermeasures.

In this context, traditional approaches apply different analysis and reasoning techniques, such as Bayesian Networks (BN) [26], in order to provide intelligent to common network management tasks. However, these models are not sufficient to guarantee the network performance according to SLAs and future 5G user needs [1]. There are some proposals to address the data analysis in 5G systems and its elements such as access and radio components [13,27], network devices [28], cloud elements [29] and resource allocation [30]. In [31], a prototype to perform mobile network analysis based on Markov Logic Network (MLN) and semantic web technologies is presented. This approach allows the optimization and network status characterization but does not explain how it cover heterogeneous data sources. For its part, Imran et al. [10] proposes a framework to provide a full view of network status based on machine learning and big data concepts. To this end, their proposal predicts the user behaviour and dynamically associate the network response to the network parameters. However it is doesn't specify how to deal with SDN or NFV components.

Meanwhile, there are reports [1,32] and projects [21,33–37] that introduce analysis capacities to cover 5G requirements. In this way, METIS Project [28] takes into account SON concept in order to provide a new level of adaptability to 5G infrastructures. Meanwhile, 5G-NORMA [35] introduces adaptive capacities to allocate network functions based on user and traffic demands over time and location. CHARISMA project [36] deploys an intelligent cloud radio access network and end devices. For its part, 5G-Ensure [21] proposes a 5G secure system based on risk assessment and mitigation methodologies. COGNET [37] takes into account Machine learning, SDN and NFV to provide dynamic adaptation of network resources. For its part, a whole approach to address not only analysis component but also the whole cycle of incident management in 5G networks is proposed in [20]. This work applies the three stages of processing information of Endsley Model [38] to 5G Networks: perception, comprehension and projection. In the perception phase the monitoring and collection of different metrics from network infrastructure (and its elements) are performed. Then in comprehension stage, the association and correlation of this information are performed in order to provide enhanced metrics to be analysed (projection phase). The analysis component includes the diagnosis and prediction of the whole state of the system. In general terms, these proposals aid to tackle 5G Requirements but they do not offer a generalized approach able to take into account several kind of metrics from heterogeneous data sources, that is the case of SELFNET Project [22].

2.2. SELFNET Project

The SELFNET H2020 Project [22] aims to provide an autonomic network management framework for 5G mobile network infrastructures through the integration of novel technologies such as SDN, NFV, SON, Cloud computing and Artificial Intelligence. SELFNET enables both autonomic corrective and preventive actions to mitigate existing or potential network problems while providing scalability, extensibility and reduce capital expenditure (capex) and operational expenditure (opex). These capabilities are provided through a layered architecture and a use-case driven approach, as is detailed in [34]. The SELFNET architecture addresses major network management problems including self-protection capabilities against distributed cyber-attacks, self-healing capabilities against network failures, and self-optimization to dynamically improve the performance of the network and the QoE of the users. For this purpose, SELFNET defines two kind of advanced network functions: (i) sensors to monitor specific information from the network and (ii) actuators to address or mitigate possible problems. In particular, the network intelligence is provided by SON Autonomic Layer. This layer collects metrics related with the network behaviour and use that information to infer the network status. Then, it decides the actions to be executed to accomplish the system goals. The SON Autonomic Layer is composed by two sublayers: (i) Monitor and Analyzer Sublayer and (ii) Autonomic Management Sublayer. The Monitor and Analyzer Sublayer follows the Endsley Situational Awareness Principles. Monitoring and Discovery, Aggregation and Correlation and Analyzer modules corresponds with the Perception, Comprehension and Projection functions as is shown in Figure 1.

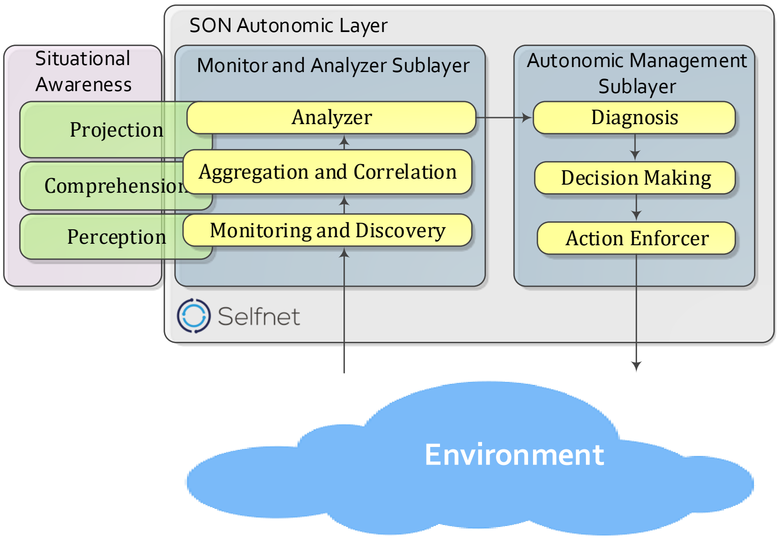

Figure 1. Endsley vs. SELFNET Autonomic Layer.

Regarding the Analyzer Module, its main goal is to infer data from the monitored metrics in order to facilitate proactive responses over the network infrastructure (i.e., enhance diagnosis and decision-making tasks). Therefore the Analyzer Module is the first step to provide intelligence to the system, where complex conclusions should be reached by reasoning about knowledge provided by the Monitoring/Aggregation stages and the definition of each use case. Because of this, the Analyzer Module distinguishes three great information processing activities: Pattern Recognition, Reasoning and Prediction. The achieved conclusions are described in the form of symptoms related with each use case. Bearing this in mind, it is possible to assert that the Analyzer Module provides a symptom-oriented Situational Awareness bounded by the situations defined for each use case.

3. SELFNET Analyzer Module Design

In this section the design of SELFNET Analyzer Module is detailed. It also describes the initial assumptions, the requirements, the design principles as well as the Analyzer architecture.

3.1. Initial Assumption and Requirements

The following describes the most relevant requirements and the main initial assumptions considered in the design of the Analyzer Module:

- *Scalability*. The approach must be allowed to add new capabilities (extensibility), according to SELFNET design principles. For this reason, the integration of additional analytic functionalities are done via plugins.
- *Use Case Driven*. Given the heavy reliance of the tasks performed with the characteristics of use cases, the basic definition of the observations to be studied (Knowledge-based objects, rules, prediction metrics, etc.) are provided by the use case operator, thus being the Analyzer Module scalable to alternative contexts.
- *Knowledge Acquisition*. It is well known that the most common disadvantage of the expert systems is the initial knowledge acquisition problem. Hence, to have skilled operators in novel use cases with the ability to properly specify rules is not always straightforward. This document does not address the issue of the innate knowledge acquisition. Our approach assumes that the use case knowledge-bases are provided by skilled operators or by accurate machine learning algorithms.
- *User-Friendly Symptom Definition Rules*. The definition of proper rule-sets is a tricky business. Thus, even the skilled operators often do coherence/ambiguity mistakes. In order to mitigate these

problems, the configuration and definition of new use cases should be user-friendly, as well as the scheme for building new rule-sets.
- *Uncertainly*. Classical logic permits only exact reasoning. It assumes that perfect knowledge always exists, but this remains far from the SELFNET reality. In order to improve the quality of the conclusions, the Analyzer Module manages the knowledge bearing in mind uncertainty. This is particularly appropriate for certain analytic features, such as studying observations based on decision thresholds or confidence intervals. In addition, closing the door on possible stochastic dependent definitions is against the SELFNET design principles, as these could be the keys to properly specify future use cases.
- *Filtering*. Initially, the filtering of symptom reports is not considered. Because of this, every inferred symptom, regardless of nature or uncertainty, is transmitted to the diagnosis/decision-making stages, where their impact and relevance are properly assessed.

3.2. Design Principles

The following design principles and limitations lay the foundation of the Analyzer framework, as well as the implementation of its internal components:

- *Big Data*. In order to deal with huge and homogeneous datasets, Big Data provides predictive algorithms, user behaviour analytics, and aggregation/correlation functionalities [39]. These capabilities are mainly taken into account in monitoring and aggregation tasks. The Analyzer Module deals with aggregated and correlated metrics, hence reducing the amount of information to be analysed. In our approach, the implementation of Big Data technologies to handle all this information is optional, leaving the decision of integrate these tools at the mercy of the SELFNET administrators, which driven by a better awareness of the use cases and the monitoring environment are more able to decide whether they are counterproductive or beneficial [40]. Because of this, our contribution is compatible with both Big Data and conventional techniques.
- *Stationary Monitoring Environment*. According to Holte et al. [41], in a stationary monitoring environment, the characteristics and distribution of the normal observations to be analysed match the reference sample population considered in the Analyzer learning processes. If the monitoring environment distribution is able to change representatively, it is considered non-stationary. Another problem that may reduce the quality of the analytics is the presence of gradual changes over time in the statistical characteristics of the class to which an observation belongs. In the literature this fluctuation is known as concept-drift. These problems are discussed at length in [42]. The assumption that the Analyzer Module operates on a stationary environment brings a simple and efficient solution, but prone to slight failures when the changes occur. On the other hand, to consider a non-stationary monitoring environment improves accuracy, but entails new challenges, among them: detection of changes, implementation of model/regression updating techniques, identifying when the calibration must be completed or selection of the samples that will be taken into account in new trainings. Given the complexity that this implies, the Analyzer Module assumes a stationary monitoring environment. The non-stationary approach will be part of future work.
- *High Dimensional Data*. The analysis of high dimensional data implies to bear in mind data whose dimension is larger than dimensions considered in classical multivariate analysis. As indicated by Bouveyron et al. [43], when conventional methods deal with high dimensional data they are susceptible to suffer the well-known curse of dimensionality, where considering a large number of irrelevant, redundant and noisy attributes leads to important prediction errors. Hence operate with this data implies the need for more specific and complex algorithms. In terms of SELFNET this means that the vector of Health of Network Metrics (HoN) is large enough to consider the implementation of specific methods adapted to optimize the processing tasks of this kind of information. A priori there are no signs of SELFNET requiring processing an important amount of High Dimensional Data. Therefore, this paper does not take into account differences between

conventional and High Dimensional data, assuming that the aggregation tasks will be able to optimize the amount of attributes to be analysed.

- *Supervision*. SELFNET training mode. The analytical methods based on modeling/regression assume that new knowledge can be inferred from observations, by a prior learning stage. The learning process often requires reference data which allows identifying the most characteristic features of the monitoring environment, such as rules, boundaries, incidence matrices, direction vectors or basic statistics. Given the complexity involved in designing a SELFNET training mode, this approach describes how the information needed for the construction of new models is obtained.
- *Centralized Design*. To assume a centralized approach lead us to pose a general purpose scheme where the onboarding of new use cases is completely configurable by specification, and which does not requires updating the implementation (see Figure 2). Therefore the centralized approach is not dependent on the characteristics of the use case, so it is highly scalable and allows performing tasks efficiently (avoiding redundancy). However, its design and the description of use cases is complex. On the other hand, the distributed approach includes an additional component for each use case in which specific pattern recognition and prediction methods are implemented via plugin. The preprocessing, selection and symptom discovery mechanisms have general purpose. In essence, this second approach is easy of design, but completely use case dependent; each time a new use case is onboarded, the Analyzer implementation must be updated. Due to the large impact on the scalability that this entails, the centralized approach is considered hereinafter.

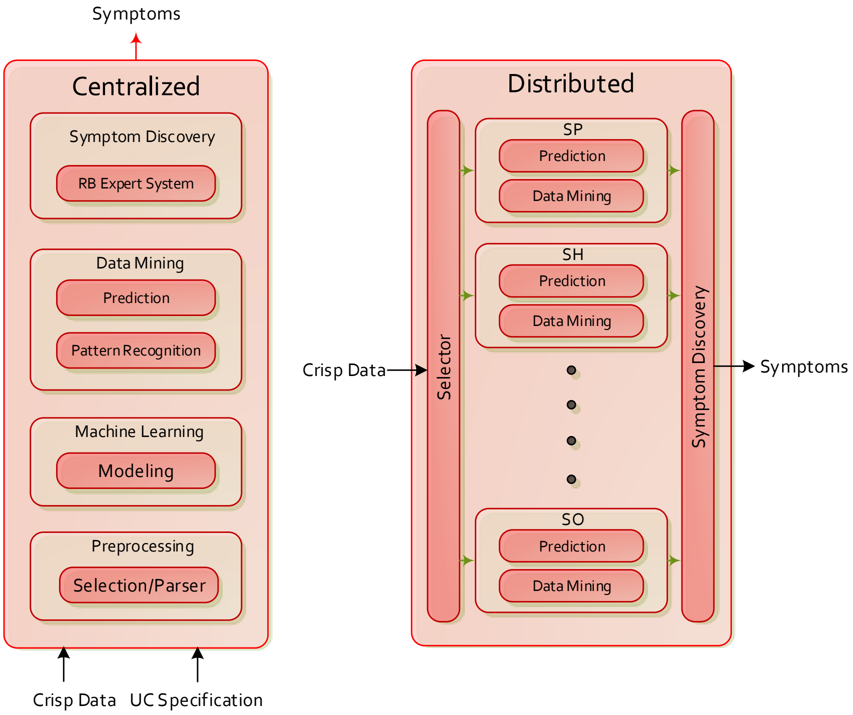

Figure 2. Centralized and Distributed Architectures.

- *Data Encapsulation*. The greatest challenge in designing the SELFNET Analyzer approach is the requirement of dealing with unknown data. It is possible to assume that use cases do not provide clear enough information about the characteristics of the information to be analyzed (in fact, several future use cases are completely unknown). At the specification stage, use cases operators tend to provide good qualitative information about the metrics to consider, but may overlook details about their quantitative nature: data type, domain, range, restrictions, etc., which is what in the first instance, will be considered in the analysis tasks. Furthermore, quantitative information is much use case dependent. In order to subtract relevance to quantitative details (which are the backbone of the aggregation/correlation tasks), and thus facilitate the incorporation of new use cases by definition of general purpose descriptors, the SELFNET Analyzer is driven by data encapsulated in two levels of abstraction: quantitative and qualitative parameters (see Figure 3). The first one is independent of the use cases, and allows designing a centralized analysis framework valid for any type of data specification. On the other hand the qualitative parameters gather information directly related with SELFNET and the use case to which they belong (metric name, source, location, tenant, etc.). This data is mainly required for aggregation/correlation, diagnosis and decision making.

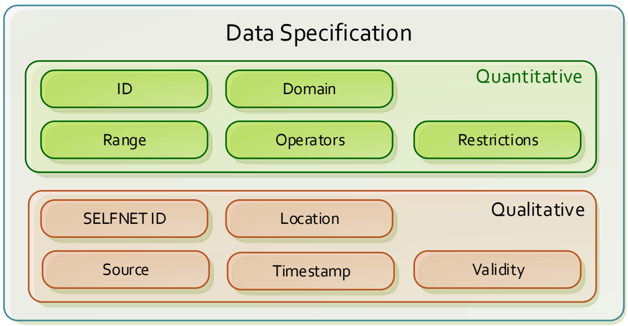

Figure 3. Example of Data Encapsulation.

3.3. Analyzer Module Architecture

In Figure 4 the architecture of the Analyzer Module is illustrated. It is centralized, and distinguishes the following eight core components: (1) Pattern Recognition, (2) Prediction, (3) Adaptive Thresholding, (4) Knowledge-base, (5) Inference Engine, (6) Memory, (7) User Interface and (8) Uncertainty Estimation. The sets (4)–(6),(8) is related with Reasoning, (1)–(3) with Projection and (7) with the administration of the use cases. Their tasks are summarized below:

- *Pattern Recognition*: identifies previously known or acquired patterns and regularities in facts related with aggregate data (i.e., $Fa(T_h), Fa(KPI), Fa(Ev)$), and returns Facts Fa with the results of their study. With this purpose, different internal tasks may be executed: study of the input data (both training data and samples to be analysed), decision of the best suited data mining strategies for each context, feature extraction, construction of models/regressions, analysis of facts related with aggregate data in order to find and labeling verification. Note that the bibliography collects a plethora of pattern recognition methods, which are adapted to the needs of the use cases and to the characteristics of the different monitoring environments [44]. The SELFNET Analyzer focuses on two fundamental actions: the identification of signatures of previously known events [45] and the detection of anomalies [46].
- *Prediction Component*: calculates the prediction metrics (as Facts Fa) associated to each use case from the observations provided by the aggregation stage (Thresholds T_H, Key Performance Indicators KPI and Events Ev). This implies different processing steps: management of a track record with the data required to build forecasting models, analysis of the data characteristics which are relevant for deciding the best suited prediction algorithms, construction of forecasting models, decision of prediction algorithms, forecasting and evaluation of the results in order to learn from the previous decisions. Note that as stated in [47], the prediction of network events enhances the optimization of resources, allows the deployment of proactive actions and anticipates risk identification. The SELFNET Analyzer focuses primarily on infer predictions from two data structures: time series and graphs. The first one aims to determine the evolution of the HoN metrics, hence it mainly implements exponential smoothing algorithms [48] and autoregressive models [49]. On the other hand, the evolution of graphs is predicted in order to anticipate the discovery of new elements [50] and facilitate the management of resources [51].

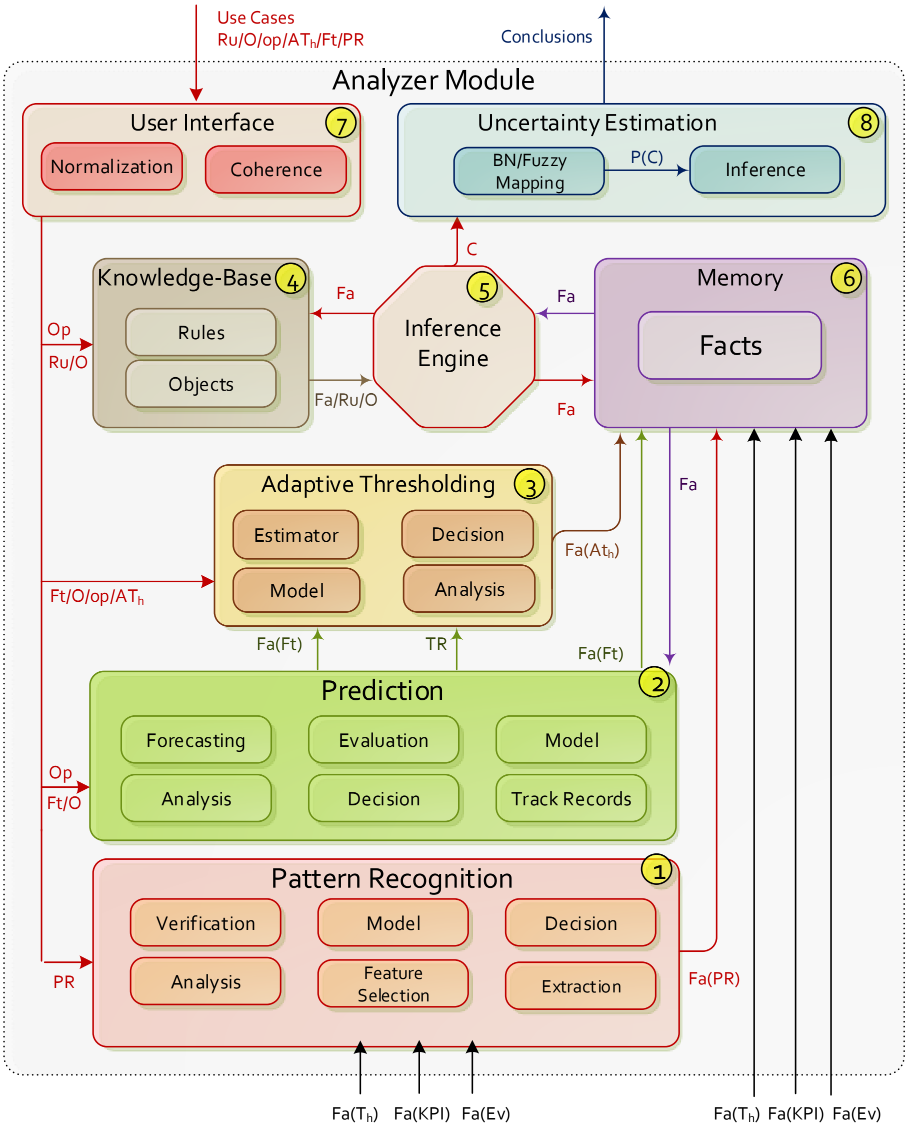

Figure 4. Analyzer Module architecture.

- *Adaptive Thresholding*: establishes measures to approximate when the forecast errors must be taken into account when identifying symptoms. Therefore it receives as input parameters the values related with the prediction metrics (Track Record TR and Forecasts Ft), and returns adaptive thresholds AT_h. Their construction involves different steps, such as analyzing and extracting the main features from the input data, decision of the best suited thresholding algorithms, modeling and estimation of thresholds. The SELFNET Analyzer build adaptive thresholds from data represented as time series or graph, which allows inferring more accurate conclusions from every forecast generated by the prediction component. The main applicability of the adaptive thresholds is considering the context of the monitoring environment in the inference of new facts related with filtering [52], and decreasing the false positives rates [53].
- *Knowledge-Base*: stores specific information about each use case. This data is represented by objects and rules. The objects O are the basic units of information (ex. temperature, congestion, latency, etc.). The rules Ru are the guidelines for reasoning that enable the inference of facts and conclusions. Facts, objects, and their values are interrelated through operations Op. A priori, in this approach online machine learning is not considered in order to acquire knowledge about the use

cases in real-time [54], such as definition of new rules, prioritization, metric weighting, etc. (i.e., all information to be considered part of the original training and the specification of the use cases and their symptoms provided by operators).

- *Inference Engine*: applies rules Ru to the knowledge base in order to deduce new knowledge. This process would iterate as each new fact Fa in the knowledge base could trigger additional rules. Traditionally, inference engines operate in one of two modes: forward chaining and backward chaining [55]. The first initially considers previously known facts and infers new facts. On the other hand, backward chaining initially considers facts and tries to infer the causes that have led to them. Because the SELFNET Analyzer infers conclusions from discovered facts, the first approach is implemented. In addition, it is important to bear in mind that the easier implementation of the inference engine considers basic implication elimination rules (i.e., modus ponens rules) driven by propositional logic [56]. They can be adapted to different representations of uncertainty, such as fuzzy logic [57], rough sets [58] or Bayesian networks [26]. But in order to facilitate the understanding of this proposal, the current specification of rules on the SELFNET Analyzer applies only basic propositional logic rules (as described in Section 5), hence postponing for future works a most complex but generic definition.
- *Memory*: stores all the known facts Fa concerning with the use cases (ex. Temperature = 3°, Latency >200 ms, etc.) considering those predicted/inferred $(Fa(PR), Fa(AT_h), Fa(Ft))$ and those provided by the SELFNET Monitoring/Aggregation stages $(Fa(T_h), Fa(KPI), Fa(Ev))$. Metadata related with qualitative additional information about the nature of the discovered facts is also stored.
- *User Interface*: configures Patter Recognition PR for each use case and allows updating the knowledge-base by inserting, modifying or deleting data associated with every use case, such as objects O, rules Ru operations Op or prediction metrics Ft. The information is preprocessed aiming to ensure compatibility and coherence [59]. The latter is particularly important, as it tries to avoid contradictions and ambiguity between rules, prior to their incorporation into the SELFNET intelligence.
- *Uncertainly Estimation*: complements the inference engine and facilitates the study of the conclusions bearing in mind their uncertainty. Its outputs are the acquired conclusions as potential symptoms of relevant incidences, their uncertainty and the information associated with their inference (facts, triggering rules, etc.). This is the only optional element of the architecture, since its use is only required when the SELFNET Diagnostic task [60] need to disambiguate conclusions, filter those of greater uncertainty or convert the logic on the Analyzer to data specified for upper layers of SELFNET. For example, when the inference engine operates on fuzzy logic rules, the element of Uncertainly Estimation generates a quantifiable result use-friendly for Diagnosis as crisp logic, given fuzzy sets and the corresponding membership degrees (i.e., defuzzification) [61].

4. Analyzer Inputs/Outputs

By studying the Analyzer Module as a black box model it is possible to focus more on its inputs/outputs and their relationship with the rest of the SELFNET components [34]. From this perspective, their information sources, nature of the data and behaviour in different circumstances are described. As shown in Figure 5, the Analyzer Module depends on three sources of information. Two of them are external: the SELFNET Aggregation component and the use case operators; the last is internal: data generated by the Analyzer Module itself. The inferred conclusions are reported to the SELFNET Diagnosis Module as symptoms [60]. The role played by each of these elements is detailed below:

- *Aggregation*. Observations in SELFNET come to the Analyzer through the Aggregation Layer (Perception capabilities within the Endsley's model). The information provided by this source contains facts concerning Events $Fa(Ev)$, Thresholds $Fa(T_H)$ and Key Performance Indicators $Fa(KPI)$ related to the current network status.

- *Use Case Operators*. The knowledge-base is specified from data acquired from the use case definitions. Because of this, use case operator may provide inference rules Ru (1), and declare the objects O (2), operations Op (3) and prediction metrics Ft (4) to be taken into account (i.e., what observations should be taken into account (2), how (3)what data must be forecasted (4) and how are they considered in order to acquire knowledge about the specific use case (1)). Optionally, the use case operator may describe the adaptive thresholds AT_h to be calculated, and if pattern recognition PR is required, then configuring how it must be addressed.
- *Analyzer*. An important part of the information necessary for proper Reasoning is generated by the Analyzer Module itself. It is gathered into a pair of groups: Perception and Machine Learning. The first block is imperative, and establishes facts Fa from pattern recognition $Fa(PR)$, forecasts $Fa(Ft)$ and adaptive thresholds $Fa(AT_h)$. On the other hand, machine learning may provide additional data to that provided by use case operators (definition of new rules Ru and description of prediction metrics Ft). Furthermore, it could generate information to improve the knowledge management (weight, prioritizations, fusion, smoothing, etc.).
- *Diagnosis*. the final conclusions and symptoms that compose the SELFNET Situational Awareness are sent to Intelligent Diagnostic Module (Autonomic Management Sublayer) [60], via reports Re.

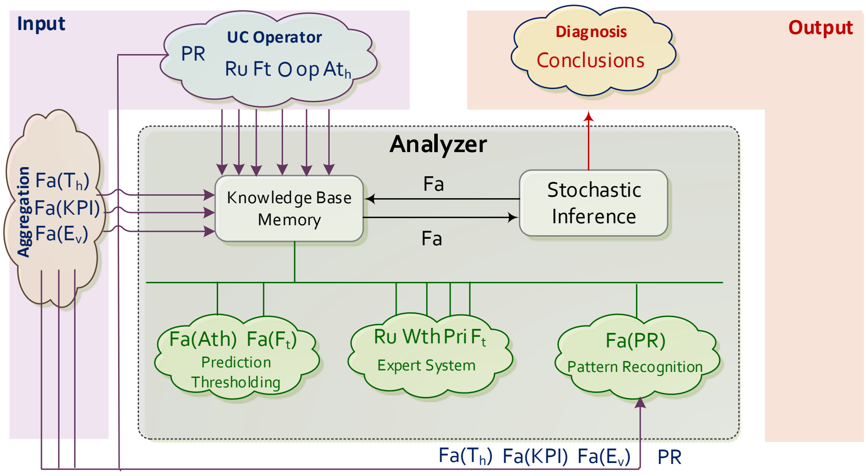

Figure 5. Analyzer Module as a Black Box.

5. Use Case Descriptors

This section describes the characteristics of Analyzer Module quantitative data and its categories. In Table 1 the quantitative data is summarized.

Table 1. Summary of UC data Specification.

Data	Category	Provider	Destination	Format
Object (simple) O	Specification	Use Case	Analyzer	$O_i : \{object\quad name \mid weight \mid noValues \mid range\quad of\quad values\quad Va\}$
Object (mult) O	Specification	Use Case	Analyzer	$O_i : \{object\quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\}$
Operation Op	Specification	Use Case	Analyzer	$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\}$
Facts Fa	Assessment	Agg-Ana	Analyzer	$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\}$
Rule Ru	Specification	Use Case	Analyzer	$Rule : \{rule \mid priority \mid use\quad case\}$
Forecast (ts) Ft	Specification	Use Case	Analyzer	$Ft_i : \{timeSeries \mid object \mid domain\quad lenght\}$
Forecast (G) Ft	Specification	Use Case	Analyzer	$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\}$
Threshold T_h	Specification	Use Case	Analyzer	$T_{hi} : T_h\quad name \mid object$
A. Threshold At_h	Specification	Use Case	Analyzer	$AT_{hi} : AT_h\quad name \mid data\quad structure \mid CI \mid forecast$
Datasets D	Specification	Use Case	Analyzer	$D_i : \{D\quad name \mid object \mid type \mid source\}$
Pattern Recognition	Specification	Use Case	Analyzer	$PR_i : \{PR\quad name \mid objectIn \mid ObjectOut \mid action \mid reference\quad data\}$
Conclusion C	Specification	Use Case	Analyzer	$C_i : \{C\quad name \mid use\quad case \mid fact\}$
Report Re	Report	Analyzer	Diagnosis	$Re_i : \{C\quad name \mid use\quad case \mid fact \mid uncertainty \mid trigger\}$

5.1. Object O

The objects $O = \{O_1, ..., O_n\}$, $n \geq 1$ are definitions of the elements from which the system infers knowledge. They are added to the knowledge-base by use case operators. Their function is to describe the nature of the data in order to facilitate the selection of proper preprocessing and prediction methods. Objects are expressed as follows:

$$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\} \tag{1}$$

Examples:
$\{Temperature \mid 1 \mid 1 \mid (-30^\circ, 150^\circ)\}$,
$\{Link \quad Status \mid 0.7 \mid 1 \mid \{"Good", "Normal", "Bad"\}\}$,
$\{Header \quad Encryption \mid 1.5 \mid 1 \mid (True, False)\}$,
$\{Upper \quad Threshold \mid 2 \mid 1 \mid Y_t : t\epsilon T, \forall Y_i \epsilon R\}$

where *object name* acts as identification of the data category and the range of values limits the values that can be assigned. The *weight* is a field reserved for the future implementation of machine learning; it determines priority. Finally, *noValues* anticipates its amount of possible values. Because of this, an object may be specified as a sequence of k previously defined objects or values interrelated. In this case, they are defined as follows.

$$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\} \tag{2}$$

Examples:
$\{pairWeather \mid 1 \mid 2 \mid [temperature][humidity]\}$,
$\{metricA \mid 2 \mid 4 \mid [TTL][lenght][port][ipAddress]\}$,
$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\}$,

The specification of sequences of the same value repeated several times in a row can be simplified by the indicator $: i$, where i is the number of times it repeats. For example, the previous example:

$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

It may be simplified as follows:

$$\{tSerieB \mid 2 \mid 8 \mid [R] : 8\},$$

5.2. Operations Op

The operations $Op = \{Op_1, ..., Op_n\}$, $n \geq 1$ are definitions of binary relationships between facts Fa, objects O or their possible values Va. Initially, the knowledge-base provides a basic battery of operations (ex. All arithmetic operations, propositional logic relationships, basic statistic expressions, etc.). When a use case is on-boarded, operators should declare the set of operations to be taken into account and their restrictions. This is achieved by the following layout:

$$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\} \tag{3}$$

Examples:
$\{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$,
$\{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid left \quad is \quad GE\}$,
$\{And \mid \wedge \mid 1 \mid (Fa) \wedge (Fa) \mid logical \quad conjuction\}$,
$\{Addition \mid + \mid 3 \mid (Fa, Vo) + (Fa, Vo) \mid addition\}$,

where *name* refer to the identification of the operation in the predefined battery, *symbol* is its shortened representation, *priority* its position in the hierarchy of operations, *operands* limits the categories of operands applicable on each side of the binary expression, and *description* briefly explains its functionality in natural language.

5.3. Facts Fa

The facts $Fa = \{Fa_1, ..., Fa_n\}$, $n \geq 1$ are the basic elements of the SELFNET reasoning. They are added to the memory of the Analyzer Module by the Aggregation layer or deduced by the inference engine. Facts are constructed by the linear grammar $GFa = (N, \sum, P, State)$ where $N = State, Operand, \sum = O, Op, Va$ and P is extended as:

$$State \longrightarrow A \quad op \quad A$$
$$Operand \longrightarrow object \mid fact \mid value$$

Facts must be accompanied by a *timestamp* indicating when they have been stated, the *location* on which they are valid and a *weight* that determine their priority. The location refers to SELFNET elements (ex. physical machines, virtual nodes, etc.). The priority is a field reserved by future machine learning weighting. *Uncertainty* describes its probability of being true. Facts are described as the following expression:

$$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\} \tag{4}$$

Examples:
$\{Ur_{threshold} = MaxValue \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\}$,
$\{Temperature \geq 80^\circ \mid 0.7 \mid 0.98 \mid Today \quad 03:41:20 \mid VM15\}$,
$\{KPI7 = UrTh + MaxT \mid 1.2 \mid 1 \mid Today \quad 03:41:20 \mid VM15\}$,

5.4. Rules Ru

The $rules = \{Ru_1, ..., Ru_n\}$, $n \geq 1$ describe how the Analyzer Module acquires new knowledge via rule-based expert system. In order to facilitate their specification, they are declared as propositional logic expressions, and according with the linear grammar $GRu = (N, \sum, P, Rule)$, where $\sum = "True"$, $"False", Facts$, $N = Rule, Atomic \mid Symbol \mid Complex$, and P is expressed as follows:

$$Rule \longrightarrow Atomic \mid Complex \tag{5}$$

$$Atomic \longrightarrow "True" \mid "False" \mid Symbol$$
$$Symbol \longrightarrow Facts$$
$$Complex \longrightarrow \neg Rule \mid (Rule \longrightarrow Rule) \mid (Rule \leftrightarrow Rule) \mid (Rule \wedge Rule) \mid (Rule \vee Rule)$$

Examples:
$(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z)$
$(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B)$
$(Fa(C) \vee Fa(Y)) \vee \neg(Fa(A) \wedge Fa(Z)) \longrightarrow Fa(B)$

The rules are accompanied by the identification of the *usecase* on which they are valid, and their *priority* of inference. Note that in order to enhance scalability, the rules of each use case are totally independent from the others. Rules are detailed as follows:

$$Rule : \{rule \mid priority \mid use \quad case\} \tag{6}$$

Examples:
$\{(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z) \mid 1 \mid SP\}$
$\{(Fa(B)) \longrightarrow Fa(Y) \mid 2 \mid SO\}$
$\{(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B) \mid 1 \mid SH\}$

5.5. Forecast Ft

The $Forecasts = \{Ft_1, ..., Ft_n\}$, $n \geq 1$ are specifications of the objects that must be projected per use case. In this way it is possible to enhance the selection of prediction algorithms and forecasting models. Given the nature of the monitoring environment, *a priori*, this approach only considers predictions on two data types: time series and graphs. The time series allow estimating the evolution of Key Performance Indicators (KPI) or thresholds from concrete locations on SELFNET (physical infrastructure, network devices, virtualization, etc.). The prediction on graphs facilitates the inference of changes on large regions of the SELFNET topology, such as spreading of congestion, inclusion of new network elements or failures. This expert system considers prediction results as facts, so Ft only refers to their specification when on-boarding new use cases. In Figure 1 predictions as facts are declared as $Fa(Ft)$. The following expression describes the forecasts on time series:

$$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\} \tag{7}$$

Examples:
$\{timeSeries \mid O_1 \mid obs \mid t + 5\}$
$\{timeSeries|O_2 \mid time \mid Today \quad 13:28:15\}$

where *timeSeries* is a reserved word indicating that the prediction is on time series, *object* declares the nature of the data to be analyzed, and *domain* is the extension of the prediction. The examples show two reserved words, *obs* (observations) and *time* (timestamp). When the time is measured in observations, the length of the prediction is indicated from the initial time instant t and the amount of coming observations (ex. t + 5 indicates forecast the next five observations). On the other hand, timestamps directly detail how long must be the prediction (ex. Today 13:28:15 indicates the requirement of forecast a certain object between now and 13:28:15 today). Note that the term *timeSeries* is used to describe the way in which data is structured and not the prediction algorithm. A record tracking of this nature could be forecasted by traditional time series methods (autoregressive moving average, exponential smoothing, extrapolation, etc.) but also by other very different approaches (drifting, naive-based algorithms, Artificial Neural Networks-ANN, Support Vector Machines-SVM, etc.). It is up to the decision component of Prediction, select the most appropriate forecasting strategy. If the prediction considers observations on graphs, the forecasts are specified as follows:

$$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\} \tag{8}$$

Examples:
$\{graph \mid O_1 \mid 30 \mid obs \mid t + 20\}$
$\{graph \mid O_2 \mid 45 \mid timestamp \mid Today \quad 19:12:07\}$
$\{graph \mid O_3 \mid 10 \mid timestamp \mid Today \quad 22:30:00\}$

where *graph* is the reserved word to declare predictions on graphs. *object* is the nature of the data on the edges of its incidence matrix. *noVertex* is the number of vertex (i.e., dimension *noVertex-by-noVertex* of its complete adjacency matrix). The last two parameters (*domain* and *length*) have the same function as in the expression of *timeSeries* prediction (indicate the measurement of time and the extension of the prediction).

5.6. Thresholds T_h

The thresholds $T_h = \{T_{h1}, ..., T_{hn}\}$, $n \geq 1$ are specifications of fault tolerance limits related with values assigned to objects O. They are calculated by the SELFNET Aggregation task, but their specification is part of the use case operators. Thresholds are described as the following expression:

$$T_{hi} : T_h \quad name \mid object \tag{9}$$

Examples:
$\{maxTemp \mid O(temperature)\}$
$\{maxConnections \mid O(nConnections)\}$
$\{minQuality \mid O(QoS)\}$

where T_h name is the threshold identification and $object$ is the object on which it acts.

5.7. Adaptive Thresholds T_h

The adaptive thresholds $AT_h = \{AT_{h1}, ..., AT_{hn}\}$, $n \geq 1$ are specification of fault tolerance limits related with values assigned to predictions Ft. They are calculated by the component of prediction of the Analyzer Module, but must be specified by the use case operators. Similarly to the forecast descriptions, initially they act on time series or graphs. They are described as follows:

$$AT_{hi} : AT_h \quad name \mid data \quad structure \mid CI \mid forecast \tag{10}$$

Examples:
$\{maxTemp \mid timeSeries \mid 0.95 \mid Ft(A)\}$
$\{maxWorkload \mid graph \mid 0.90 \mid Ft(X)\}$

where $AT_h \quad name$ is the identification of the adaptive threshold, $data \quad structure$ is *timeSeries* or *graph* depending on the representation of the predicted data, CI is the confidence interval on which it is built by the Adaptive Thresholding component and forecast is the prediction from which it is created.

5.8. Pattern Recognition PR

The pattern recognition configurations $PR = \{PR_1, \ldots, PR_n\}$, $n \geq 1$ are specifications of how facts Fa related with aggregate data are analyzed in order to determine their similarity with previously established reference information. The outputs of pattern recognition actions are facts that display the degree of the similarity observed. Each PR action is defined as follows:

$$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\} \tag{11}$$

Examples:
$\{botnetTraffic \mid O(tFlow) \mid O(dist) \mid match \mid D(dataset1)\}$
$\{paylScan \mid O(payload) \mid O(dist) \mid anomaly \mid D(dataset2)\}$
$\{usrVerify \mid O(uAction) \mid O(dist) \mid anomaly \mid D(dataset3)\}$

where $PR \quad name$ is the action identificator, *objectIn* is the nature of the data to be studied, *objectOut* is the nature of the object recipient of the similarity degree, *action* is the reserved word associated with the type of analysis to be performed. The default actions are "match" for matching observations with the reference data and "anomaly" for outlier detection. Finally, $referencedata$ is the identification of the dataset D to be taken into account.

5.9. Datasets D

The Datasets $D = \{D_1, \ldots, D_n\}$, $n \geq 1$ is the initial reference data to be required by pattern recognition actions. Given that Analyzer Module does not consider online training, all the reference data is provided by the use cases via User Interface. Datasets are declared by the following expression:

$$D_i : \{D \quad name \mid object \mid type \mid source\} \tag{12}$$

Examples:
$\{legitimatePayload \mid O(payload) \mid model \mid Repository1\}$
$\{mySet1|O(flowMetrics) \mid collection \mid Repository2\}$
$\{autoreplicationGens \mid O(binary) \mid signature \mid Repository3\}$

where $D \quad name$ is the dataset identifier and $object$ is the nature of its samples. In this first approach, the dataset can be framed by three types: "collection", "model" or "signature". Firstly, "collection" refers to a set of raw observations directly extracted from the monitoring environment. On the other hand, "model" is a preprocessed description of the data to be analysed. Finally, "signature" indicates exactly patters to be identified. The field $source$ determines where the dataset is found (ex. path, url, repository, etc.).

5.10. Conclusions C

The conclusions $C = \{C_1, \ldots, C_n\}$, $n \geq 1$ are the subset of the group of facts Fa specified for a use case to be satisfied, that form part of the Situational Awareness of the network. When a conclusion is inferred, it is reported to the Diagnostic module [60] for being a potential indicator of situations. These symptoms are defined by use case operators as follows:

$$C_i : \{C \quad name \mid use \quad case \mid fact\} \tag{13}$$

Examples:
$\{gridlock \mid SP \mid Fa(A)\}$
$\{overHeating \mid SH \mid fa(X)\}$

where C name is the conclusion identificator, use case is the associated SELFNET use case, and fact is the triggering conclusion. Conclusions are reported to Diagnostic Module as follows:

$$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\} \tag{14}$$

Examples:
$\{gridlock \mid SP \mid Fa(A) \mid 0.85 \mid Fa(B), Fa(C), Ru(1)\}$
$\{overHeating \mid SH \mid fa(X) \mid 0.75 \mid Fa(x), Ru(3)\}$

where $uncertainty$ the probability of being certain and trigger is the list of rules Ru or facts Fa that take part of its inference.

6. Examples of Specification and Workflows

This section describes three examples of data specification and workflows on the Analyzer Module.

6.1. UC 1: Device Temperature Analysis

This section describes an example of a sensor related with self-healing use case.

6.1.1. Description

The use case ($myTemp$) requires identifying symptoms related with overheat on network devices. This is a very basic example where prediction and adaptive thresholding are not considered. Therefore the decision thresholds are static and were built at Aggregation.

6.1.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

6.1.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the temperature of the devices and its upper threshold.

$$O_1 : \{Temperature \mid 1 \mid 1 \mid R\}$$
$$T_{h1} : \{maxTemp \mid O_1\}$$

Second is indicating the operators that are required and how they are taken into account:

$$Op_1 : \{Equal\ |{=}|\ 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT\ |{\geq}|\ 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions to be satisfied:

$$C_1 : \{overheat \mid myTemp \mid Fa(O_1) \geq Fa(T_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(T_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid myTemp\}$$

6.1.4. Workflow

At runtime, Aggregation layer notify to the Analyzer Module facts related with myTemp use case. Some of them concern the temperature on SELFNET devices, for example:

$$Fa_1 : \{O_1 = 35^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\}$$
$$Fa_2 : \{O_1 = 76^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeB\}$$
$$\ldots$$
$$Fa_5 : \{O_1 = 80^\circ \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

Note that uncertainty is 1 because the sensors are deterministic (100% probability of provide the correct temperature). The facts refer to the static thresholding are:

$$Fa_7 : \{T_{h1} = 79^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid All\}$$
$$Fa_8 : \{T_{h1} = 79^\circ \mid 1 \mid 1 \mid Today \quad 12:22:16 \mid All\}$$

These facts are provided by Aggregation, and they are directly included on the memory of the Analyzer Module. If they are updated for the same location (ex. Fa_5 and Fa_6), the latest version is considered by the inference engine. After certain period of observation, the inference engine tries to deduct new knowledge from the rule-set of every use case. In myTemp, the Analyzer Module tries to infer conclusions for Ru_1. At $Today \quad 12:22:17$ the systems satisfy the first conclusion: $Fa_5(O_1 = 80^\circ) \geq Fa_8(O_1 = 79^\circ)$, so the fact $Fa(C_1)$ is added to memory:

$$Fa_9 : \{Fa_5 \geq Fa_8 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

The location NodeB is considered because it is the more restrictive between NodeB,All. So the symptom C_1 has been discovered, and it is reported to Diagnostic Module as follows:

$$Re_1 : \{overheat \mid myTemp \mid Fa_9 \mid 1 \mid Fa_5, Fa_8, Ru_1\}$$

The inference engine will continue operating looking for new symptoms.

6.2. UC 2: Network Congestion Analysis

This section describes an example of a sensor related with self-optimization use case.

6.2.1. Description

The use case to be managed (Self-Congestion (SC)) requires identifying symptoms related with traffic congestion on SELFNET elements. In this example, prediction and adaptive thresholding are considered.

6.2.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

6.2.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the congestion level monitored and its prediction.

$$O_1 : \{congestion \mid 1 \mid 1 \mid [0,1]\}$$
$$Ft_1 : \{timeSeries \mid O_1 \mid obs \mid t+3\}$$

Next, they define an adaptive threshold to be automatically generated from the information provided by the record tracking and the Adaptive Thresholding.

$$AT_{h1} : \{maxCongestion \mid timeSeries \mid 0.95 \mid Ft_1\}$$

Second, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions are identified:

$$C_1 : \{gridlock \mid SC \mid Fa(O_1) \geq Fa(AT_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid SC\}$$

6.2.4. Workflow

At runtime, Aggregation layer notify to the Analyzer Module facts related with the SC use case, for example:

$$\begin{aligned}
Fa_1 &: \{O_1 = 0.6 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ServerA\} \\
Fa_2 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ServerA\} \\
Fa_3 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ServerA\} \\
Fa_4 &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:21 \mid ServerA\} \\
Fa_5 &: \{O_1 = 0.68 \mid 1 \mid 1 \mid Today \quad 12:22:24 \mid ServerA\} \\
&\ldots. \\
Fa_{44} &: \{O_1 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:50 \mid ServerA\} \\
Fa_{45} &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:52 \mid ServerA\} \\
Fa_{47} &: \{O_1 = 0.69 \mid 1 \mid 1 \mid Today \quad 12:22:56 \mid ServerA\} \\
Fa_{48} &: \{O_1 = 0.86 \mid 1 \mid 1 \mid Today \quad 12:22:58 \mid ServerA\} \\
Fa_{49} &: \{O_1 = 0.97 \mid 1 \mid 1 \mid Today \quad 12:23:01 \mid ServerA\}
\end{aligned}$$

The construction of predictive models requires certain amount of previous observations; in this case, it considered the first 45 facts. They are handled by the record tracking in order to extract the needed information and define time series. At *Today* 12 : 22 : 52 (when Fa_{45} is deducted), the forecasting component provides the first prediction Ft_1 for the instant t+3. Then a new fact is included to the memory:

$$Fa_{46} : \{AT_{h1} = 90 \mid 1 \mid 1 \mid Today \ \ 12:23:52 \mid ServerA\}$$

It triggers the rule Ru_1, because $Fa_49(O_1 = 0.97) \geq Fa_46(AT_{h1} = 90)$, and the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50} : \{Fa_{49} \geq Fa_{46} \mid 1 \mid 1 \mid Today \ \ 12:23:01 \mid NodeB\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1 : \{gridlock \mid SC \mid Fa_{50} \mid 1 \mid Fa_{49}, Fa_{46}, Ru_1\}$$

6.3. UC 3: Payload Analysis

This section describes an example of a sensor related with self-protection use case.

6.3.1. Description

This use case (Self-Guard(SG)) requires identifying symptoms related with anomalous payloads on SELFNET traffic. In this example, pattern recognition actions are considered.

6.3.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions. The external repositories (Rep1, Rep2) provide collection of Legitimate (Rep1) and malicious (Rep2) SELFNET traffic observations.

6.3.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account; in this example they are the payload of the SELFNET traffic O_1, its similarity with the legitimate payload dataset O_2 and the malicious samples O_3.

$$O_1 : \{payload \mid 1 \mid 1 \mid hexadecimal\}$$
$$O_2 : \{simLegi \mid 1 \mid 1 \mid \{0...1\}\}$$
$$O_3 : \{simMal \mid 1 \mid 1 \mid \{0...1\}\}$$

Second, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LT \mid > \mid 1 \mid (Fa, O, Va) > (Fa, O, Va) \mid leftisG\}$$

Next, they define the datasets to be taken into account.

$$D_{legi} : \{legitimatePayload \mid O(payload) \mid collection \mid Rep1\}$$
$$D_{mal} : \{maliciousPayload \mid O(payload) \mid collection \mid Rep2\}$$

And then the pattern recognition actions to be executed:

$$PR_1 : \{legMeasure \mid O_1 \mid O_2 \mid anomaly \mid D(D_{legi})\}$$
$$PR_2 : \{malMeasure \mid O_1 \mid O_3 \mid anomaly \mid D(D_{mal})\}$$

Conclusions are identified as follows:

$$C_1 : \{maliciousContent \mid SC \mid Fa(O_2) < Fa(O_3)\}$$

And the following rules are onboarded:

$$Ru_1 : \{Fa(O_2) < Fa(O_3) \longrightarrow Fa(C_1) \mid 1 \mid SP\}$$

6.3.4. Workflow

At runtime, Aggregation layer notifies to the Analyzer Module facts related with the SG use case:

$$Fa_1 : \{O_1 = FF217 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\}$$
$$Fa_2 : \{O_1 = FFFFF \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionB\}$$
$$Fa_3 : \{O_1 = 00DE8 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_4 : \{O_1 = A4FC9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_5 : \{O_1 = FF218 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\}$$
$$....$$
$$Fa_{38} : O_1 = F0279 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA$$

Each time a new payload is observed, the SELFNET Analyzer Module performs the pattern recognition actions PR_1 and PR_2. This returns new facts:

$$\begin{array}{r}
Fa_{1R1}: \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\
Fa_{1R2}: \{O_3 = 0.2 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\
Fa_{2R1}: \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\
Fa_{2R2}: \{O_3 = 0.18 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\
Fa_{3R1}: \{O_2 = 0.8 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\
Fa_{3R2}: \{O_3 = 0.21 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\
\ldots.
\end{array}$$

At *Today* $12:22:23$ the following facts are discovered:

$$\begin{array}{r}
Fa_{32R1}: \{O_2 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\} \\
Fa_{32R2}: \{O_3 = 0.92 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\}
\end{array}$$

They trigger the rule Ru_1, because $Fa_{32R1}(O_2 = 0.66) < Fa_{32R2}(O_3 = 0.92)$, and then the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50}: \{Fa_{32R1} < Fa_{32R1} \mid 1 \mid 1 \mid Today \quad 12:23:23 \mid ConexionA\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1: \{suspiciousPayload \mid SC \mid Fa_{50} \mid 1 \mid Fa_{32R1}, Fa_{32R2}, Ru_1\}$$

7. Discussion

Analysis and intelligence capabilities play an important role to address 5G requirements, in combination with key-enabled technologies such as SDN, NFV, cloud computing, etc. All of these domains can take advantage of forecasting, pattern recognition, artificial intelligence and advanced intelligence concepts. In this way, 5G networks will be able to provide enhanced capacities related to the network management and the detection of possible harmful problems. For its part, the diagnosis of data information is required in order to know what the real cause of the event is. Because of this, the intelligence is provided in two phases: (i) analysis stage and (ii) decision-making; similar to a medical evaluation, where firstly the symptoms are detected and then based on it a treatment is applied. In general terms, the proposed architecture and data specification for enhancing a use-case driven analysis on 5G networks presupposes substantial improvements over previous approaches. On the one hand, 5G data analysis is partially covered in some works [27,30,31]. These proposals take into account specific requirements such as context awareness of radio components [27]. In [30] a context aware resource allocation algorithm based on the user mobility is presented. This work proposes some resource management schemes, handover procedures and cell activations. For its part, Apajalahti et al. [31] use ontologies and statistical reasoning in order to analyze and configure the mobile network. This approach can be used as a complementary methodology to the SELFNET Analyzer approach. On the other hand, some ongoing works [28,29,36] are still at an early stage and are complementary to our proposal. METIS project [28] is dealing with 5G radio access network components (e.g., spectrum usage or air interface). In [29] intelligent capabilities applied to virtualized environments are proposed. Furthermore, Charisma Project [36] introduces intelligent mobile cloud in order to meet low latency and security requirements.

To the best of our knowledge, the SELFNET Analyzer framework is the first proposal that provides a generalized framework to deal with both traditional technologies and currently 5G key-enabled technologies. The SELFNET Analyzer Module provides a general purpose scheme easily adapted to the operator needs and hence to overcome the design constraints in different monitoring environments. This is a very important feature bearing in mind the great amount of technologies that can be part of a 5G scenario, as well those that still under development to be deployed in the near future [10,14,15].

Note that SELFNET Analyzer Module is able to analyse information from heterogeneous sources such as SDN elements, virtual devices or metrics from specialized sensors. Another important characteristic is that the SELFNET Analyzer facilitates the incorporation of different analysis strategies, such as novel prediction or pattern recognition algorithms. This framework was developed to be able to operate indistinctly with very different data mining and machine learning paradigms, among them conventional information, big data or high dimensional data. The use of any of them does not imply design changes, being simply an implementation problem. As a result, this proposal is easily adaptable to future projects.

The proposed data specification to accommodate the onboarding of new use cases is simple and adjustable. This is also corroborated by the fact that SELFNET has been able to incorporate every use case without modifications on the original definitions. This does not mean that in future use cases, more relevant changes will not be required. But without doubt, this robustness provides a solid base for design analytic schemes on similar contexts. Note that SELFNET implements a triad of services: self-protection, self-healing and self-optimization with completely different features and dependences (metrics, network devices to be monitored, prediction/pattern recognition algorithms, etc.). But despite these advantages, the proposal presents some weakness, most of them related with the limitations previously mentioned at the design principles (see Section 3). For example, the SELFNET Analyzer is not able to deal with complex stationary monitoring environments [42], where the quality of the analytics will decrease with time. Given the importance of this kind of scenarios on network environments, this is an aspect that must be studied.

Another point to be keep in mind is that, according to the experience of the SELFNET consortium, the effectiveness of the Analyzer Module depends on the quality of their specification. It means that once deployed, the approach follows the guidelines provided by operators, which indicate what information should be processed, how it should be analysed and what results can be obtained from it. Despite of the simplicity of the proposed data specification, if the operator makes errors, there is a greatest chance of unexpected results. So in this sense, its robustness and scalability imply a high dependence on the quality of the specification inserted by operators where configuring the analyzer functionality. It is important to emphasize that loading new use cases is based only on configuration changes, without the need to modify the Analyzer implementation or to include additional software. Furthermore, there are a number of challenges that need to be addressed related to how the data received from underlying layers will be organized or how the analysis process will be performed. Regarding to data organization is important to determine if the data will be processed as a raw data or in an aggregated manner because it may become a performance issue. This information will be loaded and converted into facts by the Analyzer framework in order to provide the network state in real time. Another important aspect to bear in mind is the execution pipeline of the Analyzer components in order to provide consistence and facilitate the organization of the received information. Thus, the investigation of methods to process and analyze the received information is also part of the ongoing work.

8. Conclusions

In this paper, the application of analysis and intelligence capabilities and how these concepts are used in 5G networks were explained. We introduced the design of SELFNET Analyzer Module and its data specification. Our design provides pattern recognition, reasoning and prediction capabilities to infer the possible symptoms, facilitating the diagnosis and decision-making tasks, which are part of future work. The main contribution of SELFNET Analyzer Module is its general, simple and scalable approach, allowing new rules and metrics in the analysis process when a new use case is added by the operator. SELFNET sensors gather information from several data sources such as virtual elements, LTE, SDN and traditional network devices; and thus the gathered information can be subject of analysis. Furthermore, this proposal was built to support new analytic capabilities by means of a plugin based approach. Meanwhile, the implementation of Analyzer Module is part of ongoing work as well as the introduction of mechanisms to work in non-stationary monitoring environments.

Acknowledgments: This work is supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Lorena Isabel Barona López are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT (Quito, Ecuador) under Convocatoria Abierta 2013 Scholarship Program.

Author Contributions: The authors contributed equally to this research. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. NGMN Alliance. NMGN 5G White Paper 2015. Available online: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf (accessed on 9 January 2017).
2. Agyapong, P.K.; Iwamura, M.; Staehle, D.; Kiess, W.; Benjebbour, A. Design Considerations for a 5G Network Architecture. *IEEE Commun. Mag.* **2014**, *52*, 65–75.
3. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76.
4. Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1–27.
5. ETSI Industry Specification Group (ISG). Network Function Virtualization (NFV) Architectural Framework. Available online: http://www.etsi.org/technologies-clusters/technologies/nfv (accessed on 9 January 2017)
6. Mijumbi, R.; Serrat, J.; Gorricho, J.L.; Bouten, N.; Turck, F.; Boutaba, R. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 236–262.
7. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud Computing: State-of-the-art and Research Challenges. *J. Int. Serv. Appl.* **2010**, *1*, 7–18.
8. Baldo, N.; Giupponi, L.; Mangues-Bafalluy, J. Big Data Empowered Self Organized Networks. In Proceedings of the 20th European Wireless Conference, Barcelona, Spain, 14–16 May 2014.
9. Aliu, O.G.; Imran, A.; Imran, M.A.; and Evans, B. A Survey of Self Organisation in Future Cellular Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 336–361.
10. Imran, A.; Zoha, A.; Abu-Dayya, A. Challenges in 5G: How to Empower SON with Big Data for enabling 5G. *IEEE Netw.* **2014**, *28*, 27–33.
11. Quick, D.; Choo, K.K.R. Digital forensic Intelligence: Data Subsets and Open Source Intelligence (DFINT + OSINT): A timely and Cohesive Mix. *Future Gener. Comput. Syst.* **2016**, doi:10.1016/j.future.2016.12.032.
12. Quick, D.; Choo, K.K.R. Big Forensic Data Management in Heterogeneous Distributed Systems: Quick Analysis of Multimedia Forensic Data. Software: Practice and Experience. *J. Netw. Comput. Appl.* **2016**, doi:10.1002/spe.2429.
13. Demestichas, P.; Georgakopoulos, A.; Karvounas, D.; Tsagkaris, K.; Stavroulaki, V. 5G on the horizon: Key Challenges for the Radio-Access Network. *IEEE Veh. Technol. Mag.* **2013**, *8*, 47–53.
14. Boccardi, F.; Heath, R.W.; Lozano, A.; Marzetta, T.L.; Popovski, P. Five Disruptive Technology directions for 5G. *IEEE Commun. Mag.* **2014**, *52*, 74–80.
15. Akyildiz, I.F.; Lin, S.C.; Wang, P. Wireless Software-Defined Networks (W-SDNs) and Network Function Virtualization (NFV) for 5G Cellular Systems: An Overview and Qualitative Evaluation. *Comput. Netw.* **2015**, *93*, 66–79.
16. Abdelwahab, S.; Hamdaoui, B.; Guizani, M.; Znati, T. Network Function Virtualization in 5G. *IEEE Commun. Mag.* **2016**, *54*, 84–91.
17. Lin, X.; Choo, K.K.R.; Lin, Y.D.; Mueller, P. Guest Editorial: Network Forensics and Surveillance for Emerging Networks. *IEEE Netw.* **2016**, *30*, 4–5.
18. 5G Infrastructure Public Private Partnership—5G PPP. Available online: https://5g-ppp.eu (accessed on 16 December 2016).
19. 5G Americas, 2016. Available online: http://www.5gamericas.org/es/ (accessed on 16 December 2016).

20. Barona López, L.I.; Valdivieso Caraguay, Á.L.; Maestre Vidal, J.; Sotelo Monge, M.A.; García Villalba, L.J. Towards Incidence Management in 5G based on Situational Awareness. *Future Internet* **2017**, *9*, 1–15.
21. 5G Ensure. Deliverable D 2.3, Risk Assessment, Mitigation and Requirements (Draft). Available online: http://www.5gensure.eu/deliverables (accessed on 19 December 2016).
22. SELFNET Project. Framework for Self-Organized Network Management in Virtualized and Software Defined Networks. Available online: https://selfnet-5g.eu/ (accessed on 15 February 2017).
23. Tahaei, H.; Salleh, R.; Khan, S.; Izard, R.; Choo, K.K.R.; Anuar, N.B. A multi-objective Software Defined Network Traffic Measurement. *Measurement* **2016**, *95*, 317–327.
24. Agiwal, M.; Roy, A.; Saxena, N. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1617–1655.
25. Liu, D.; Wang, L.; Chen, Y.; Elkashlan, M.; Wong, K.K.; Schober, R. User association in 5G networks: A survey and an outlook. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1018–1044.
26. Fenton, N.; Neil, M. Making decisions: Using Bayesian Nets and MCDA. *Knowl. Syst.* **2001**, *14*, 307–325.
27. Marquezan, C.C.; Mahmood, K.; Zafeiropoulos, A.; Krishna, R.; Huang, X.; An, X.; Corujo, D. Context Awareness in Next Generation of Mobile Core Networks. Available online: https://arxiv.org/ftp/arxiv/papers/1611/1611.05353.pdf (accessed on 9 January 2017).
28. Tullberg, H.; Popovski, P.; Li, Z.; Uusitalo, M.A. The METIS 5G System Concept–Meeting the 5G Requirements. *IEEE Commun. Mag.* **2016**, *54*, 132–139.
29. CONTENT Project. Convergence of Wireless Optical Network and iT rEsources iN SupporT of Cloud Services. FP7-ICT. Project Reference: 318514, Funded under: FP7-ICT. Available online: http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/content-factsheet.pdf (accessed on 9 January 2017).
30. Kuruvatti, N.P.; Schotten, H.D. Framework to Support Mobility Context Awareness in Cellular Networks. In Proceedings of the IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, China, 15–18 May 2016.
31. Apajalahti, K.; Eero, H.; Juha, N.; Vilho, R. StaRe: Statistical Reasoning Tool for 5G Network Management. In Proceedings of the 2016 Semantic Web-ESWC, Heraklion, Greece, 29 May–2 June 2016.
32. Martin, B.A.; Marinos, L.; Rekleitis, E.; Spanoudakis, G.; Petroulakis, N.E; Threat Landscape and Good Practice Guide for Software Defined Networks/5G. Available online: http://openaccess.city.ac.uk/15504/7/SDN%20Threat%20Landscape.pdf (accessed on 9 January 2017).
33. You, I.; Sharma, V.; Atiquzzaman, M.; Choo, K.K.R. GDTN: Genome-Based Delay Tolerant Network Formation in Heterogeneous 5G Using Inter-UA Collaboration. *PLoS ONE* **2016**, *11*, 1–37.
34. Neves, P.; Calé, R.; Costa, M.R.; Parada, C.; Parreira, B.; Alcaraz-Calero, J.; Wang, Q.; Nightingale, J.; Chirivella-Perez, E.; Jiang, W.; et al. The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm. *Int. J. Distrib. Sens. Net.* **2016**, *2016*, 1–17.
35. 5G-NORMA Project. 5G NOvel Radio Multiservice Adaptive Network Architecture. Project Reference: 671584. Funded under: H2020-ICT-2014-2. Available online: https://5gnorma.5g-ppp.eu/ (accessed on 9 January 2017).
36. CHARISMA Project. Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access. Project Reference: 671704. Funded under: H2020-ICT-2014-2. Available online: http://www.charisma5g.eu// (accessed on 9 January 2017).
37. Xu, L.; Assem, H.; Yahia, I.G.B.; Buda, T.S. CogNet: A Network Management Architecture Featuring Cognitive Capabilities. In Proceedings of the 2016 European Conference on Networks and Communications (EuCNC), Athens, Greece, 27–30 June 2016.
38. Endsley, N.R. Design and Evaluation for Situation Awareness Enhancement. In Proceedings of the 32nd Annual Meeting on Human Factors and Ergonomics Society, Anaheim, CA, USA, 24–28 October 1988; Volume 32, pp. 97–101.
39. Sivarajah, U.; Kamal, M.M.; Irani, Z.; Weerakkody, V. Critical analysis of Big Data Challenges and Analytical Methods. *J. Bus. Res.* **2017**, *70*, 263–286.
40. Heijungs, R.; Henriksson, P.J.; Guinée, J.B. Measures of Difference and Significance in the Era of Computer Simulations, Meta-Analysis, and Big Data. *Entropy* **2016**, *18*, 361.
41. Holte, R.C. Very Simple Classification Rules Perform well on most commonly used Datasets. *Mach. Learn.* **1993**, *11*, 63–90.
42. Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R. Learning in Nonstationary Environments: A Survey. *IEEE Comput. Intell. Mag.* **2015**, *10*, 12–25.

43. Bouveyron, C.; Brunet-Saumard, C. Model-based Clustering of high-dimensional Data: A Review. *Comput. Stat. Data Anal.* **2014**, *71*, 52–78.
44. De Sanctis, M.; Bisio, I.; Araniti, G. Data Mining Algorithms for Communication Networks Control: Concepts, Survey and Guidelines. *IEEE Netw.* **2016**, *30*, 24–29.
45. Meng, W.; Li, W.; Kwok, L.F. EFM: Enhancing the Performance of signature-based Network Intrusion Detection Systems using enhanced Filter Mechanism. *Comput. Secur.* **2014**, *43*, 189–204.
46. Aggarwal, C.C. Outlier Analysis. Available online: http://www.charuaggarwal.net/outlierbook.pdf (accessed on 15 January 2017).
47. Katris, C.; Daskalaki, S. Comparing Forecasting Approaches for Internet Traffic. Expert Systems with Applications. *Expert Syst. Appl.* **2015**, *42*, 8172–8183.
48. Gardner, E.S.; Dannenbring, D.G. Forecasting with Exponential Smoothing: Some Guidelines for Model Selection. *Decis. Sci.* **1980**, *11*, 370–383.
49. Kadri, F.; Harrou, F.; Chaabane, S.; Sun, Y.; Tahon, C. Seasonal ARMA-based SPC Charts for Anomaly Detection: Application to Emergency Department Systems. *Neurocomputing* **2016**, *173*, 2102–2114.
50. Berlingerio, M.; Pinelli, F.; Calabrese, F. Abacus: Apriori-based Community Discovery in Multidimensional Networks. *Data Min. Knowl. Discov.* **2013**, *27*, 294–320.
51. Radenkovic, M.; Grundy, A. Efficient and Adaptive Congestion Control for Heterogeneous delay-Tolerant Networks. *Ad Hoc Netw.* **2012**, *10*, 1322–1345.
52. Zhang, T.; Wang, J.; Huang, J.; Huang, Y. Adaptive Marking Threshold Method for delay-sensitive TCP in Data Center Network. *J. Netw. Comput. Appl.* **2016**, *61*, 222–234.
53. Boem, F.; Ferrari, R.M.; Keliris, C.; Parisini, T.; Polycarpou, M.M. A Distributed Networked Approach for Fault Detection of Large-Scale Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 18–33.
54. Venkatesan, R.; Er, M.J. A Novel Progressive Learning Technique for Multi-class Classification. *Neurocomputing* **2016**, *207*, 310–321.
55. Hayes-Roth, F.; Waterman, D.A.; Lenat, D.B. *Building Expert Systems*; Addison-Wesley: Boston, MA, USA, 1983.
56. Mas, M.; Monserrat, M.; Ruiz-Aguilera, D.; Torrens, J. RU and (U,N)-implications Satisfying Modus Ponens. *Int. J. Approx. Reason.* **2016**, *73*, 123–137.
57. Morsi, N.N.; Fahmy, A.A. On Generalized Modus Ponens with Multiple Rules and a Residuated Implication. *Fuzzy Sets Syst.* **2002**, *129*, 267–274.
58. Chen, H.; Li, T.; Luo, C.; Horng, S.J.; Wang, G. A Decision-Theoretic Rough Set Approach for Dynamic Data Mining. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 1958–1970.
59. Gilio, A. Generalizing Inference Rules in a Coherence-based Probabilistic default Reasoning. *Int. J. Approx. Reason.* **2012**, *53*, 413–434.
60. SELFNET Consortium. Deliverable 5.3: Report and Prototypical Implementation of the Integration of the Algorithms and Techniques Used to Provide Intelligence to the Decision-Making Framework. Available online: https://selfnet-5g.eu/2016/12/15/deliverables-online/ (accessed on 9 January 2017).
61. Talon, A.; Curt, C. Selection of Appropriate Defuzzification Methods: Application to the Assessment of Dam Performance. *Expert Syst. Appl.* **2017**, *70*, 160–174.