(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 March 2021 (18.03.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/048742 A1

(51) International Patent Classification:
H04L 12/24 (2006.01) *G06F 11/30* (2006.01)
H04W 24/10 (2009.01) *H04L 12/26* (2006.01)

(21) International Application Number:
PCT/IB2020/058346

(22) International Filing Date:
08 September 2020 (08.09.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/897,695 09 September 2019 (09.09.2019) US

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: **KALLUS, Zsófia**; Javorka Adam u. 54. 1/5, H-1147 Budapest (HU). **BORSOS, Tamas**; Bács u. 16, H-1161 Budapest (HU). **KERSCH, Péter**; Drégelyvár u. 33. 3. em. 9, H-1158 Budapest (HU). **VADERNA, Peter**; Kisfaludy u.93, H-1174 Budapest (HU).

(74) Agent: **HERRERA, Stephen**; 1400 Crescent Green, Suite 300, Cary, North Carolina 27518 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) Title: SYSTEM AND METHOD OF SCENARIO-DRIVEN SMART FILTERING FOR NETWORK MONITORING

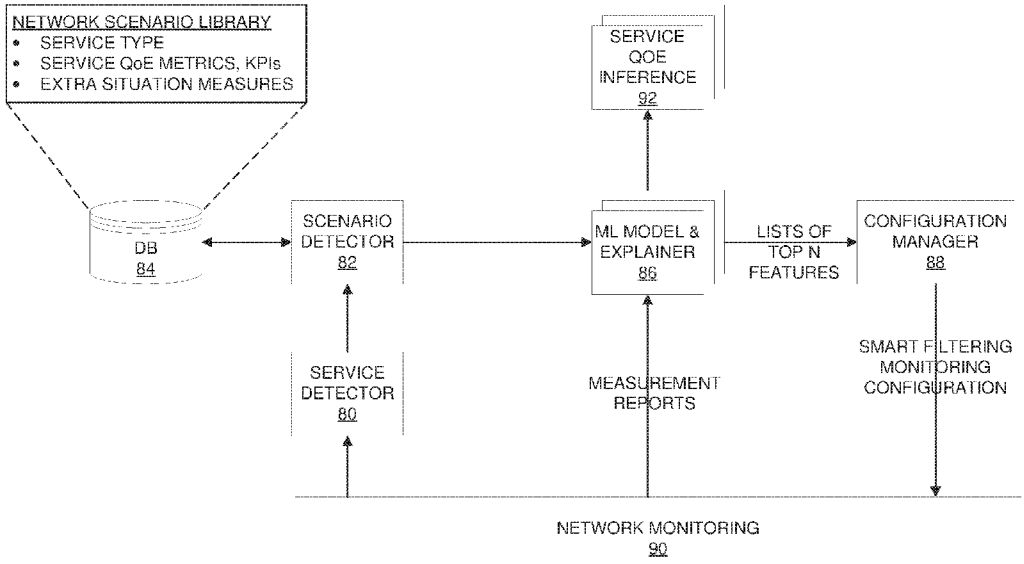

FIG. 3

(57) Abstract: A network node (140) leverages machine learning models and their corresponding model explainers (86) to automatically optimize network monitoring configuration for service assurance. The active services of a user session serve to identify applicable service assurance scenarios. Using a knowledge base representing corresponding machine learning models and their corresponding model explainers, one or more low-level features to be reported are selected. The selected features are those that are determined to have greatest relative importance on service-level performance indicators. The metrics associated with the selected features are then input into the network configuration management system.

WO 2021/048742 A1

[Continued on next page]

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

SYSTEM AND METHOD OF SCENARIO-DRIVEN SMART FILTERING FOR NETWORK MONITORING

RELATED APPLICATIONS

This application claims priority to U.S. Application No. 62/897,695, filed 9 September 2019, the disclosure of which is incorporated in its entirety by reference herein.

TECHNICAL FIELD

The present disclosure relates generally to network monitoring systems, and more particularly, to the automatic optimization of network monitoring using machine learning (ML) models.

BACKGROUND

The Quality of Experience (QoE) in telecom services is affected by the performance of various network components. End-to-end service assurance, therefore, relies on the horizontal monitoring of low-level event streams from a variety of sources. These low-level event streams are heterogeneous event time series that can be correlated on a per user basis, such as is done in the Ericsson Expert Analytics (EEA) solution, to create per user session descriptors, for example, for the duration of a call. Upon detecting a degradation of performance, these network monitoring reports should also provide information useful for root cause analysis.

SUMMARY

Embodiments of the present disclosure leverage machine learning (ML) models and state-of-the-art feature impact analysis to enable the automatic optimization of network monitoring in a closed-loop, flexible control framework.

In one embodiment, the present disclosure provides a closed-loop method implemented by a network node of a communication network. In this embodiment, the method comprises identifying a service assurance scenario for a user equipment (UE) based on feedback received from a network monitoring system, selecting, from a repository, a machine learning (ML) model and a corresponding model explainer based on the identified service assurance scenario, determining, based on a feature impact analysis of the ML model selected from the repository, a list of N features defining one or more metrics to be measured and reported by the network monitoring system, and configuring the network monitoring system to measure and report the one or more metrics based on the list of N features.

Additionally, embodiments of the present disclosure also provide a network node in a communication network. In one embodiment, the network node comprises an interface circuit and a processing circuit. The interface circuit is configured for communication with one or more nodes in the communication network. The processing circuit is configured to identify a service

assurance scenario for a user equipment (UE) based on feedback received from a network monitoring system, select, from a repository, a machine learning (ML) model and a corresponding model explainer based on the identified service assurance scenario, determine, based on a feature impact analysis of the ML model selected from the repository, a list of N features defining one or more metrics to be measured and reported by the network monitoring system, and configure the network monitoring system to measure and report the one or more metrics based on the list of N features.

In one embodiment, the present disclosure provides a computer program product stored on a non-transitory computer readable medium. In this embodiment, the computer program product comprises instructions that, when executed by at least one processor of a network node, causes the network node to identify a service assurance scenario for a user equipment (UE) based on feedback received from a network monitoring system, select, from a repository, a machine learning (ML) model and a corresponding model explainer based on the identified service assurance scenario, determine, based on a feature impact analysis of the ML model selected from the repository, a list of N features defining one or more metrics to be measured and reported by the network monitoring system, and configure the network monitoring system to measure and report the one or more metrics based on the list of N features.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a general network monitoring architecture.

Figure 2 illustrates an exemplary communication network according to one embodiment of the present disclosure.

Figure 3 illustrates an exemplary architecture, functional units, and logical steps of a closed-loop control of a flexible monitoring system that implements smart-filtering logic for resource optimization according to one embodiment of the present disclosure.

Figure 4 is a graph illustrating the top N features in terms of aggregate feature impact, and the top N features in terms of top maximum absolute model impact, for individual samples according to one embodiment of the present disclosure.

Figure 5 explains some feature naming convention used in the embodiment of Figure 4.

Figure 6 illustrates an exemplary closed-loop method implemented by a network node in a communication network according to one embodiment of the present disclosure.

Figure 7 is a block diagram illustrating some components of a network node configured according to one embodiment of the present disclosure.

Figure 8 is a functional block diagram illustrating some functional modules/units executing on a processing circuit of a network node according to one embodiment of the present disclosure.

DETAILED DESCRIPTION

Embodiments of the present disclosure leverage machine learning (ML) models and their explainers to automatically optimize network monitoring configuration for service assurance. In particular, the present embodiments provide a near real-time solution where the active services of a user session, as well as any additional content information for the user, are used to identify applicable service assurance scenarios. For example, such context information may be used according to one embodiment to determine that a user is located at or near the edge of a cell and/or is experiencing undesirable radio conditions. Using a knowledge base representing corresponding ML models and their explainers, the disclosure dynamically specifies the low-level features to be reported with highest impact on the service-level performance indicators. Not only does this "smart filtering" aspect of the present embodiments serve as input for a network configuration management, but it also creates a dynamic and optimized closed-loop control solution for flexible monitoring.

Referring now to the drawings, Figure 1 is a functional block diagram illustrating a general architecture 10 for monitoring a network. As seen in Figure 1, a Network Management System 30 receives raw measurement data from different domains 20 of the network. Examples of such domains 20 include, but are not limited to, the Radio Access Network (RAN) 22, the core network 24, the IP Multimedia Subsystem (IMS) 26, and the passive probing 28 on transmission lines. The Network Management System 30, usually comprising an Operating Support System (OSS) 32 and Business Support System (BSS) 34, is responsible for collecting and processing the raw data into measurements.

The node logging and reporting functions are configurable in mobile networks. Logging can, in some cases, be highly detailed such that it is possible to generate fine granularity reports. By way of example only, the radio nodes in 4G mobile networks are able to report per-user events and measurements from various protocol layers down to sub-second granularity in hundreds of different report types and including thousands of different event parameters. In 5G networks, the possibility for even more detailed logging is expected.

However, the ability to produce such granular reports is not without its issues. For example, setting an appropriate level of logging is a complex optimization problem. Although increasing the granularity and the level of reporting detail improves network observability, it also implies higher costs due to an increase in the use of resource in the monitoring and analytics systems, as well as in the network nodes themselves. Because of the prohibitive cost of such increased resource usage, enabling all reporting at the finest granularity (i.e., the highest level of detail) is not economical at scale. Other network monitoring setups, such as passive probing systems and UE-side reporting (e.g., MDT), for example, experience similar tradeoffs between observability and resource costs.

Currently, state-of-the-art network monitoring methods maintain monitoring loads at acceptable levels by filtering measurements at the node level, the sub-system level, by report

types, or by users. Further improvements can be achieved by dynamically changing the reporting configurations for different scenarios. For example, U.S. Pat. No. 8,909,208, to Loborg, et. al., provides a solution for reporting of a NodeB network component. Optimization in Loborg is based on node type, hour of the week, number of end users, or traffic volume. Other systems, such as those described in U.S. Pat. Nos. 8,787,901 and 9,955,373 to Racz, et. al. and Siomina et. al., respectively, dynamically change reporting configurations responsive to different activity triggers, states, or context information. Still other conventional systems, such as the one described in U.S. Pat. No. 9,026,851 to Mondal et al, triggers more detailed tracing responsive to detecting problem conditions, which also facilitates the subsequent performance of a root cause analysis.

Another technique that may be used in service assurance is Machine Learning (ML)/Artificial Intelligence (AI). One dominant area in adopting ML/AI for service assurance lies in the detection of anomalies and performance degradation. While each sub-system of a network can be treated as separate models, predictive models for end-to-end QoE metrics operate based on the results of horizontal monitoring of network components. The resulting heterogeneous event time series need to be vectorized to create the input of the ML models. Ultimately, root cause analysis rules created by domain experts should be replaced by automated methods. This would allow the implementation of closed feedback loops for the network management system to self-correct subs-system configurations where possible.

One example of this is described in the paper authored by Veena Mendiratta entitled, "Unsupervised Anomaly Detection and Root Cause Analysis in Mobile Networks", ODSC India, September 1, 2018. This paper describes a method in which procedure-based data acquisition is used for system-wide failure detection. The method implements unsupervised anomaly detection and finite-state machine based Root Cause Analysis (RCA) for network outage detection. The input, which is generated for each 10-second time interval, is created from network log data that has been vectorized into a fixed dimensional representation of the network state. Any problematic subsequences or failure message patterns that are identified can infer root cause via the use of additional error codes.

Currently, the area of interpretable ML is a hot topic for ML research efforts. Normally considered a “black box” in the stage of inference, the goal of interpretable ML research is to explain the inner logic of a trained model. Explanations can be provided, for example, in the form of a feature impact analysis. For example, given a trained model, (e.g., a boosted decision tree or a neural network) and an input vector, the research efforts aim to determine the role of each feature in forming the output of the model (i.e., the inferred label prediction). In the SHapley Additive exPlanations (SHAP) method, for example, (see e.g., “A Unified Approach to Interpreting Model Predictions”, arXiv:1705.07874v2, 2017 to Lundberg et. al.), an “explainer” or “model explainer” is generated from an ML model.

Multiple factors are considered when model explainers perform feature impact analysis. One consideration, for example, is the overall absolute importance, or the distribution of per input vector importance, of a given feature is measured from a set of input vectors. The effect of a feature relative to the average effect of all features is also considered to determine whether the feature is pushing the output over an average with a positive effect, or pulling the output below the average with a negative effect. These effects can be regarded as a force, with both a direction and a quantified strength. With this analysis, an order of importance can be created for the features or single input vectors can be analyzed.

None of the existing network monitoring technologies provide fully automated methods to intelligently select a set of the most important parameters to be reported in a given scenario. Rather, this particular task is conventionally defined manually by human experts. However, this is costly, takes a lot of time to implement for new technologies, services, or scenarios, and often times yields sub-optimal results. Additionally, conventional solutions are inflexible and do not allow for dynamically adapting to changes that are made to the network (e.g., software updates, configuration updates, changes in usage patterns, etc.). As telco systems become increasingly more complex (e.g., 4G → 5G transition), the scale of these issues is also expected to increase. The present disclosure, however, addresses such issues by leveraging the ML models and their corresponding model explainers to automatically optimize network monitoring configuration for service assurance.

An exemplary embodiment of the disclosure will be described in the context of a 5G or NR wireless communication network 40, such as the one seen in Figure 2. Those skilled in the art will appreciate, however, that the methods and apparatus herein described are not limited to use in 5G or NR networks, but may also be used in wireless communication networks 40 where multiple beams within a single cell are used for communication with wireless devices in the cell.

In more detail, Figure 2 illustrates a wireless communication network 40 according to the NR standard currently being developed by Third Generation Partnership Project (3GPP). The wireless communication network 40 comprises one or more base stations 50 providing service to user equipment (UEs) 60a, 60b in respective cells 70 of the wireless communication network 40. The base stations 50 are also referred to as Evolved NodesBs (eNBs) and gNodeBs (gNBs) in 3GPP standards. Although only one cell 70 and one base station 50 are shown in Figure 2, those skilled in the art will appreciate that a typical wireless communication network 40 comprises many cells 70 served by many base stations 50.

One feature of NR networks is the ability of the base stations 50 to transmit and/or receive on multiple beams in the same cell 70. Figure 2 illustrates two such beams 72a, 72b (collectively, "beams 72"), although the number of beams 72 in a cell 40 may be different.

The UEs 60a, 60b may comprise any type of equipment capable of communicating with the base station 20 over a wireless communication channel. For example, UEs 60a, 60b may comprise cellular telephones, smart phones, laptop computers, notebook computers, tablets,

machine-to-machine (M2M) devices (also known as machine type communication (MTC) devices), embedded devices, wireless sensors, or other types of wireless end user devices capable of communicating over wireless communication networks 40.

As previously stated, embodiments of the present disclosure leverage ML models and state-of-the-art feature impact analysis to enable the automatic optimization of network monitoring in a closed-loop, flexible control framework. In particular, the ML models use low-level network measurement inputs to predict service experience measures. The automated feature impact analysis methods are performed on top of these models to reveal the relative predictive power of each low-level input feature. According to the present embodiments, the network monitoring can therefore dynamically be configured in a closed-loop manner based on this analysis to report only on those features that are determined to have the most impact on the indicators associated with the service-level performance.

To manage multiple services, monitor multiple service experience measures, and perform root cause analysis for different quality degradations, embodiments of the present disclosure use a knowledge base representing service assurance scenarios. In the context of this disclosure, a "scenario" is defined as the context-dependent list of active ML models that are used either to infer quality of experience metrics for active services, or to perform root cause analysis upon service degradations. Scenarios can be detected dynamically, and a configuration management system of the present disclosure will receive, within defined monitoring constraints, information representing a union of the top most impactful features (i.e., the features that are determined as having the most impact on service-level performance indicators) from each of the corresponding ML models. This scenario detection can be performed, according to the present embodiments, on a per-subscriber basis, a per-network function basis, or on a network node basis.

The embodiments described herein provide advantages and benefits that conventional systems and methods of network monitoring do not or cannot provide. For example, embodiments of the present disclosure are referred to as performing "smart filtering" since they are automated and data-driven, and thus, minimize the reporting of low-level dimensions. Such "smart filtering" reduces the monitoring load with minimal loss of performance in service assurance functionalities.

Further, because the present embodiments are automatic and data-driven, they create a closed control loop control for various network monitoring optimization problems, such as the issues related to the tradeoff between network observability and monitoring load. Additionally, with the present embodiments, important low-level network metrics that are required to characterize a service QoE are automatically identified and configured to be reported by the monitoring system. This advantageously provides good network observability dynamically adapted to eventual changes, while keeping the monitoring load as low as possible. The closed loop automation of the present embodiments also decreases network operations costs and

makes it easier and faster to rollout new complex network services (e.g., 5G, NB-IoT, delay critical industry systems, etc.).

Thus, a system and method configured according to the present disclosure leverages the ML models that are built for service assurance use cases for smart filtering optimization techniques in order to monitor the system more efficiently. To accomplish this function, the system utilizes a Network Scenario Library (NSL). The NSL represents a knowledge base comprising information related to the services and QoE measures. In particular, the library comprises expert knowledge related to the available network services and service assurance scenarios. For each type of service that is available, the library specifies the QoE metrics and the corresponding performance indicators along with any extra situational measures or descriptors.

The NSL also comprises, for each service type and QoE measure pair, a method for embedding raw event streams and a trained ML model (e.g., an extreme boosted tree regressor). One or more corresponding model explainers (e.g., SHAP explainers) are also available on a per-model basis from the NSL. The model explainers provide a list of the top *N* features of the corresponding ML model along with an indication of each feature's respective relative importance in the QoE prediction process. According to the present embodiments, *N* can be fixed, or *N* can be an optional configuration parameter of the optimization, or *N* can be dynamically derived from network monitoring and service assurance constraints (e.g., maximum network monitoring load, maximum model degradation, etc.).

The particular type and methods for training and re-training the learning models, as well as the details of those methods, are not germane to the present embodiments. However, those of ordinary skill in the art should readily appreciate that, according to the present disclosure, training and/or retraining the models could be performed periodically or continuously, or online when necessary, depending on the timescale of changes in the underlying network.

Figure 3 illustrates an architecture, functional units, and logical steps of a closed-loop control of a flexible monitoring system that implements smart-filtering logic for resource optimization according to one embodiment of the present disclosure. As previously stated, the filtering is data-driven and leverages ML models created for Service Assurance QoE inference.

As seen in Figure 3, the architecture comprises a Service Detector 80, a Scenario Detector 82, a Network Scenario Library (NSL) 84, ML model(s) and model explainers 86, a Configuration Manager 88, a Network Monitoring System 90, and a Service QoE Inference function 92.

The Service Detector 80 identifies the active service(s) in a user session from the Network Monitoring feedback. The detected services are input into the Scenario Detector 82, where the Network Scenario Library (NSL) 84 is used to acquire the corresponding ML and model explainer(s) 86, respectively. Through each model explainer, a list of respective the top N features considered to have the most impact on the predictions of service-level performance is

generated. That is, the top N features represent the low-level metrics and can be logged and reported as having the highest impact on one or more target variables in the ML model.

These feature lists are used by the Configuration Manager 88 to configure the Network Monitoring System 90 by implementing the proposed smart-filtering method of the present embodiments. Specifically, in order to optimize the high-level Service QoE inference function 92 with a restricted monitoring budget, only the most impactful features of the low-level events will be reported to the service assurance inference functions. In embodiments where multiple services are being considered, a union of the corresponding multiple parallel scenarios is considered as the basis for the configuration.

In one embodiment, the Configuration Manager 88 sends a list of events/reports to the Network Monitoring System 90 identifying the events to be activated in a specified node for a given UE 60. However, the present embodiments are not so limited. In one or more other embodiments, for example, the Configuration Manager 88 provides a rule set, a condition list, or a program agent to be activated in the given nodes, which may be required if local decisions are to be made. Some possible reasons for making local decisions may be (i) the need to make such decisions in real-time; or (ii) to use a wider set of internal events/reports to start/stop/select events for final reporting.

Thus, the method according to the present embodiments helps to ensure that the monitoring optimization best serves the service assurance scenarios that are related to the active services of the user. The higher the model's predictive power and the lower the monitoring load, the better the optimization.

In one embodiment, the control loop is closed within the nodes themselves. In these cases, the Configuration Manager 88 can implement smart filtering. In another embodiment, however, the Configuration Manager 88 is deployed within the OSS system 32.

It should be noted that the embodiment seen in Figure 3 deploys solutions for the automated optimization of near real-time monitoring in communication networks, such as the one in network 40 seen in Figure 2. However, those of ordinary skill in the art should readily appreciate that the present embodiments are not so limited. According to the present disclosure, the components seen in Figure 3 can be part of local system configuration solution, but may also be realized as cloud-native micro-services.

VoLTE Service Assurance with smart filtering

As an aid in understanding the present disclosure, a first embodiment is implemented using actual mobile network data. In particular, the first embodiment considers a single, simplified scenario for Voice over LTE (VoLTE) service assurance where the VoLTE QoE is measured via a Mean Opinion Score (MOS) metric on a scale of 1 - 5. Using an analytics product, low-level RAN, core network and IMS reports are correlated into VoLTE call session records from both legs of each call. Each call has been partitioned into 10 second long slices.

Further, for both RTP stream directions in those slices 1) vectorized representations containing >1000 parameters (input features) are created and 2) a MOS score is computed from RTCP reports sent by a subset of the UEs (labels). Using this data, decision tree ensemble models are trained to infer the VoLTE MOS metric. A feature impact analysis is then performed on the model using the SHAP methodology.

Figure 4 is a graph illustrating an overview of which features are most important for a model. In particular, the graph plots the SHAP values of some features for samples. Those of ordinary skill in the art will appreciate how to interpret such graphs; however, for further reading on such graphs, the interested reader is directed to https://github.com/slundberg/shap.

In more detail, Figure 4 is a graph illustrating the top N features 100 in terms of the combination of two feature importance metrics - aggregate feature impact, and the top maximum absolute model impact for individual samples. In this example, there are 20 such features, and thus, N=20. Figure 4 also illustrates a bar graph 110 indicating the impact of each feature 100 on the model output from low impact to high impact. The top feature 102 (i.e., "*ul-ran_period_state_ave_ul_pusch_sinr*") relates to the average PUSCH SINR on the uplink side, and as indicated by bar 112, is the top feature based on the aggregate model impact. Another feature 104 (i.e., "*dl_ran_period_state_cqi_0*") relates to the ratio of "CQI=0" samples on the downlink side. As indicated by its corresponding bar 114, a high value for this feature decreases the MOS value on the model output by up to 1.0. A set of other features 106 relate to features having a small aggregate impact on the model output, but as indicated by the corresponding bars 116, can still have a high impact individually in rare cases. Typically, such high impacts are associated with failure events.

Configuring reporting only for those top 20 features in the mobile network, rather than for a list of many features (e.g., over 1000 features), can significantly decrease the monitoring load with only small deterioration in model quality (e.g., ~10% as is the example above). Therefore, according to the present disclosure, the process of selecting what to report in the Network Monitoring System 90 can be fully automated. Specifically, operators need only to specify the desired constraints, such as, for example, the maximum reporting volume in the Network Monitoring System 90, how much model degradation can be allowed, and the like. So defined, a system configured according to the present disclosure can utilize the desired constraints to automatically select the number of top N features to keep.

Figure 5 illustrates the feature naming convention explanations 120 for the embodiment of Figure 4. In particular, the feature name is a concatenation of:

Direction + Blockname + Event Name/Event Parameter + Event Value

By way of example only, consider feature 104 (i.e., "*dl_ran_period_state_cqi_0*") in Figure 4. For this feature, the "*dl*" is the Direction portion of the feature name and indicates that the feature represents a measurement on the downlink leg of the RTP stream. The

"*ran_period_state*" is the Blockname portion of the feature name and indicates that the measurement refers to RAN data collected for the entire duration of the analyzed call session slice. The "*cqi*" is the EventName/Event Parameter portion of the feature name and is the name of the actual measured RAN metric. Finally, the "0" is the Event Value portion of the feature name and indicates a value for the CQI. Thus, in this aspect, the feature 104 is "the ratio of samples with CQI = 0."

Multi-service Service Assurance with Smart Filtering

This embodiment relates to scenario detection according to the present disclosure. As an illustrative example, scenario detection is described for a mobile network subscriber X.

In one example scenario, consider subscriber X who starts a VoLTE call. In response, the Service Detector 80 identifies the start of this VoLTE session from IMS signaling. The NSL 84 contains an ML model for the VoLTE service that is used to infer a MOS metric for this service. Scenario Detector 80 then obtains a list of the top low-level network metrics that are required for this model from NSL 84 and provides the list to the Configuration Manager 88 with instructions to activate the monitoring of these metrics for subscriber X.

Aspects of the present embodiments can also be implemented in scenarios other than call scenarios. For example, consider a case where subscriber X begins watching a YouTube video once the VoLTE call is finished. In some embodiments, NSL 84 can be configured to store a plurality of ML models for the YouTube service. By way of example only, NSL 84 in this embodiment stores three ML models - one ML model to infer YouTube session boundaries, one ML model to infer video bitrate, and one ML model to infer stall time metrics.

According to the present disclosure, the Service Detector 80 in this example scenario is configured to determine that subscriber X has started watching the YouTube video based on the DNS requests made by subscriber X for YouTube video servers. Additionally, Service Detector 80 determines the set of QoE metrics to be inferred for YouTube sessions and fetches the corresponding ML models and their explainers provided by NSL 84. The model explainers provide a list of the top *N* most impactful features required for those models to the Configuration Manager 88, along with an instruction to activate the monitoring of the measurements corresponding to those features for subscriber X.

Those of ordinary skill in the art should appreciate that the present disclosure can be implemented when subscribers are in fixed positions, and when they are moving. For example, consider a situation in which subscriber X is sitting on a bus while watching the YouTube video. At some point, as the bus is moving, the bus would approach a cell-edge and the radio conditions would get worse. Responsive to detecting such a cell-edge situation, the Scenario Detector 82 would obtain an additional fourth ML model from NSL 84 – used for in-depth radio-level root cause analysis - and provide the list of the top low-level network metrics that are required for this model to the Configuration Manager 88 along with instructions to activate the

monitoring of these metrics for subscriber X. The Configuration Manager 88, is then used for updating the low-level metrics to be monitored for subscriber X based on the top N features of the 4th model.

Figure 6 illustrates an exemplary closed-loop method 130 implemented by a network node in a communication network (e.g., network 40) according to one embodiment of the present disclosure. In particular, method 130 begins with the network node identifying a service assurance scenario for a user equipment (UE) based on feedback received from a network monitoring system (box 132). The network node then selects, from a repository, a machine learning (ML) model and a corresponding model explainer based on the identified service assurance scenario (box 134), and determines, based on a feature impact analysis of the ML model selected from the repository, a list of N features defining one or more metrics to be measured and reported by the network monitoring system (box 136). So determined, the network node configures the network monitoring system to measure and report the one or more metrics based on the list of N features (box 138).

According to the embodiments of method 130, the ML model comprises information used to predict end-to-end Quality of Experience (QoE) metrics for a given service type from performance measurements associated with the UE, and one or more descriptors describing the network performance measurements.

In some embodiments of method 130, selecting the ML model and the corresponding model explainer based on the identified service assurance scenario comprises determining one or more active services associated with the UE, and obtaining, from the repository, the ML model predicting the QoE metrics for the one or more active services.

In one embodiment, the model explainer comprises information indicating a respective relative importance of each of the input features of the ML model in predicting the QoE metrics.

Thus, in one embodiment, selecting the list of N features based on the respective relative importance of each feature in predicting the QoE metrics.

In some embodiments of method 130, N is a fixed value.

In other embodiments, however, N is a configurable value.

In some embodiments, configuring the network monitoring system comprises configuring the network monitoring system to measure and report the one or more metrics responsive to one or more predefined events.

In other embodiments, configuring the network monitoring system comprises providing the network monitoring system with a set of rules for measuring and reporting the one or more metrics.

In still other embodiments, configuring the network monitoring system comprises providing a list of one or more conditions to the network monitoring system defining the conditions under which the network monitoring system will measure and report the one or more metrics.

In at least one embodiment, configuring the network monitoring system comprises activating a program agent in each of one or more nodes of the network monitoring system to measure and report the one or more metrics.

In some embodiments of method 130, selecting the ML model and the corresponding model explainer comprises selecting, from the repository, a plurality of ML models and corresponding model explainers. In these embodiments, each ML model is associated with a corresponding different service assurance scenario and defines a list of M input features. Further, each input feature defines one or more network performance metrics to be measured and reported by the network monitoring system.

Additionally, in some embodiments, each of the plurality of ML models provides information used in predicting the end-to-end QoE metrics for a respective different service type.

In some embodiments of method 130, each model explainer in the plurality of model explainers provides information indicating a respective relative importance of the M input features in its corresponding ML model in predicting the QoE metrics.

According to some embodiments of method 130, the one or more metrics defined in the list of N features comprises a union of the one or more network performance metrics defined in one or more of the lists of M input features.

Thus, in at least one embodiment, configuring the network monitoring system comprises configuring the network monitoring system to measure and report the union of the one or more network performance metrics.

An apparatus can perform any of the methods herein described by implementing any functional means, modules, units, or circuitry. In one embodiment, for example, the apparatus comprises respective circuits or circuitry configured to perform the steps shown in the method figures. The circuits or circuitry in this regard may comprise circuits dedicated to performing certain functional processing and/or one or more microprocessors in conjunction with memory. For instance, the circuitry may include one or more microprocessor or microcontrollers, as well as other digital hardware, which may include Digital Signal Processors (DSPs), special-purpose digital logic, and the like. The processing circuitry may be configured to execute program code stored in memory, which may include one or several types of memory such as read-only memory (ROM), random-access memory, cache memory, flash memory devices, optical storage devices, etc. Program code stored in memory may include program instructions for executing one or more telecommunications and/or data communications protocols as well as instructions for carrying out one or more of the techniques described herein, in several embodiments. In embodiments that employ memory, the memory stores program code that, when executed by the one or more processors, carries out the techniques described herein.

Figure 7 illustrates a network node 140 according to one embodiment that may be configured to function as herein described. The network node 140 comprises processing

circuitry 142, a memory 144 configured to store a computer program 146, and communication interface circuitry 148.

The processing circuity 142 controls the overall operation of the network node 140 and processes the signals sent to or received by the network node 140. Such processing can include, but is not limited to, coding and modulation of transmitted data signals, and the demodulation and decoding of received data signals. The processing circuity 142 may comprise one or more microprocessors, hardware, firmware, or a combination thereof, and as stated above, is configured to execute a control program, such as computer program 146, to perform the previously described functions.

Memory 144 comprises both volatile and non-volatile memory for storing computer program code and data needed by the processing circuity 142 for operation. Memory 144 may comprise any tangible, non-transitory computer-readable storage medium for storing data including electronic, magnetic, optical, electromagnetic, or semiconductor data storage. Memory 144 stores, as stated above, a computer program 146 comprising executable instructions that configure the processing circuity 142 to implement method 130 according to Figure 6 as described herein. A computer program in this regard may comprise one or more code modules corresponding to the means or units described above.

In general, computer program instructions and configuration information are stored in a non-volatile memory, such as a ROM, erasable programmable read only memory (EPROM) or flash memory. Temporary data generated during operation may be stored in a volatile memory, such as a random access memory (RAM). In some embodiments, computer program 146 for configuring the processing circuity 142 as herein described may be stored in a removable memory, such as a portable compact disc, portable digital video disc, or other removable media. The computer program 146 may also be embodied in a carrier such as an electronic signal, optical signal, radio signal, or computer readable storage medium.

The communications interface circuitry 148 is configured to communicate data, signals, and information with other devices and/or systems via the network 40. In some cases, the communications interface circuitry 148 may be coupled to one or more antennas and comprise the radio frequency (RF) circuitry needed for transmitting and receiving signals over a wireless communication channel. In other cases, the communications interface circuitry 148 is configured to send and receive such information via an ETHERNET-based network.

Figure 8 illustrates processing circuitry 142 for a network node 140configured in accordance with one or more embodiments. In this embodiment, the processing circuitry 142 comprises a scenario identification module/unit 150, a model obtaining module/unit 152, a feature determination module/unit 154, and a network monitoring system configuration module/unit 156. The various modules/units 150, 152, 154, and 156 can be implemented by hardware and/or by software code that is executed by a processor or processing circuit.

The scenario identification module/unit 150 is configured to identify a service assurance scenario for a user equipment (UE) based on feedback received from a network monitoring system. The model obtaining module/unit 152 is configured to select, from a repository, a machine learning (ML) model and a corresponding model explainer based on the identified service assurance scenario, as previously described. The feature determination module/unit 154 is configured to determine, based on a feature impact analysis of the ML model selected from the repository, a list of N features defining one or more metrics to be measured and reported by the network monitoring system, as previously described. The network monitoring system configuration module/unit 156 is configured to configure the network monitoring system to measure and report the one or more metrics based on the list of N features, as previously described.

Those of ordinary skill in the art will readily appreciate that aspects of the present disclosure may be executed as one or more network functions on the processing circuit 142 of a single network node 140, or on the processing circuits 142 of multiple network nodes 140 in the communication network 40. Further, aspects of the present disclosure may be implemented on a virtual network node.

Those skilled in the art will also appreciate that embodiments herein further include corresponding computer programs. A computer program comprises instructions which, when executed on at least one processor of an apparatus, cause the apparatus to carry out any of the respective processing described above. A computer program in this regard may comprise one or more code modules corresponding to the means or units described above.

Embodiments further include a carrier containing such a computer program. This carrier may comprise one of an electronic signal, optical signal, radio signal, or computer readable storage medium.

In this regard, embodiments herein also include a computer program product stored on a non-transitory computer readable (storage or recording) medium and comprising instructions that, when executed by a processor of an apparatus, cause the apparatus to perform as described above.

Embodiments further include a computer program product comprising program code portions for performing the steps of any of the embodiments herein when the computer program product is executed by a computing device. This computer program product may be stored on a computer readable recording medium.

Additional embodiments will now be described. At least some of these embodiments may be described as applicable in certain contexts and/or wireless network types for illustrative purposes, but the embodiments are similarly applicable in other contexts and/or wireless network types not explicitly described.

Any appropriate steps, methods, features, functions, or benefits disclosed herein may be performed through one or more functional units or modules of one or more virtual apparatuses.

Each virtual apparatus may comprise a number of these functional units. These functional units may be implemented via processing circuitry, which may include one or more microprocessor or microcontrollers, as well as other digital hardware, which may include digital signal processors (DSPs), special-purpose digital logic, and the like. The processing circuitry may be configured to execute program code stored in memory, which may include one or several types of memory such as read-only memory (ROM), random-access memory (RAM), cache memory, flash memory devices, optical storage devices, etc. Program code stored in memory includes program instructions for executing one or more telecommunications and/or data communications protocols as well as instructions for carrying out one or more of the techniques described herein. In some implementations, the processing circuitry may be used to cause the respective functional unit to perform corresponding functions according one or more embodiments of the present disclosure.

Generally, all terms used herein are to be interpreted according to their ordinary meaning in the relevant technical field, unless a different meaning is clearly given and/or is implied from the context in which it is used. All references to a/an/the element, apparatus, component, means, step, etc. are to be interpreted openly as referring to at least one instance of the element, apparatus, component, means, step, etc., unless explicitly stated otherwise. The steps of any methods disclosed herein do not have to be performed in the exact order disclosed, unless a step is explicitly described as following or preceding another step and/or where it is implicit that a step must follow or precede another step. Any feature of any of the embodiments disclosed herein may be applied to any other embodiment, wherever appropriate. Likewise, any advantage of any of the embodiments may apply to any other embodiments, and vice versa. Other objectives, features and advantages of the enclosed embodiments will be apparent from the description.

The term unit may have conventional meaning in the field of electronics, electrical devices and/or electronic devices and may include, for example, electrical and/or electronic circuitry, devices, modules, processors, memories, logic solid state and/or discrete devices, computer programs or instructions for carrying out respective tasks, procedures, computations, outputs, and/or displaying functions, and so on, as such as those that are described herein.

Some of the embodiments contemplated herein are described more fully with reference to the accompanying drawings. Other embodiments, however, are contained within the scope of the subject matter disclosed herein. The disclosed subject matter should not be construed as limited to only the embodiments set forth herein; rather, these embodiments are provided by way of example to convey the scope of the subject matter to those skilled in the art.

CLAIMS

What is claimed is:

1. A closed-loop method (130) implemented by a network node (140) in a communication network (40), the method comprising:

identifying (132) a service assurance scenario for a user equipment (UE) (60) based on feedback received from a network monitoring system (90);

selecting (134), from a repository (84), a machine learning (ML) model and a corresponding model explainer (86) based on the identified service assurance scenario;

determining (136), based on a feature impact analysis of the ML model selected from the repository, a list of N features (100) defining one or more metrics to be measured and reported by the network monitoring system; and

configuring (138) the network monitoring system to measure and report the one or more metrics based on the list of N features.

2. The method according to claim 1 wherein the ML model comprises information used to predict end-to-end Quality of Experience (QoE) metrics (92) for a given service type from performance measurements associated with the UE, and one or more descriptors describing the network performance measurements.

3. The method according to any of claims 1-2 wherein selecting the ML model and the corresponding model explainer based on the identified service assurance scenario comprises:

determining one or more active services associated with the UE; and

obtaining, from the repository, the ML model predicting the QoE metrics for the one or more active services.

4. The method according to any of claims 1-3 wherein the model explainer provides information indicating a respective relative importance of each of the input features of the ML model in predicting the QoE metrics.

5. The method according to claim 4 further comprising selecting the list of N features based on the respective relative importance of each feature in predicting the QoE metrics.

6. The method according to claim 5 wherein N is a fixed value.

7. The method according to claim 5 wherein N is a configurable value.

8. The method according to any of claims 1-7 wherein configuring the network monitoring system comprises configuring the network monitoring system to measure and report the one or more metrics responsive to one or more predefined events.

9. The method according to any of claims 1-7 wherein configuring the network monitoring system comprises providing the network monitoring system with a set of rules for measuring and reporting the one or more metrics.

10. The method according to any of claims 1-7 wherein configuring the network monitoring system comprises providing a list of one or more conditions to the network monitoring system defining the conditions under which the network monitoring system will measure and report the one or more metrics.

11. The method according to any of claims 1-7 wherein configuring the network monitoring system comprises activating a program agent in each of one or more nodes of the network monitoring system to measure and report the one or more metrics.

12. The method according to any of claims 1-11 wherein selecting the ML model and the corresponding model explainer comprises selecting, from the repository, a plurality of ML models and corresponding model explainers, wherein each ML model is associated with a corresponding different service assurance scenario and defines a list of M input features, and wherein each input feature defines one or more network performance metrics to be measured and reported by the network monitoring system.

13. The method according to claim 12 wherein each of the plurality of ML models provides information used in predicting the end-to-end QoE metrics for a respective different service type.

14. The method according to any of claims 12-13 wherein each model explainer in the plurality of model explainers provides information indicating a respective relative importance of the M input features in its corresponding ML model in predicting the QoE metrics.

15. The method according to any of claims 12-14 wherein the one or more metrics defined in the list of N features comprises a union of the one or more network performance metrics defined in one or more of the lists of M input features.

16. The method according to any of claims 12-15 wherein configuring the network monitoring system comprises configuring the network monitoring system to measure and report the union of the one or more network performance metrics.

17. A network node (140) in a communication network (40), the network node comprising:
communications interface circuitry (148) configured for communication with one or more nodes in the communication network; and
a processing circuit (142) configured to:
identify (132) a service assurance scenario for a user equipment (UE) (60) based on feedback received from a network monitoring system (90);
select (134), from a repository, a machine learning (ML) model and a corresponding model explainer (86) based on the identified service assurance scenario;
determine (136), based on a feature impact analysis of the ML model selected from the repository, a list of N features (100) defining one or more metrics to be measured and reported by the network monitoring system; and
configure (138) the network monitoring system to measure and report the one or more metrics based on the list of N features.

18. The network node according to claim 17 further configured to perform the method of any of claims 2-16.

19. A non-transitory computer-readable storage medium (144) containing a computer program (146) comprising executable instructions that, when executed by a processing circuit (142) in a network node (140) in a communication network (40) causes the network node to perform any one of the methods of claims 1-16.

20. A computer program (146) comprising instructions which, when executed by at least one processing circuit (142) of a network node (140), causes the network node to perform the method of any of claims 1-16.

21. A carrier containing the computer program of claim 20, wherein the carrier is one of an electronic signal, optical signal, radio signal, or computer readable storage medium.

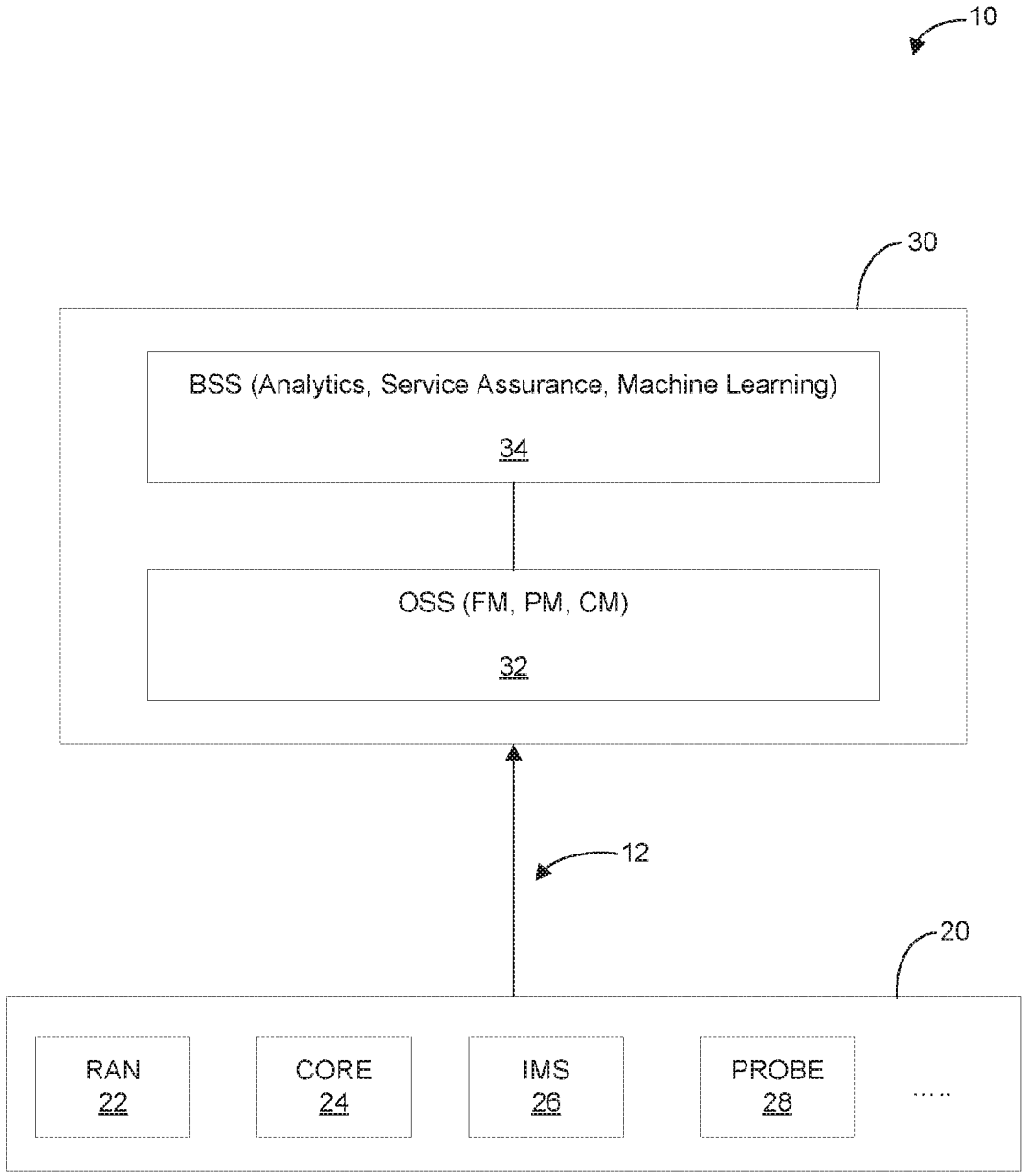
10
30
BSS (Analytics, Service Assurance, Machine Learning)
34
OSS (FM, PM, CM)
32
12
20
RAN
22
CORE
24
IMS
26
PROBE
28
.....

FIG. 1

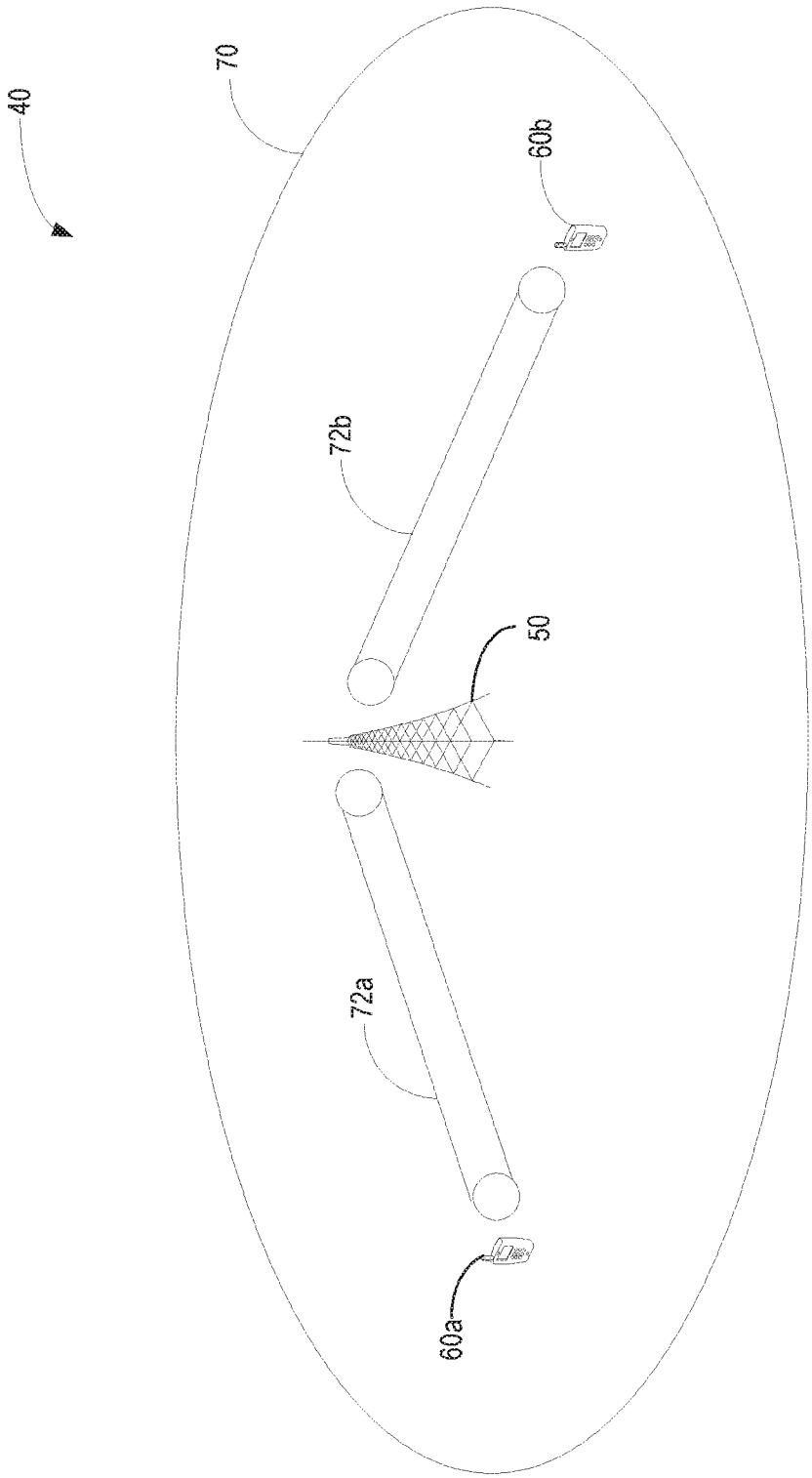

FIG. 2

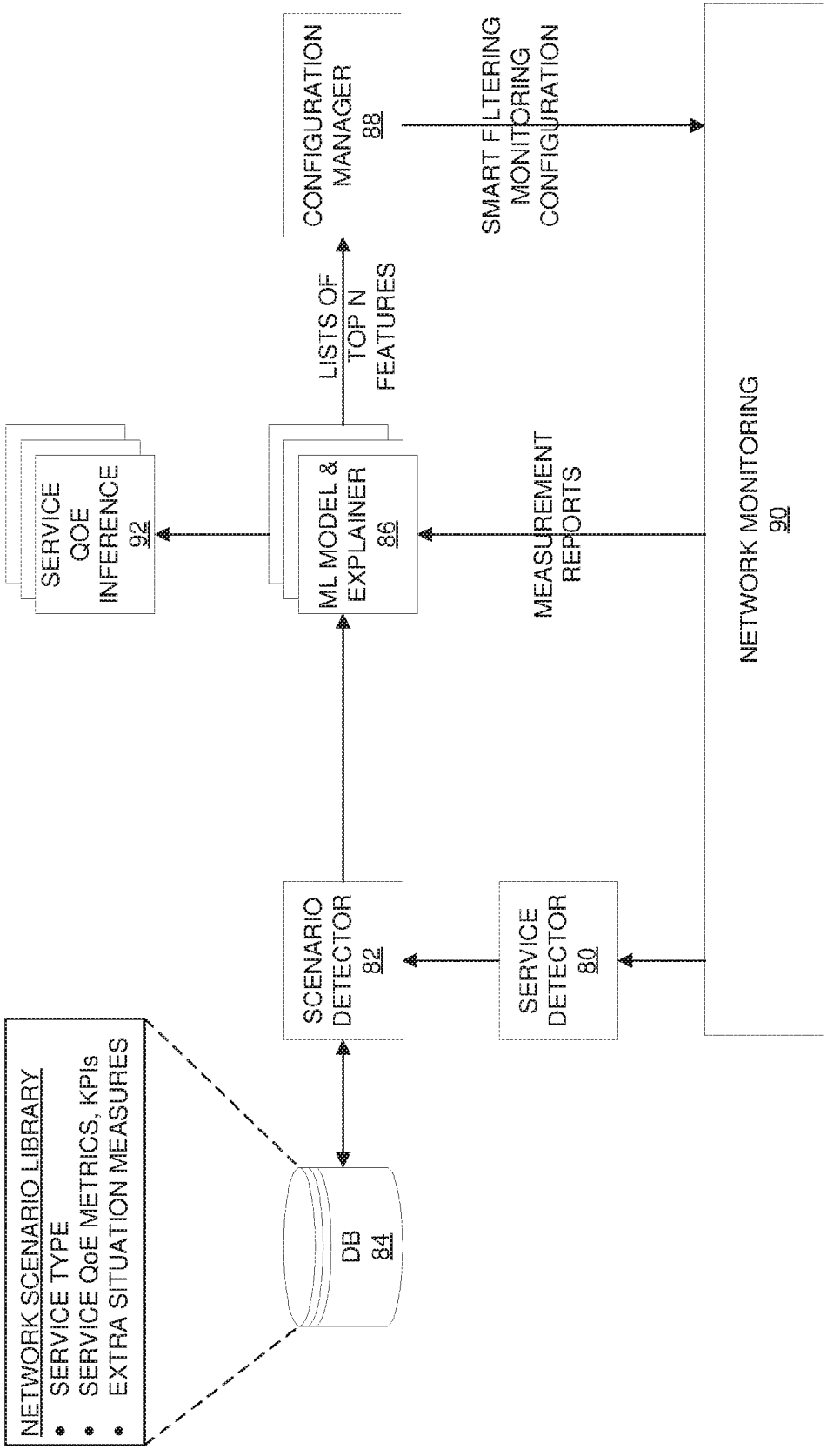
NETWORK SCENARIO LIBRARY
• SERVICE TYPE
• SERVICE QoE METRICS, KPIs
• EXTRA SITUATION MEASURES
DB 84
SCENARIO DETECTOR 82
SERVICE DETECTOR 80
NETWORK MONITORING 90
ML MODEL & EXPLAINER 86
SERVICE QOE INFERENCE 92
MEASUREMENT REPORTS
LISTS OF TOP N FEATURES
CONFIGURATION MANAGER 88
SMART FILTERING MONITORING CONFIGURATION

FIG. 3

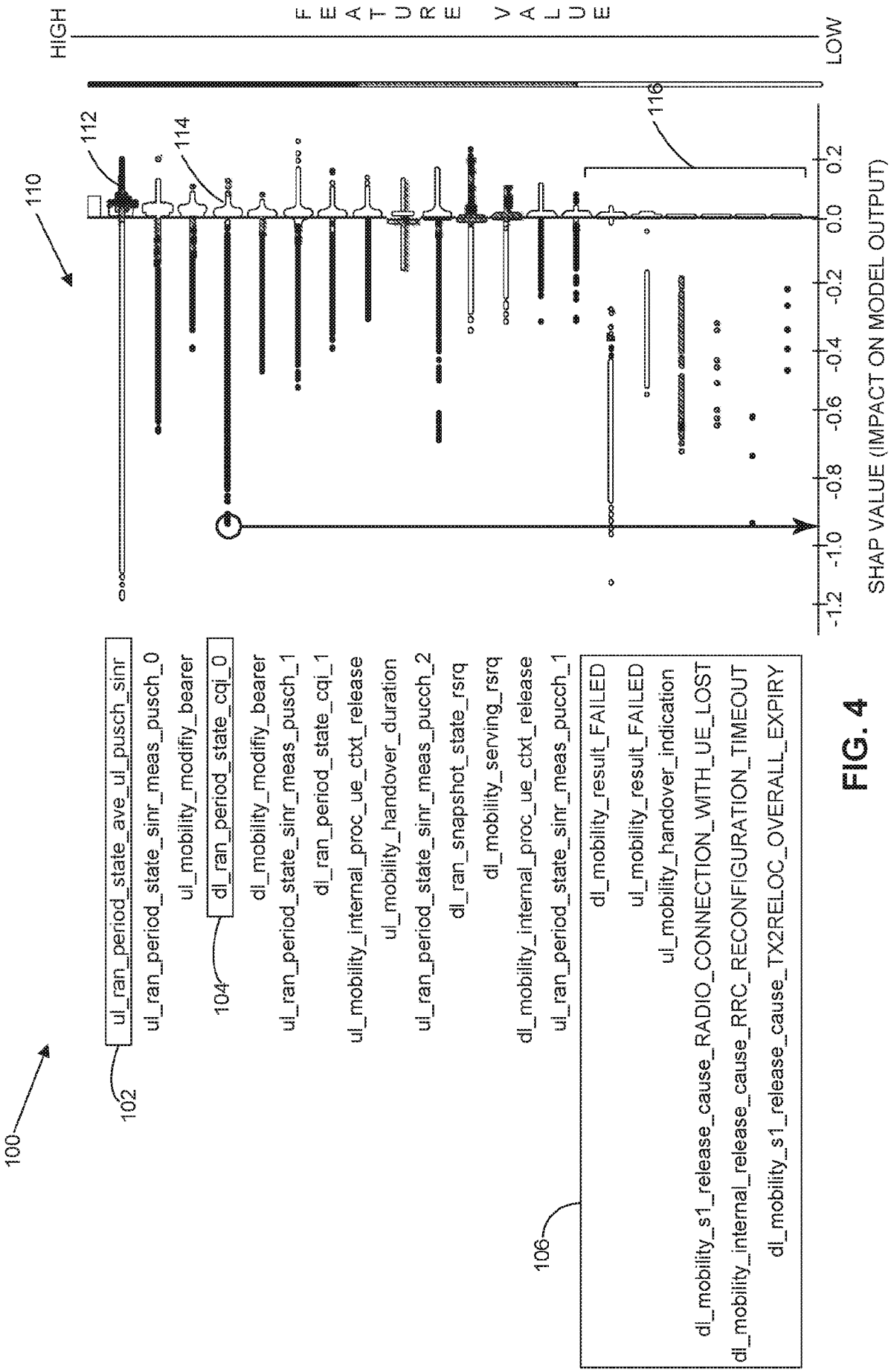
100
102
ul_ran_period_state_ave_ul_pusch_sinr
ul_ran_period_state_sinr_meas_pusch_0
ul_mobility_modifiy_bearer
104
dl_ran_period_state_cqi_0
dl_mobility_modifiy_bearer
ul_ran_period_state_sinr_meas_pusch_1
dl_ran_period_state_cqi_1
ul_mobility_internal_proc_ue_ctxt_release
ul_mobility_handover_duration
ul_ran_period_state_sinr_meas_pucch_2
dl_ran_snapshot_state_rsrq
dl_mobility_serving_rsrq
dl_mobility_internal_proc_ue_ctxt_release
ul_ran_period_state_sinr_meas_pucch_1
106
dl_mobility_result_FAILED
ul_mobility_result_FAILED
ul_mobility_handover_indication
dl_mobility_s1_release_cause_RADIO_CONNECTION_WITH_UE_LOST
dl_mobility_internal_release_cause_RRC_RECONFIGURATION_TIMEOUT
dl_mobility_s1_release_cause_TX2RELOC_OVERALL_EXPIRY
110
112
114
116
HIGH
LOW
FEATURE VALUE
-1.2 -1.0 -0.8 -0.6 -0.4 -0.2 0.0 0.2
SHAP VALUE (IMPACT ON MODEL OUTPUT)

FIG. 4

FEATURE NAME = DIRECTION__	BLOCKNAME__	EVENT NAME[EVENT PARAMETER [EVENT VALUE]
ul	rtp	Event name: occurrences of the specific event
dl	ran_period_state	Event parameter: parameter value (for numeric parameters)
	ran_snapshot_state	Event value: one hot encoding for categorical parameter values
	mobility	
	sip	

FIG. 5

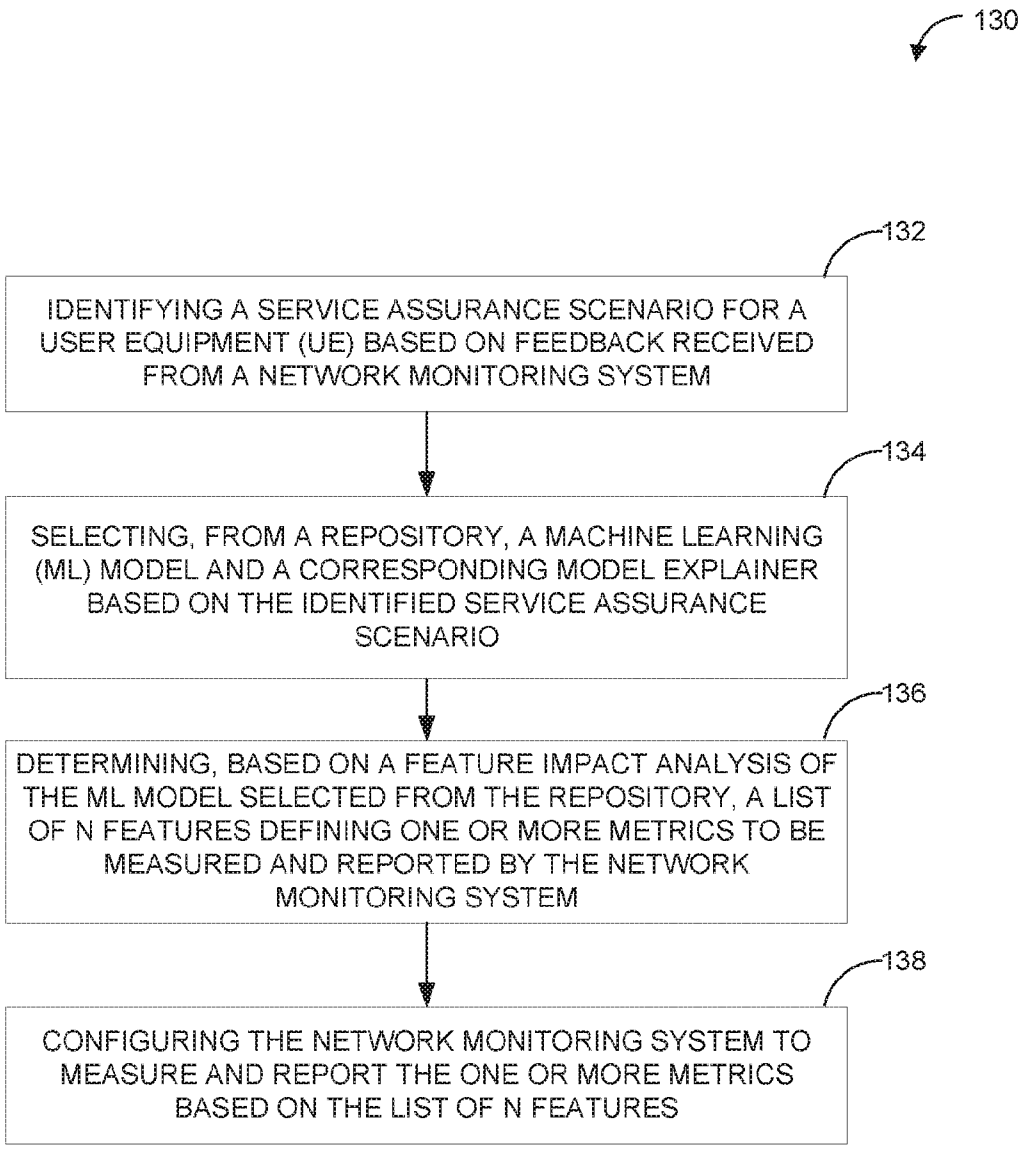
130
132
IDENTIFYING A SERVICE ASSURANCE SCENARIO FOR A USER EQUIPMENT (UE) BASED ON FEEDBACK RECEIVED FROM A NETWORK MONITORING SYSTEM
134
SELECTING, FROM A REPOSITORY, A MACHINE LEARNING (ML) MODEL AND A CORRESPONDING MODEL EXPLAINER BASED ON THE IDENTIFIED SERVICE ASSURANCE SCENARIO
136
DETERMINING, BASED ON A FEATURE IMPACT ANALYSIS OF THE ML MODEL SELECTED FROM THE REPOSITORY, A LIST OF N FEATURES DEFINING ONE OR MORE METRICS TO BE MEASURED AND REPORTED BY THE NETWORK MONITORING SYSTEM
138
CONFIGURING THE NETWORK MONITORING SYSTEM TO MEASURE AND REPORT THE ONE OR MORE METRICS BASED ON THE LIST OF N FEATURES

FIG. 6

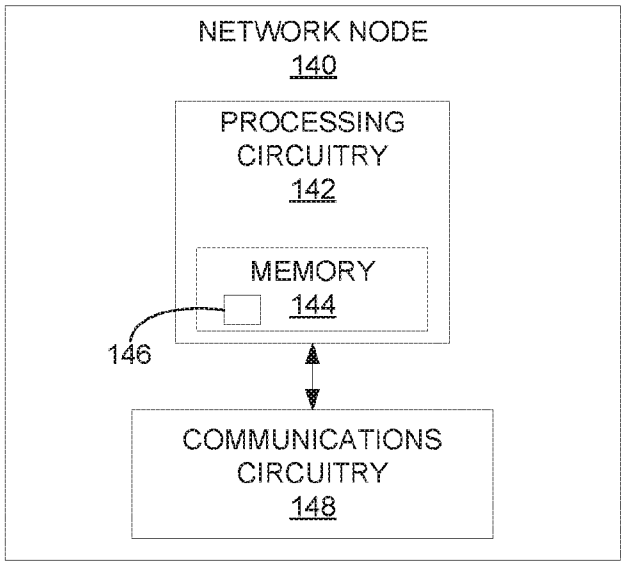
NETWORK NODE
140
PROCESSING CIRCUITRY
142
MEMORY
144
146
COMMUNICATIONS CIRCUITRY
148

FIG. 7

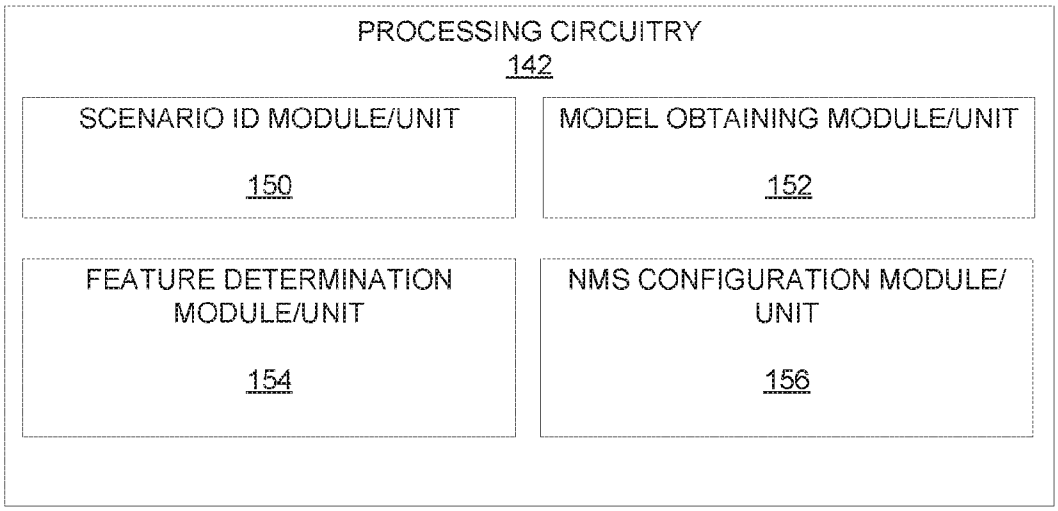
PROCESSING CIRCUITRY
142
SCENARIO ID MODULE/UNIT
150
MODEL OBTAINING MODULE/UNIT
152
FEATURE DETERMINATION MODULE/UNIT
154
NMS CONFIGURATION MODULE/ UNIT
156

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2020/058346

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/24 H04W24/10 G06F11/30
ADD. H04L12/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L H04W G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/270126 A1 (TAPIA PABLO [ES]) 20 September 2018 (2018-09-20) paragraphs [0003], [0004], [0011] - [0015], [0021], [0025], [0049], [0051], [0055], [0065], [0070], [0073], [0081] figure 1	1-21
X	US 2019/239158 A1 (WULFF SHARON SHOSHANA [CH] ET AL) 1 August 2019 (2019-08-01) paragraphs [0012], [0037], [0038], [0041], [0048] - [0052], [0056], [0096] figure 3	1,17, 19-21
A	EP 3 518 467 A1 (CISCO TECH INC [US]) 31 July 2019 (2019-07-31) paragraphs [0025] - [0027], [0035], [0042]	1-21

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 November 2020	12/11/2020

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Ramenzoni, Stefano

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2020/058346

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/365581 A1 (VASSEUR JEAN-PHILIPPE [FR] ET AL) 20 December 2018 (2018-12-20) paragraphs [0040], [0042], [0049] - [0054], [0095], [9658]	1-21
A	US 2018/352025 A1 (ANYA OBINNA B [US] ET AL) 6 December 2018 (2018-12-06) paragraphs [0020], [0021], [0054] - [0059]	1-21
A	ZHOU ZHU ET AL: "Applying Machine Learning to Service Assurance in Network Function Virtualization Environment", 2018 FIRST INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR INDUSTRIES (AI4I), IEEE, 26 September 2018 (2018-09-26), pages 112-115, XP033529881, DOI: 10.1109/AI4I.2018.8665716 [retrieved on 2019-03-11] the whole document	1-21

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/IB2020/058346

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018270126 A1	20-09-2018	US 2018270126 A1	20-09-2018
		WO 2018169894 A1	20-09-2018
US 2019239158 A1	01-08-2019	NONE	
EP 3518467 A1	31-07-2019	EP 3518467 A1	31-07-2019
		US 2019238443 A1	01-08-2019
US 2018365581 A1	20-12-2018	NONE	
US 2018352025 A1	06-12-2018	NONE	