SUPPORT FOR MACHINE LEARNING ENABLED ANALYTICS

TECHNICAL FIELD

[0001] The present disclosure relates to wireless communications, and more specifically to data analytics services.

BACKGROUND

[0002] A wireless communications system may include one or multiple network communication devices, such as base stations, which may support wireless communications for one or multiple user communication devices, which may be otherwise known as user equipment (UE), or other suitable terminology. The wireless communications system may support wireless communications with one or multiple user communication devices by utilizing resources of the wireless communication system (e.g., time resources (e.g., symbols, slots, sub-frames, frames, or the like) or frequency resources (e.g., subcarriers, carriers, or the like)). Additionally, the wireless communications system may support wireless communications across various radio access technologies including third generation (3G) radio access technology, fourth generation (4G) radio access technology, fifth generation (5G) radio access technology, among other suitable radio access technologies beyond 5G (e.g., sixth generation (6G)).

[0003] Additionally, the system comprises one or more wireless communication platforms at the edge and /or cloud data networks. Such platforms may include application and/or edge enablement services.

SUMMARY

[0004] An article “a” before an element is unrestricted and understood to refer to “at least one” of those elements or “one or more” of those elements. The terms “a,” “at least one,” “one or more,” and “at least one of one or more” may be interchangeable. As used herein, including in the claims, “or” as used in a list of items (e.g., a list of items prefaced by a phrase such as “at least one of” or “one or more of” or “one or both of”) indicates an inclusive list such that, for example, a list of at least one of A, B, or C means A or B or C or AB or AC or BC or ABC (i.e., A and B and C). Also, as used herein, the phrase “based on” shall not be

construed as a reference to a closed set of conditions. For example, an example step that is described as “based on condition A” may be based on both a condition A and a condition B without departing from the scope of the present disclosure. In other words, as used herein, the phrase “based on” shall be construed in the same manner as the phrase “based at least in part on. Further, as used herein, including in the claims, a “set” may include one or more elements.

[0005] Some implementations of the method and apparatuses described herein may include an analytics enabler entity for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, the analytics enabler entity comprising, at least one memory; and at least one processor coupled with the at least one memory and configured to cause the analytics enabler entity to receive a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task, identify a requirement for providing machine learning-enabled analytics for the application layer data analytics task, and upon identifying the requirement for providing machine learning-enabled analytics, obtain machine learning model information from a machine learning model repository and select at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter and at least one application entity parameter.

[0006] In an embodiment, the analytics enabler entity is further configured to send the machine learning model information to the at least one selected application entity, configure the at least one selected application entity using the machine learning model information, and to receive derived application data analytics output based on trained and/or inferred machine learning model data from the at least one selected application entity.

[0007] In an embodiment, the analytics enabler entity is further configured to process the received derived application data analytics output based on at least one of the at least one analytics parameter of the service requirement, wherein the processing comprises one or more of aggregating or filtering data from the received derived analytics output and send the processed derived analytics output to the analytics consumer.

[0008] In an embodiment, the machine learning model processing comprises at least one ML model lifecycle operation, wherein the model lifecycle operation is one of: a machine learning model inference or machine learning training operation.

[0009] In an embodiment, the analytics enabler entity is further configured to identify at least one candidate entity, wherein, optionally, at least one of the at least one candidate entities is an application enablement server and/or client.

[0010] In an embodiment, the analytics enabler entity is further configured to receive in the service requirement an analytics parameter comprising an indication of consumer type, wherein, optionally, the consumer type is one of: a Vertical Application Layer server, an Edge Application Servers, an Edge Enabler Server Application Client, an Edge Enabler Client, a Vertical Application Layer client, a Network Function, an Application Function, a management function or server, or an external application, and select the at least one application entity based at least partly on the consumer type.

[0011] In an embodiment, the analytics enabler entity is further configured to identify the requirement for machine learning-enabled analytics for the application layer data analytics task by either: receiving from the analytics consumer an indication, or determining based on the at least one analytics parameter, that machine learning-enabled analytics is required.

[0012] In an embodiment, the at least one application entity parameter is one of: a proximity to data producers, a signaling cost, a latency, a data accessibility, a trust level, a reliability, an energy usage, a cost, or, a location of and applications supported in an edge network.

[0013] In an embodiment, the analytics enabler entity is further configured to select at least one application entity for performing the machine learning model processing, wherein selecting at least one application entity comprises selecting one or both of an application layer machine model training functionality or application layer machine learning model inference functionality.

[0014] Some implementations of the method and apparatuses described herein may further include a method for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, the method

comprising receiving a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task, identifying a requirement for providing machine learning-enabled analytics for the application layer data analytics task and upon identifying the requirement for providing machine learning-enabled analytics, obtaining machine learning model information from an application layer machine learning model repository and selecting at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter and at least one application entity parameter.

[0015] In an embodiment, the method may further comprise configuring the at least one selected application entity using the machine learning model information, and receiving derived analytics output based on trained and/or inferred machine learning model data from the at least one selected application entity.

[0016] In an embodiment, the method may further comprise processing the received derived analytics output, in particular by aggregating or filtering the data based on the vertical service requirement and sending the processed derived analytics output to the analytics consumer.

[0017] In an embodiment, the machine learning model processing comprises at least one ML model lifecycle operation, wherein the model lifecycle operation is one of: a machine learning model inference or machine learning training operation.

[0018] In an embodiment, the method may further comprise identifying at least one candidate entity, wherein, optionally, at least one of the at least one candidate entities is an application enablement server and/or client.

[0019] In an embodiment, the method may further comprise receiving in the service requirement an analytics parameter comprising an indication of consumer type, wherein, optionally, the consumer type is one of: a Vertical Application Layer server, an Edge Application Servers, an Edge Enabler Server Application Client, an Edge Enabler Client, a Vertical Application Layer client, a Network Function, an Application Function, a

management function or server, or an external application and selecting the at least one application entity based at least partly on the consumer type.

[0020] In an embodiment, the method may further comprise identifying the requirement for machine learning-enabled analytics for the application layer data analytics task comprises either receiving from the analytics consumer an indication, or determining based on the at least one analytics parameter, that machine learning-enabled analytics is required.

[0021] In an embodiment, the method may further comprise at least one application entity parameter is one of: a proximity to data producers, a signaling cost, a latency, a data accessibility, a trust level, a reliability, an energy usage, a cost, or, a location of and applications supported in an edge network.

[0022] In an embodiment, the method may further comprise selecting at least one application entity comprises selecting one or both of an application layer machine model training functionality or application layer machine learning model inference functionality.

[0023] Some implementations of the method and apparatuses described herein may further include a processor for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, comprising at least one controller coupled with at least one memory and configured to cause the processor to obtain a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task, identify a requirement for machine learning-enabled analytics for the application layer data analytics task, and upon identifying the requirement for machine learning-enabled analytics obtain machine learning model information from an application layer machine learning model repository, and select at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter, and at least one application entity parameters, and output an indication of the at least one application entity.

[0024] In an embodiment, the processor is further configured to configure the at least one selected application entity using the machine learning model information, and receive

derived analytics output based on trained and/or inferred machine learning model data from the at least one selected entity.

[0025] In an embodiment, the processor is further configured to process the received derived analytics output based on at least one of the at least one analytics parameter of the service requirement, wherein the processing comprises one or more of aggregating or filtering data from the received derived analytics; and output the processed derived analytics to the analytics consumer.

[0026] A network entity comprising an application entity for implementing machine learning model processing for application layer data analytics, comprising at least one memory and at least one processor coupled with the at least one memory and configured to cause the network entity to receive from an analytics enabler entity machine learning model information, to be configured, based on the machine learning model information to implement one or more of: a machine learning training task, a machine learning inference task or a data collection task, and to send derived analytics output and/or collected data to the analytics enabler entity.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] Figure 1 illustrates an example of a wireless communications system in accordance with aspects of the present disclosure.

[0028] Figure 2 is a schematic diagram illustrating a high-level architecture for ADAE services.

[0029] Figure 3 illustrates an example of coordinated deployment in accordance with aspects of the present disclosure.

[0030] Figure 4 illustrates an example of machine learning lifecycle building blocks in accordance with aspects of the present disclosure.

[0031] Figure 5 illustrates the high level steps of a process for configuration of machine learning analytics according to an embodiment of the present disclosure.

[0032] Figure 6 illustrates an example of a signaling and data flow diagram in accordance with aspects of the present disclosure.

[0033] Figure 7 illustrates an example of a signaling and data flow diagram in accordance with aspects of the present disclosure.

[0034] Figure 8 illustrates an example of a user equipment (UE) in accordance with aspects of the present disclosure.

[0035] Figure 9 illustrates an example of a processor in accordance with aspects of the present disclosure.

[0036] Figure 10 illustrates an example of a network equipment (NE) in accordance with aspects of the present disclosure.

[0037] Figure 11 illustrate a flowcharts of method performed by a NE in accordance with aspects of the present disclosure.

[0038] Figure 12 illustrate a flowcharts of method performed by a NE in accordance with an embodiment.

DETAILED DESCRIPTION

[0039] In 3GPP, data analytics services are provided by the Network Data Analytics Function (NWDAF) in 3GPP standard TS 23.288 and aim to support network data analytics services in 5G Core network. Application Data Analytics Enablement Servers may support the utilization of machine learning enabled analytics for predicting the application layer performance and edge load. However, there is yet no mechanism as to how the artificial intelligence/machine learning lifecycle aspects can supported in coordinated ADAES deployments. These aspect include, where and how the ML models are trained for deriving analytics, for example, internally or externally to ADAE layer, whether machine model inference is needed for enhancing analytics using real time application data, and what the impact is on signaling and capabilities required in SEAL/ADAE layer.

[0040] The present invention provides an apparatus and method for selecting, one or more application entities for implementing machine learning model processing for application layer data analytics. The selection may be based on a number of criteria for both the selected entities and the consumer type. This has the advantage of providing an efficient and optimal selection of entities for which to implement the machine learning data analytics.

[0041] Aspects of the present disclosure are described in the context of a wireless communications system.

[0042] Figure 1 illustrates an example of a wireless communications system 100 in accordance with aspects of the present disclosure. The wireless communications system 100 may comprise one or more wireless communication platforms. The wireless communications system 100 may include one or more NE 102, one or more UE 104, and a core network (CN) 106. The CN 106 may be implemented by way of at least one NE 102. The NE 102 may comprise a base station, for example a gNB. The one or more NE 102 may embody at least one of an application enablement layer, an edge enablement layer, a service or a functionality. Such a layer is part of the system and may be implemented with the CN 106.

[0043] The wireless communications system 100 may support various radio access technologies. In some implementations, the wireless communications system 100 may be a 4G network, such as an LTE network or an LTE-Advanced (LTE-A) network. In some other implementations, the wireless communications system 100 may be a NR network, such as a 5G network, a 5G-Advanced (5G-A) network, or a 5G ultrawideband (5G-UWB) network. In other implementations, the wireless communications system 100 may be a combination of a 4G network and a 5G network, or other suitable radio access technology including Institute of Electrical and Electronics Engineers (IEEE) 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20. The wireless communications system 100 may support radio access technologies beyond 5G, for example, 6G. Additionally, the wireless communications system 100 may support technologies, such as time division multiple access (TDMA), frequency division multiple access (FDMA), or code division multiple access (CDMA), etc.

[0044] The one or more NE 102 may be dispersed throughout a geographic region to form the wireless communications system 100. One or more of the NE 102 described herein may be or include or may be referred to as a network node, a base station, a network element, a network function, a network entity, a radio access network (RAN), a NodeB, an eNodeB (eNB), a next-generation NodeB (gNB), or other suitable terminology. An NE 102 and a UE 104 may communicate via a communication link, which may be a wireless or wired connection. For example, an NE 102 and a UE 104 may perform wireless communication (e.g., receive signaling, transmit signaling) over a Uu interface.

[0045] An NE 102 may provide a geographic coverage area for which the NE 102 may support services for one or more UEs 104 within the geographic coverage area. For example, an NE 102 and a UE 104 may support wireless communication of signals related to services (e.g., voice, video, packet data, messaging, broadcast, etc.) according to one or multiple radio access technologies. In some implementations, an NE 102 may be moveable, for example, a satellite associated with a non-terrestrial network (NTN). In some implementations, different geographic coverage areas 112 associated with the same or different radio access technologies may overlap, but the different geographic coverage areas may be associated with different NE 102.

[0046] The one or more UE 104 may be dispersed throughout a geographic region of the wireless communications system 100. A UE 104 may include or may be referred to as a remote unit, a mobile device, a wireless device, a remote device, a subscriber device, a transmitter device, a receiver device, or some other suitable terminology. In some implementations, the UE 104 may be referred to as a unit, a station, a terminal, or a client, among other examples. Additionally, or alternatively, the UE 104 may be referred to as an Internet-of-Things (IoT) device, an Internet-of-Everything (IoE) device, or machine-type communication (MTC) device, among other examples.

[0047] A UE 104 may be able to support wireless communication directly with other UEs 104 over a communication link. For example, a UE 104 may support wireless communication directly with another UE 104 over a device-to-device (D2D) communication link. In some implementations, such as vehicle-to-vehicle (V2V) deployments, vehicle-to-everything (V2X) deployments, or cellular-V2X deployments, the communication link 114 may be referred to as a sidelink. For example, a UE 104 may support wireless communication directly with another UE 104 over a PC5 interface.

[0048] An NE 102 may support communications with the CN 106, or with another NE 102, or both. For example, an NE 102 may interface with other NE 102 or the CN 106 through one or more backhaul links (e.g., S1, N2, N3, or network interface). In some implementations, the NE 102 may communicate with each other directly. In some other implementations, the NE 102 may communicate with each other indirectly (e.g., via the CN 106). In some implementations, one or more NE 102 may include subcomponents, such as an access

network entity, which may be an example of an access node controller (ANC). An ANC may communicate with the one or more UEs 104 through one or more other access network transmission entities, which may be referred to as a radio heads, smart radio heads, or transmission-reception points (TRPs).

[0049] The CN 106 may support user authentication, access authorization, tracking, connectivity, and other access, routing, or mobility functions. The CN 106 may be an evolved packet core (EPC), or a 5G core (5GC), which may include a control plane entity that manages access and mobility (e.g., a mobility management entity (MME), an access and mobility management functions (AMF)) and a user plane entity that routes packets or interconnects to external networks (e.g., a serving gateway (S-GW), a Packet Data Network (PDN) gateway (P-GW), or a user plane function (UPF)). In some implementations, the control plane entity may manage non-access stratum (NAS) functions, such as mobility, authentication, and bearer management (e.g., data bearers, signal bearers, etc.) for the one or more UEs 104 served by the one or more NE 102 associated with the CN 106.

[0050] The CN 106 may communicate with a packet data network over one or more backhaul links (e.g., via an S1, N2, N3, or another network interface). The packet data network may include an application server. In some implementations, one or more UEs 104 may communicate with the application server. A UE 104 may establish a session (e.g., a protocol data unit (PDU) session, or the like) with the CN 106 via an NE 102. The CN 106 may route traffic (e.g., control information, data, and the like) between the UE 104 and the application server using the established session (e.g., the established PDU session). The PDU session may be an example of a logical connection between the UE 104 and the CN 106 (e.g., one or more network functions of the CN 106).

[0051] In the wireless communications system 100, the NEs 102 and the UEs 104 may use resources of the wireless communications system 100 (e.g., time resources (e.g., symbols, slots, subframes, frames, or the like) or frequency resources (e.g., subcarriers, carriers)) to perform various operations (e.g., wireless communications). In some implementations, the NEs 102 and the UEs 104 may support different resource structures. For example, the NEs 102 and the UEs 104 may support different frame structures. In some implementations, such as in 4G, the NEs 102 and the UEs 104 may support a single frame structure. In some other

implementations, such as in 5G and among other suitable radio access technologies, the NEs 102 and the UEs 104 may support various frame structures (i.e., multiple frame structures). The NEs 102 and the UEs 104 may support various frame structures based on one or more numerologies.

[0052] One or more numerologies may be supported in the wireless communications system 100, and a numerology may include a subcarrier spacing and a cyclic prefix. A first numerology (e.g., μ=0) may be associated with a first subcarrier spacing (e.g., 15 kHz) and a normal cyclic prefix. In some implementations, the first numerology (e.g., μ=0) associated with the first subcarrier spacing (e.g., 15 kHz) may utilize one slot per subframe. A second numerology (e.g., μ=1) may be associated with a second subcarrier spacing (e.g., 30 kHz) and a normal cyclic prefix. A third numerology (e.g., μ=2) may be associated with a third subcarrier spacing (e.g., 60 kHz) and a normal cyclic prefix or an extended cyclic prefix. A fourth numerology (e.g., μ=3) may be associated with a fourth subcarrier spacing (e.g., 120 kHz) and a normal cyclic prefix. A fifth numerology (e.g., μ=4) may be associated with a fifth subcarrier spacing (e.g., 240 kHz) and a normal cyclic prefix.

[0053] A time interval of a resource (e.g., a communication resource) may be organized according to frames (also referred to as radio frames). Each frame may have a duration, for example, a 10 millisecond (ms) duration. In some implementations, each frame may include multiple subframes. For example, each frame may include 10 subframes, and each subframe may have a duration, for example, a 1 ms duration. In some implementations, each frame may have the same duration. In some implementations, each subframe of a frame may have the same duration.

[0054] Additionally or alternatively, a time interval of a resource (e.g., a communication resource) may be organized according to slots. For example, a subframe may include a number (e.g., quantity) of slots. The number of slots in each subframe may also depend on the one or more numerologies supported in the wireless communications system 100. For instance, the first, second, third, fourth, and fifth numerologies (i.e., μ=0, μ=1, μ=2, μ=3, μ=4) associated with respective subcarrier spacings of 15 kHz, 30 kHz, 60 kHz, 120 kHz, and 240 kHz may utilize a single slot per subframe, two slots per subframe, four slots per subframe, eight slots per subframe, and 16 slots per subframe, respectively. Each slot may

include a number (e.g., quantity) of symbols (e.g., OFDM symbols). In some implementations, the number (e.g., quantity) of slots for a subframe may depend on a numerology. For a normal cyclic prefix, a slot may include 14 symbols. For an extended cyclic prefix (e.g., applicable for 60 kHz subcarrier spacing), a slot may include 12 symbols. The relationship between the number of symbols per slot, the number of slots per subframe, and the number of slots per frame for a normal cyclic prefix and an extended cyclic prefix may depend on a numerology. It should be understood that reference to a first numerology (e.g., μ =0) associated with a first subcarrier spacing (e.g., 15 kHz) may be used interchangeably between subframes and slots.

[0055] In the wireless communications system 100, an electromagnetic (EM) spectrum may be split, based on frequency or wavelength, into various classes, frequency bands, frequency channels, etc. By way of example, the wireless communications system 100 may support one or multiple operating frequency bands, such as frequency range designations FR1 (410 MHz – 7.125 GHz), FR2 (24.25 GHz – 52.6 GHz), FR3 (7.125 GHz – 24.25 GHz), FR4 (52.6 GHz – 114.25 GHz), FR4a or FR4-1 (52.6 GHz – 71 GHz), and FR5 (114.25 GHz – 300 GHz). In some implementations, the NEs 102 and the UEs 104 may perform wireless communications over one or more of the operating frequency bands. In some implementations, FR1 may be used by the NEs 102 and the UEs 104, among other equipment or devices for cellular communications traffic (e.g., control information, data). In some implementations, FR2 may be used by the NEs 102 and the UEs 104, among other equipment or devices for short-range, high data rate capabilities.

[0056] FR1 may be associated with one or multiple numerologies (e.g., at least three numerologies). For example, FR1 may be associated with a first numerology (e.g., μ=0), which includes 15 kHz subcarrier spacing; a second numerology (e.g., μ=1), which includes 30 kHz subcarrier spacing; and a third numerology (e.g., μ =2), which includes 60 kHz subcarrier spacing. FR2 may be associated with one or multiple numerologies (e.g., at least 2 numerologies). For example, FR2 may be associated with a third numerology (e.g., μ=2), which includes 60 kHz subcarrier spacing; and a fourth numerology (e.g., μ =3), which includes 120 kHz subcarrier spacing.

[0057] In the Third Generation Partnership Project (3GPP), data analytics services are provided by the Network Data Analytics Function (NWDAF), as standardized in 3GPP standard TS 23.288, and aim to support network data analytics services in Fifth Generation (5G) Core Networks. Such analytics can collect data from other network functions (NFs), application functions (AF) and/or Operation and Maintenance (OAM) and the analytics results can be provided to the third party application functions to provide statistics and predictions related to slice Load level, observed Service experience, NF Load, Network Performance, user equipment (UE) related analytics (mobility, communication), User data congestion, Quality of Service (QoS) sustainability, data network, DN, performance, etc. Furthermore, in 3GPP standard SA5 (TS 28.104), Management Data Analytics Service (MDAS) provides data analytics for the network. MDAS can be deployed at different levels, for example, network element level, e.g., gNB, at domain level (e.g., Radio Access Network (RAN), Core Network (CN), network slice subnet) or in a centralized manner (e.g. in a Public Land Mobile Network (PLMN) level). The objective of MDAS is to provide root cause analysis on complex problems and optimize the network resource allocation (at network / domain level, and at slice / slice subnet level).

[0058] An additional analytics function in 3GPP is discussed in 3GPP standard SA6 (TS 23.436) where an Application Data Analytics Enablement Service (ADAES) is defined for performing application layer and edge / cloud analytics outside 3GPP domain. ADAES can be seen as an AF which has analytics capability, and also has an interface to the UE side (to ADAE client) as well as to OAM. Figure 2 is a schematic diagram illustrating a high- level architecture for ADAE services, 200. The architecture comprises the client side, 201, comprising a Vertical Application Layer, VAL, client, 202, and an Application data analytics enablement client (ADAE-C), 203, and a server side, 204, comprising one or more of a VAL server, 205, and an Application data analytics enablement server (ADAE-S), 206, and a 3GPP network, 207.

[0059] In some embodiments, application layer data analytics refers to the analytics defined in 3GPP standard TS 23.436, which is herein incorporated by reference and is hereafter referred to as TS 23.436. In other embodiments, the term “application layer data analytics” may be more generic and refer to analytics which provide statistics, predictions or prescriptions for at least one application parameter which is used for an application service

or session, where the application service or session is communicated via the wireless communications system. Examples of such parameters are the application service performance indicators such as Round Trip Time (RTT), throughput, latency, jitter, application Quality of Service (QoS) or Quality of Experience (QoE); application or UE mobility related parameters, server availability and reliability related parameters, edge load or performance parameters.

[0060] Referring to Figure 2, the VAL server(s), 205, may communicate with the ADAES, 206, over the ADAE-S reference point, 208. The ADAES, acting as an application function, AF, may communicate with the 5G Core Network functions, 207. This communication may be with one or more of the Network Exposure Function, NEF, the User Plane Function, UPF, or the Operations and Maintenance, OAM, which are within the 3GPP network, but not shown in Figure 2. Communication may be over the N33 reference point, 209, when communicating with the NEF, over the N6 reference point, 210, when communicating with the UPF, and over the ADAE-OAM interface when communicating with the OAM.

[0061] ADAES supports analytics (e.g., VAL server performance, edge load analytics, location analytics etc.) and it is also possible that such analytics methods may be machine learning enabled. There are different deployments and business models for ADAES supported by TS 23.436. In examples of different deployments, the ADAES can be within the Public Land Mobile Network, PLMN, at an Edge Computing Service Provider, ECSP, or in a vertical domain. Regarding the deployment models, there are three possible scenarios, namely centralized, distributed, coordinated.

[0062] Figure 3 illustrates an example of coordinated deployment, 300, in accordance with aspects of the present disclosure. As stated in TS 23.436, multiple ADAESs can be located at different Edge Data Networks (EDNs) or data networks (DNs) and can be deployed by the same ADAE provider. Such coordinated deployments allow the local – global analytics derivation, which may be needed for improving the analytics confidence level. The centrally deployed ADAES can also act as an ADAE analytics aggregator entity, which can configure an edge deployed ADAES to derive analytics on different sub-area

[0063] In the example of Figure 3, there are two EDNs, EDN#1, 301 and EDN#2, 302, and a centralized DN, 303. ADAE servers, ADAE#1.1, 304, ADAE#1.2, 305, and ADAE#1,

306 are located in the respective networks. The EDNs each have an Edge Application Server (EAS), 307, 308 and an Edge Enabler Server (EES), 309, 310, and the centralized network has a VAL server, 311.

[0064] One example is the use of analytics, which may be implemented in the example network of Figure 3, is the use of the EDN#1 or EDN#2 load values which can be used to assist the prediction of the performance of the VAL server, 311, in the centrally located network. Such deployment is also applicable for machine learning based analytics methods, such as supervised learning, where the centrally located ADAES acts as a machine learning model training entity, and the edge located ADAESs can act as machine learning model inference entities (using edge data to improve the prediction accuracy).

[0065] The statistics/predictions that the edge deployed ADAES correspond to the ADAES service areas, 316, 317, which is equivalent to the EES/EAS service areas. The central ADAE server covers all PLMN area, 318, and is used to coordinate or jointly perform analytics with the distributed ADAES. Such analytics services can be provided to consumers at the central DN, like the VAL servers or SEAL services or even at the PLMN side (e.g. NWDAF consuming service experience analytics).

ADAES may support the utilization of machine learning enabled analytics for predicting the application layer performance and edge load. However, as discussed above, there is yet no mechanism as to how the artificial intelligence/machine learning lifecycle aspects can supported in coordinated ADAES deployments. These aspect include, where and how the machine learning models are trained for deriving analytics, for example, internally or externally to ADAE layer, whether machine model inference is needed for enhancing analytics using real time application data, and what the impact is on signaling and capabilities required in SEAL/ADAE layer.

[0066] Figure 4 illustrates an example of machine learning lifecycle building blocks, 400, in accordance with aspects of the present disclosure. These building blocks comprise data collection, 401, data Preparation, 402, machine learning model selection, 403, machine learning model deployment, 404, machine learning model training, 405 and inference, 406, and the prediction/prescription based on the training/inference, 407. All these building blocks may be in one place, for example, in centralized deployments, like a data center, or alternatively can be distributed over edge/cloud and network domains, i.e. core network,

management system, Radio Access Network, (RAN)). Furthermore, different variants may be possible based on the objectives and requirements for different vertical deployment. Referring to Figure 4, three tailored options are illustration, for Vehicle to Anything (V2X), 408, Future Factories (FF), 409, and Unified Access Services (UAS), 410. The interactions of the building blocks of the machine learning lifecycle with the Application Specific Layer, 411, the Application Enabler Layer, 412, the Edge Enabler Layer, 413, and the NWDAF/MDAS, 414, are illustrated.

[0067] In an example of the use of ML analytics, namely for automotive/vehicle to anything (V2X) verticals, 408, different ML algorithms and ML model types may be used to meet more critical traffic efficiency requirements. In another example, for gaming verticals, the ML model lifecycle may be different for ensuring best Quality of Experience (QoE) for the multiplayer games. For the above reasons, it is important to configure:

(i) The AI/ML parameters based on the service requirements and analytics service (e.g. ML model type and algorithms to be used).

(ii) The AI/ML lifecycle operations placement to support different service requirements and network deployments.

(iii) Whether and how to expose some AI/ML lifecycle operations to external entities or entities within 5GS/UEs.

(iv) How to multiplex processing of AI/ML lifecycle operations of different verticals to minimize signaling and complexity load while ensuring meeting all vertical customers' requirements.

[0068] In order to efficiently implement AI/ML analytics as described above, there is provided, in embodiments, an analytics enabler entity which has the role of selecting one or more application entities for implementing machine learning model processing for application layer data analytics. This is performed in response to the receipt of a service requirement for an application layer data analytics task from an analytics consumer. The analytics enabler entity identifies a requirement for providing machine learning-enabled analytics, whereupon it obtains machine learning model information from a machine learning model repository and selects at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one

analytics parameter, the obtained machine learning model information, and at least one application entity parameter.

[0069] The analytics enabler entity may be an Application Data Analytics Enabler (ADAE) server or client as specified in 3GPP standard TS 23.434, which is herein incorporated by reference, and is referred to hereafter as TS 23.434. Alternatively, The analytics enabler entity may be any other SEAL defined entity, as defined in TS 23.434 or EDGEAPP, as defined in 3GPP standard TS 23.558, which is herein incorporated by reference, and is referred to hereafter as TS 23.558. Alternatively the analytics enabler entity may be a new AI/ML support enablement entity which will be dedicated for ML model configuration and processing.

[0070] The analytics enabler entity is configured to provide support for the AI/ML lifecycle control aspects such as the control of the ML model processing entities and the associated parameters.

[0071] The analytics enabler entity in certain implementations can be deployed as part of Core Network and/or as part of OAM (as enhancement of MDAS or as new AI/ML support functionality.

[0072] An application entity has the role of implementing the machine learning processing in response to selection and configuration by the analytics enabler entity, and may be located in a network entity. In embodiments, an application entity may be an application enablement entity or an edge enablement entity as defined in 3GPP SA6, which is incorporated herein by reference, or an application function provided by the network operator or a trusted 3rd party to the network operator, as defined in 3GPP SA2 (3GPP TS 23.501), which is incorporated herein by reference. In embodiments, the Application Entity may comprise also the application enablement client at a user equipment, UE.

[0073] The service requirement comprises at least one analytics parameter for the application layer data analytics task, identifying a requirement for machine learning-enabled analytics for the application layer data analytics task.

[0074] In some embodiments, an application entity parameter is either a current status parameter based on measurements and/or historical data related to this parameter, but also can be an expected or predicted parameter, or an estimated parameter.

[0075] Figure 5 illustrates the high level steps, 500, of a process for configuration of machine learning analytics according to an embodiment of the present disclosure. The architecture comprises a VAL customer, 501, a central ADAES, 502, a machine learning training entity, 503, an (A-ADRF), 504, an Edge ADAES, 505, a central ADAES, 506, a 3GPP network, 507, and a plurality of ADAECs, 508, 509, 510, 511. Figure 5 further illustrates the instances of ML model inference, 512, and an analytics output, 513. In certain implementations the functionality illustrated at ADAES/ADAEC may be deployed as a new enablement capability at a potentially standalone AI/ML enabler server/client.

[0076] According to an embodiment of the disclosure, a method, which may be implemented in an architecture such as that of Figure 5, enables the selection of the ML model entities and configuration of model lifecycle parameters based on the analytics required and, in an embodiment, also on a customer type. This solution provides an enhancement of analytics services provided by the analytics enablement functionality, and, in embodiments, analytics related to VAL server or session performance, as well as analytics related to edge load, such as specified in 3GPP standard TS 23.436, which is incorporated by reference. It will be appreciated by the person skilled in the art that the architecture is an example only, and that many different arrangements of application entities and numbers of different types of application entities are possible and that the disclosure is not limited by any one arrangements. A network entity which is capable of performing machine learning processing may be referred to as an application entity.

[0077] The analytics process or service for which the application entities are selected may be referred to as an application layer data analytics task. The terms analytics service, analytics process or analytics task may be used interchangeably. An entity which requests an analytics task may be referred to as an analytics customer, and the request for the task may be referred to as a service request. The service request may comprise one or more of a vertical requirement and an application requirement. The service request may comprise one or more of: a Consumer ID, an Analytics ID, analytics filter information, an analytics type (prediction,

statistics), a VAL service ID, a target data producer profile criteria, a preferred confidence level, an Area of Interest, a time validity, a Destination EAS ID, a Destination EES ID, or a DNN/DNAI. A consumer can be a VAL server who is defined in TS 23.434. The VAL server can be a generalized server applicable to different verticals; hence in standards VAL servers can be V2X servers (as defined in TS 23.286, TS 23.287), UAS servers (as defined in TS 23.255), IIOT servers. A consumer can be also an Edge Application Servers (EAS) or Edge Enabler Server (EES), or Application Client or Edge Enabler Client (as defined in TS 23.558), VAL clients (as defined in TS 23.434), a NF (like NWDAF) or AF as defined in TS 23.501, a Management function/server or an external application (e.g. MEC server or MEC application).

[0078] In the example of Figure 5, The analytics customer is the VAL customer, 501, and the analytics enabler entity is the central ADAES, 502. However, the person skilled in the art would appreciate that the disclosure is not limited to this arrangement.

[0079] According to an embodiment, the following steps are performed. In a first step, the VAL server, which may be a vertical specific server or an edge/cloud server, subscribes, 514, to C-ADAES (anchor analytics server) for an analytics service (e.g., for Analytics ID = 'VAL server #1 perf analytics'). This request may also indicate the use of AI/ML enabled analytics, and optionally the ML model info or profile and ID(s). This request may be referred to as a service requirement.

[0080] In a second step, the C-ADAES identifies a need for ML-enabled analytics and in particular the application entities best suited to perform model training and inference, based on the deployment. C-ADAES also identifies the ML models info /profile (if not provided) for the given analytics ID/event ID. C-ADAES may also configure the ML model training parameters (algorithms, ...) based on the request.

[0081] In a third step, the C-ADAES requests, 515, from an ML model repository the ML model(s) based on the identification/profile or based on the analytics ID. Such repository can be the A-ADRF or an external registry or a registry at core network (e.g. NRF, ADRF). C-ADAES receives/fetches the ML models based on the request. Figure 5 illustrates an A-ADRF, but the person skilled in the art would recognize that this is not the only option and the disclosure is not limited to a particular source of model information.

[0082] In a fourth step, the C-ADAES determines which entity or entities, i.e. the application entities, will undertake the ML model training (denoted as "selected ML model training entity") based on the requirements / criteria for data collection and model processing. This is selected from a list of candidate ML model training entities which can be:

a. the C-ADAES,

b. other locally/edge deployed ADAESs,

c. one or more VAL UEs supporting ADAECs with capability to perform ML model training (capabilities related to processing power, energy constraints and latency limitations).

d. An external entity (e.g. ML forecast service by a third party provider).

[0083] The criteria for the selection of an application entity by the analytics enabler entity may be one or more of the following:

(i) The model training to be performed close to the data producers. This criterion is applicable to the case where the training is preferable to be locally performed at an entity which is close to the source of the data which are going to be used as inputs for the training. One example can be the case of performing ML model training for edge load analytics at the destination edge platform, and not at the central cloud, when the data sources are expected to be provided by the edge platform itself (e.g. real-time and historical load data from the edge platform such as number of connections per EAS/EES).

(ii) The signaling cost and latency to be minimized for collecting data from distributed data producers. In this criterion, a key decision factor for the selection is based on the cost and possible delays for collecting data from multiple sources. For example, if it is expected to perform ML model training at the central cloud (C-ADAES) or at the edge/VAL UE, and it is expected to need inputs for the training from the OAM, edge platforms and VAL UE, we need to evaluate what is the cost for selecting the ML model training entity to be either at the central cloud or at the edge or at the VAL UE, considering the needed time to collect data and whether this may impact the efficiency of the analytics service.

(iii) The trustfulness/rating or the reliability or availability of the candidate ML model training entity. For some entities which are considered to be non-trusted by the network operator (for example the application enabler server can be in trusted operator domain, however the VAL UE may be non-trusted) or their availability may change frequently or unpredictably (for example a VAL UE may go from connected to idle mode, or may go out of coverage for certain time or may have bad channel quality in certain areas), one criterion for the selection is the reliability/availability of these entities, and the selection may be based on this factor.

(iv) Energy constraints and processing power constraints for the ML model candidates. ML model training requires lots of processing power and some devices (mainly UEs) may have energy constraints and we may need to restrict the usage of these entities based on their capabilities as well as their power levels (e.g. battery status) if available. Such criterion can be either evaluated based on inputs from the VAL UEs or based on profiling the entities to serve as model training entities as “power constrained” and use only if other entities are not available or based on the analytics service

(v) Data accessibility given different stakeholders involved. ML model training requires lots of processing power and some devices (mainly UEs) may have energy constraints and we may need to restrict the usage of these entities based on their capabilities as well as their power levels (e.g. battery status) if available. Such criterion can be either evaluated based on inputs from the VAL UEs or based on profiling the entities to serve as model training entities as “power constrained” and use only if other entities are not available or based on the analytics service

(vi) Price/cost for the model training at an external entity and the trade-off between cost and benefit. this criterion is for the case when the training can be external based on agreements with IT companies which specialize in ML model training.

This may be common practice to offload the power consuming training, however this comes at a high cost since the external entity may charge a lot to accomplish that. So, one decision factor is whether to offload or not the training if the ADAE layer has the capability to do it internally given accessible data

(vii) Location of the edge network (e.g. based on DNAI) the application services supported by the edge network. In this case the selection of the ML model training entity can be based on the location of the edge network and its area coverage, so for edge analytics if the training is not possible at the destination edge network, then the training needs to be performed at another entity close to it (topologically or physically).

(viii) The application services supported by the edge network: this criterion is about selecting the ML model training entity based on the app services supported at the destination edge. For example, if the analytics is for EAS performance or load for certain EASs (e.g. gaming app services) or for V2X services or for group of different application services e.g. for a target vertical like a Stadium Provider, then the edge network which will undertake the ML model training needs to supports these services; or if there are more than one candidate entities to serve as ML model training entity to select the one supporting the more services (as possible criterion).

(ix) The requirements related to the analytics task (time granularity, confidence level requirement, analytics service area). such requirements are mainly performance and service requirements related to the analytics event for which the ML model training applies. For example, given the high confidence level or the granularity (how fast it is expected to provide outputs) then we may select different entity (closer to the edge or at a central location) since different selection may have impact on the efficiency of analytics.

[0084] The list of available entities to serve as ML model training or inference entities, i.e. the application entities, may be known at the analytics enabler entity when the application entities are instantiated and connect to analytics enabler entity. Alternatively, a list of candidate application entities may be pre-configured, or such information can be provided by the ML model repository, in response to the request for ML model information. The application entities may register their capabilities as well as the data availability to the ML model repository or at the analytics enabler entity at the time of initial registration/instantiation at the respective edge cloud.

[0085] If a combination of different criteria are to be used, then the determination may be based on pre-configured polices per analytics event and/or per consumer type. Such policies may be enforced as "weight" or "rating" factors which are used to allow the ADAES to select the optimum entity to perform the ML model processing. In certain embodiments, the determination of the selection may be based on solving a weighted utility optimization problem for finding the optimal entity(ies) to serve as ML training entity based on a given network utility target/objective (e.g. based on the pre-configured policies, like energy efficiency or delay minimum, or high confidence level), and constraints related to the use of certain entities (e.g. related to trust or data accessibility). Such problem may be solved by various approaches either heuristic/metaheuristic (e.g. based on graph theory, game theory) or approximate solutions.

[0086] In a fifth step of the embodiment of Figure 5, the C-ADAES subscribes, 516, to the selected ML model training entity, 503, to initiate the ML model training, and when requesting the subscription, it may also provide the ML model ID or the address to fetch directly from A-ADRF or may provide a local model parameters. A subscription response follows, 517, as positive or negative result. If the selected ML model training entity is external to ADAE layer, this may involve more interactions for setting service agreements and interactions related to charging which are out of scope of this invention.

[0087] In a sixth step the C-ADAES receives, 518, the trained ML model data from the selected ML training entity.

[0088] In a seventh step, if the C-ADAES doesn't perform itself the inference, it may also determine a list of entities (local ADAES and/or ADAECs) to perform ML model inference for supporting real-time analytics. This can be done for instance to increase the confidence level of analytics using real time or near-real time data, or if the analytics event requires edge/local real time data (e.g. for example edge load analytics).

[0089] In an embodiment, the Consumer may request the C-ADAES to just train the ML model and provide the analytics output (in case the consumer performs ML model itself). In this embodiment, the C-ADAES sends the analytics outputs based on the trained ML model to the consumer and the eighth, ninth and tenth steps, discussed below, may be omitted.

[0090] It is possible in certain embodiments, that such determination in the seventh step happens with or just after the fourth step. In this instance, the ML training entity sends directly the trained ML models to the ADAES/ADAEC which is expected to perform ML model inference to derive the online analytics (in that case the fifth step also includes the list of entities which will serve as ML model inference entities and their addresses).

[0091] In an eighth step, if the C-ADAES has delegated the model inference to the local/edge or UE entities, then C-ADAES distributes, 519, the ML model to the selected ML model inference entities (local ADAES or ADAEC).

[0092] In a ninth step, the C-ADAES or the one or more ADAE layer entities which are performing ML model inference derive, the analytics outputs, 513.

[0093] In a tenth step, if the C-ADAES has delegated the model inference to the local/edge or UE entities, then C-ADAES collects the analytics outputs, 520, and decides to further aggregate or process the outputs or just stores at the A-ADRF.

[0094] In a eleventh step, the C-ADAES sends, 521, the derived analytics outputs (processed or aggregated or original) to the VAL server / consumer.

[0095] Figure 6 illustrates an example of a signaling and data flow diagram, 600, in accordance with aspects of the present disclosure. This embodiment is directed at improving VAL performance analytics service using ML-enabled training and inference at the ADAE layer or via external entity. In this embodiment, the analytics enabler entity is a centralized ADAE server (C-ADAES), 601, which receives instructions from an analytics consumer, 602. There is provided an application model training entity, 603, a 5G network, 604, an EDN, 605, with a VAL server, 606, comprising an ADAE-C, 607, and an A-ADRF registry 608. The signaling and data transfers comprise the following steps:

[0096] Step 1: The consumer of the ADAES analytics service sends, 609 a VAL performance analytics subscription request to ADAES (including the ML model inference function). This request may be referred to as a service requirement. In embodiments, this request may include one or more of:

- Consumer ID, Analytics ID, Analytics filter information, Analytics type (prediction, statistics), VAL service ID, Target

-VAL UE ID(s), Target VAL server ID, Target data producer profile criteria, Preferred confidence level, Area of Interest, and Time validity.

- ML-enabled analytics requirement / flag, ML model profile/context to assist the selection of ML model by the ADAE layer

[0097] Step 2: The ADAES sends a subscription response, 610, as a positive or negative acknowledgement to the consumer of the analytics service.

[0098] Step 3: In an embodiment, the ADAES determines, 611, the need for ML-enabled analytics. In an alternative embodiment, a request for ML analytics may be sent by the analytics consumer. The ADAES then determines at least one entity to be considered for ML model training entity (internal to SEAL or external), and at least one entity for performing ML model inference for the analytics ID and/or consumer ID. In embodiments, the

determination can be based on one or more of the capabilities of the application entities (i.e. one or more of energy constraints, processing capabilities, availability/reliability of entity) as well as the Analytics Event expected output granularity (for example based on the expected analytics output and whether it is offline or online analytics), the preference to entities to serve as ML model training/inference entities which are close to the related data producers, interoperability constraints assuming multi-vendor models and data producers.

[0099] One example, in an embodiment, is the ML model training for predicting RTT deviation for a V2X application session in a target area and time. The ML model training can be performed offline at a central or edge ADAES (given the location of the V2X server), whereas the ML model inference may be selected to be performed at a V2X group leading UE or at a V2X-UE serving as Road Side Unit (RSU) or at a selected set of V2X UEs which are selected based on their capabilities.

[0100] Step 4a: The ADAES (central or anchor) sends a subscribe request, 612, to the ML model registry (this can be, in an embodiment, an Application layer – Analytics Data Repository Function (A-ADRF)) to receive the one or more ML model identities/information for the Analytics ID and/or the Consumer type (e.g. V2X server #1). The ML model registry is assumed to have a pre-configured mapping between Analytics ID and ML model(s) or this can be determined and provided by the ADAES or in step 1 by the analytics customer.

[0101] Step 4b: The ADAES (central or anchor) receives a subscribe response, 613 which includes the ML model information and the mapping to the analytics ID / consumer type, the ML model file address for the initial model, the permissions for updating the ML model (related to who is the consumer), the ML model vendor ID, list of ML model training entities which are allowed/preferred to train the model, charging model for using the ML model.

[0102] Step 5: The ADAES (central or anchor) subscribes, 614, to the (local / edge/ central) function which is selected to perform ML model training (App layer ML model training function) and provides the ML model information, as well as the file address of the initial ML model or the ML model itself. It may also provide the requirements in terms of training algorithms to be used, type of training, whether it is online or offline training, time to provide trained models and the address / entity ID to send the trained model (if the ML model inference entity is not the C-ADAES by another entity e.g. ADAEC or edge ADAES.).

[0103] Step 6: The ADAES may either, 615, select to derive analytics for the VAL performance using ML model inference by itself or may delegate the ML model inference to one or more ADAECs or edge ADAES and acts as aggregator. In the latter, the ADAES subscribes to the entity which acts as ML model inference entity, and sends a requirement for performing analytics as well as the configuration parameters for performing ML model inference. It also may provide the ID and address of the ML model training entity, the ML model aggregator and the ML model reporting configuration information. As following step X, the App layer ML model training function (local/edge or global) collects the data to train the model from the data producers (based on the assumptions for data producers as in procedures in clauses 8.2.2 and 8.2.3 of TS 23.436).

[0104] Step 7. The ML model training entity performs, 616, the training of the ML model (how the collection of data for model training happens is out of scope of this solution) and sends, 617, the trained model either indirectly (via C-ADAES) or directly, 618 to the ML model inference entity. If the ML model inference entity is the VAL UE, this happens via ADAE-UU interface.

[0105] Step 8. The ML model inference entity (ADAES or ADAEC) performs ML model inference, 618, and then derives analytics (predictions) based on procedures in clauses 8.2.2 and 8.2.3 of TS 23.436 for VAL performance analytics.

[0106] Step 9. The ML model inference entity (-ies) sends the analytics outputs to the C-ADAES.

[0107] Step 10. The C-ADAES processes/filters or aggregates, 620 the analytics outputs from one or more ML model inference entities based on the Analytics ID and the Consumer type. Different aggregation or processing may be possible based on the service requirement.

[0108] Step 11. C-ADAES sends, 621, the processed/aggregated or original analytics outputs as notification to the Consumer.

[0109] In the above, the entities selected to perform the ML training or inference may be referred to as application entities.

[0110] Figure 7 illustrates an example of a signaling and data flow diagram, 700, in accordance with aspects of the present disclosure. This embodiment is directed to improving edge load analytics service using ML-enabled training and inference at the ADAE layer or via external entity. This procedure is enhancement / complementary to the procedure in clause 8.8 of TS 23.436 (applying to both request-response and subscribe-notify models).

[0111] In this embodiment, more than one edge ADAES may be used to provide either ML model inference, whereas the ML model training happens at the cloud. The scenario for having a plurality of edge ADAES to provide ML model inference can be for the following assumptions:

- Edge load analytics can be per EAS/EES in cases where an EAS/EES service area is covered by more than one EDNs which may be partially overlapping; hence edge load analytics will need to considered analytics outputs from more than one ADAES in more than one EDNs
- Edge load analytics per EES/EAS/EDN/DNAI when the analytics request is covering a list of destination EES/EAS/EDN/DNAI (and edge service areas are not necessarily overlapping)
- Edge load analytics per EDN/EES/EAS, when the processing load or energy consumption in the destination EDN (where the EES/EAS is hosted) is high and the ML model training / inference is handled at another ADAES in another central DN or EDN with lower expected load.

[0112] In this embodiment, the analytics enabler entity is a centralized ADAE server (C-ADAES), 702, which receives instructions from an analytics consumer, 701. There is provided an application model training entity, 703, an Edge ADAES, 704, a 5G network, 705, an EDN, 706, with EAS/EES, 707, and an A-ADRF registry 708. The signaling and data transfers comprise the following steps:

[0113] Step 1: The consumer of the ADAES analytics service sends an Edge Analytics subscription request to ADAES, 709. This request may be referred to as a service requirement. In an embodiment, this request may include one or more of:

- Consumer ID, Analytics ID, Analytics filter information, Analytics type (prediction, statistics), VAL service ID, Target data producer profile criteria, Preferred confidence

level, Area of Interest, Time validity, Destination EAS ID, Destination EES ID, DNN/DNAI

- ML-enabled analytics requirement / flag, ML model information including ML model profile/context to support selection of ML model by the ADAE layer, List of ML model identities (if known by the VAL customer) and optionally the initial ML models.
- Service Area identifying one or a list of EDN service areas.

[0114] Step 2: The ADAES sends a subscription response, 710 as a positive or negative acknowledgement to the consumer of the edge analytics service.

[0115] Step 3: The ADAES determines, in an embodiment, the need for ML-enabled analytics, 711. In an alternative embodiment, a request for ML analytics may be sent by the analytics consumer. The ADAES then determines at least one entity to be considered for ML model training entity (internal to SEAL or external), and at least one entity for performing ML model inference for the analytics ID, DNN/DNAI and/or consumer ID. The determination can be based, in embodiments on one or more the capabilities (energy constrains, processing capabilities, availability/reliability of entity at the target edge platform) as well as the Analytics Event expected output granularity (for example based on the expected analytics output and whether it is offline or online analytics), the preference to entities to serve as ML model training/inference entities which are close to the related data producers/EDN, interoperability constraints assuming multi-vendor models and data producers. The entities selected to perform the ML training or inference may be referred to as application entities.

[0116] The determination of an entity can be not only for an instantiated ADAES, but may indicate the target EDN where the ADAES can be instantiable and needs to be instantiated so as to undertake the ML training and/or inference. Such determination in some embodiments may trigger the instantiation of the ADAES at the destination EDN.

[0117] Step 4a: The ADAES (central or anchor) sends, 712 a subscribe request to the ML model registry (this can be the A-ADRF) to receive the one or more ML model identities/information for the Analytics ID and/or the Consumer type (e.g. EAS #1) as well as based on the destination EDN/DNAI. The ML model registry is assumed to have a pre-

configured mapping between Analytics ID / EDN(s) and ML model(s) or this can be determined and provided by the ADAES or in step 1 by the VAL customer.

[0118] Step 4b: The ADAES (central or anchor) receives a subscribe response, 713, which includes the ML model information and the mapping to the analytics ID / consumer type / EDN, the ML model file address for the initial model, the permissions for updating the ML model (related to who is the consumer), the ML model vendor ID, the ECSP identify who is expected provide the ML model, a list of ML model training entities which are allowed/preferred to train the model at one or more EDNs, a charging model for using the ML model provided by the vendor / ECSP.

[0119] Step 5: The ADAES, if not the ML model training entity, subscribes, 714, to the entity which is selected to perform ML model training and provides the ML model information, as well as the file address of the initial ML model or the ML model itself. It may also provide the requirements in terms of training algorithms to be used, type of training, whether it is online or offline training, time to provide trained models and the address / entity ID to send the trained model (if the ML model inference entity is not the C-ADAES by one or more edge ADAESs).

[0120] Step 6: The ADAES may either, 715, select to derive analytics for the VAL performance using ML model inference by itself or may delegate the ML model inference to one or more ADAECs or edge ADAES and acts as aggregator. In the latter, the ADAES subscribes to the entity which acts as ML model inference entity and sends a requirement for performing analytics as well as the configuration parameters for performing ML model inference. It also may provide the ID and address of the ML model training entity, the ML model aggregator and the ML model reporting configuration information.

[0121] Step 7. The ML model training entity performs, 716, the training of the ML model (the collection of data for model training is out of scope of this solution) and sends the trained model either indirectly (via C-ADAES), 717, or directly, 718, to the ML model inference entity.

[0122] Step 8. The ML model inference entity (one or more edge ADAESs in one or more EDNs) performs ML model inference, 719, and then derives analytics (predictions) based on procedures in clause 8.8.2 of TS 23.436 for Edge load performance analytics.

[0123] Step 9. The ML model inference entity (-ies) sends, 720, the analytics outputs to the C-ADAES

[0124] Step 10. The C-ADAES processes/filters or aggregates, 721, the analytics outputs from one or more ML model inference entities based on the Analytics ID and the Consumer type. Different aggregation or processing may be possible based on the service requirement.

[0125] Step 11. C-ADAES sends, 722 the processed/aggregated or original analytics outputs as notification to the Consumer.

[0126] Figure 8 illustrates an example of a UE 800 in accordance with aspects of the present disclosure. The UE 800 may include a processor 802, a memory 804, a controller 806, and a transceiver 808. The processor 802, the memory 804, the controller 806, or the transceiver 808, or various combinations thereof or various components thereof may be examples of means for performing various aspects of the present disclosure as described herein. These components may be coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces.

[0127] The processor 802, the memory 804, the controller 806, or the transceiver 808, or various combinations or components thereof may be implemented in hardware (e.g., circuitry). The hardware may include a processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), or other programmable logic device, or any combination thereof configured as or otherwise supporting a means for performing the functions described in the present disclosure.

[0128] The processor 802 may include an intelligent hardware device (e.g., a general-purpose processor, a DSP, a CPU, an ASIC, an FPGA, or any combination thereof). In some implementations, the processor 802 may be configured to operate the memory 804. In some other implementations, the memory 804 may be integrated into the processor 802. The processor 802 may be configured to execute computer-readable instructions stored in the memory 804 to cause the UE 800 to perform various functions of the present disclosure.

[0129] The memory 804 may include volatile or non-volatile memory. The memory 804 may store computer-readable, computer-executable code including instructions when executed by the processor 802 cause the UE 800 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 804 or another type of memory. Computer-readable media includes both non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A non-transitory storage medium may be any available medium that may be accessed by a general-purpose or special-purpose computer.

[0130] In some implementations, the processor 802 and the memory 804 coupled with the processor 802 may be configured to cause the UE 800 to perform one or more of the functions described herein (e.g., executing, by the processor 802, instructions stored in the memory 804). For example, the processor 802 may support wireless communication at the UE 800 in accordance with examples as disclosed herein. The UE 800 may be configured to support a means for receiving from an analytics enabler entity machine learning model information, for being configured, based on the machine learning model information to implement one or more of: a machine learning training task, a machine learning inference task or a data collection task; and sending derived analytics output and/or collected data to the analytics enabler entity. Although these features are illustrated and described in terms of a user equipment, this is only an example of the type of entity that may perform these tasks. Any of the entities described above or in the claims as application entities, may perform these tasks.

[0131] The controller 806 may manage input and output signals for the UE 800. The controller 806 may also manage peripherals not integrated into the UE 800. In some implementations, the controller 806 may utilize an operating system such as iOS®, ANDROID®, WINDOWS®, or other operating systems. In some implementations, the controller 806 may be implemented as part of the processor 802.

[0132] In some implementations, the UE 800 may include at least one transceiver 808. In some other implementations, the UE 800 may have more than one transceiver 808. The

transceiver 808 may represent a wireless transceiver. The transceiver 808 may include one or more receiver chains 810, one or more transmitter chains 812, or a combination thereof.

[0133] A receiver chain 810 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 810 may include one or more antennas for receive the signal over the air or wireless medium. The receiver chain 810 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 810 may include at least one demodulator configured to demodulate the receive signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal. The receiver chain 810 may include at least one decoder for decoding the demodulated signal to receive the transmitted data.

[0134] A transmitter chain 812 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 812 may include at least one modulator for modulating data onto a carrier signal, preparing the signal for transmission over a wireless medium. The at least one modulator may be configured to support one or more techniques such as amplitude modulation (AM), frequency modulation (FM), or digital modulation schemes like phase-shift keying (PSK) or quadrature amplitude modulation (QAM). The transmitter chain 812 may also include at least one power amplifier configured to amplify the modulated signal to an appropriate power level suitable for transmission over the wireless medium. The transmitter chain 812 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0135] Figure 9 illustrates an example of a processor 900 in accordance with aspects of the present disclosure. The processor 900 may be an example of a processor configured to perform various operations in accordance with examples as described herein. The processor 900 may include a controller 902 configured to perform various operations in accordance with examples as described herein. The processor 900 may optionally include at least one memory 904, which may be, for example, an L1/L2/L3 cache. Additionally, or alternatively, the processor 900 may optionally include one or more arithmetic-logic units (ALUs) 906. One or more of these components may be in electronic communication or otherwise coupled

(e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces (e.g., buses).

[0136] The processor 900 may be a processor chipset and include a protocol stack (e.g., a software stack) executed by the processor chipset to perform various operations (e.g., receiving, obtaining, retrieving, transmitting, outputting, forwarding, storing, determining, identifying, accessing, writing, reading) in accordance with examples as described herein. The processor chipset may include one or more cores, one or more caches (e.g., memory local to or included in the processor chipset (e.g., the processor 900)) or other memory (e.g., random access memory (RAM), read-only memory (ROM), dynamic RAM (DRAM), synchronous dynamic RAM (SDRAM), static RAM (SRAM), ferroelectric RAM (FeRAM), magnetic RAM (MRAM), resistive RAM (RRAM), flash memory, phase change memory (PCM), and others).

[0137] The controller 902 may be configured to manage and coordinate various operations (e.g., signaling, receiving, obtaining, retrieving, transmitting, outputting, forwarding, storing, determining, identifying, accessing, writing, reading) of the processor 900 to cause the processor 900 to support various operations in accordance with examples as described herein. For example, the controller 902 may operate as a control unit of the processor 900, generating control signals that manage the operation of various components of the processor 900. These control signals include enabling or disabling functional units, selecting data paths, initiating memory access, and coordinating timing of operations.

[0138] The controller 902 may be configured to fetch (e.g., obtain, retrieve, receive) instructions from the memory 904 and determine subsequent instruction(s) to be executed to cause the processor 900 to support various operations in accordance with examples as described herein. The controller 902 may be configured to track memory address of instructions associated with the memory 904. The controller 902 may be configured to decode instructions to determine the operation to be performed and the operands involved. For example, the controller 902 may be configured to interpret the instruction and determine control signals to be output to other components of the processor 900 to cause the processor 900 to support various operations in accordance with examples as described herein. Additionally, or alternatively, the controller 902 may be configured to manage flow of data

within the processor 900. The controller 902 may be configured to control transfer of data between registers, arithmetic logic units (ALUs), and other functional units of the processor 900.

[0139] The memory 904 may include one or more caches (e.g., memory local to or included in the processor 900) or other memory, such RAM, ROM, DRAM, SDRAM, SRAM, MRAM, flash memory, etc. In some implementations, the memory 904 may reside within or on a processor chipset (e.g., local to the processor 900). In some other implementations, the memory 904 may reside external to the processor chipset (e.g., remote to the processor 900).

[0140] The memory 904 may store computer-readable, computer-executable code including instructions that, when executed by the processor 900, cause the processor 900 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such as system memory or another type of memory. The controller 902 and/or the processor 900 may be configured to execute computer-readable instructions stored in the memory 904 to cause the processor 900 to perform various functions. For example, the processor 900 and/or the controller 902 may be coupled with or to the memory 904, the processor 900, the controller 902, and the memory 904 may be configured to perform various functions described herein. In some examples, the processor 900 may include multiple processors and the memory 904 may include multiple memories. One or more of the multiple processors may be coupled with one or more of the multiple memories, which may, individually or collectively, be configured to perform various functions herein.

[0141] The one or more ALUs 906 may be configured to support various operations in accordance with examples as described herein. In some implementations, the one or more ALUs 906 may reside within or on a processor chipset (e.g., the processor 900). In some other implementations, the one or more ALUs 906 may reside external to the processor chipset (e.g., the processor 900). One or more ALUs 906 may perform one or more computations such as addition, subtraction, multiplication, and division on data. For example, one or more ALUs 906 may receive input operands and an operation code, which determines an operation to be executed. One or more ALUs 906 be configured with a variety of logical and arithmetic circuits, including adders, subtractors, shifters, and logic gates, to process and

manipulate the data according to the operation. Additionally, or alternatively, the one or more ALUs 906 may support logical operations such as AND, OR, exclusive-OR (XOR), not-OR (NOR), and not-AND (NAND), enabling the one or more ALUs 906 to handle conditional operations, comparisons, and bitwise operations.

[0142] The processor 900 may support wireless communication in accordance with examples as disclosed herein. The processor 900 may be configured to or operable to support a means for obtaining a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task, identifying a requirement for machine learning-enabled analytics for the application layer data analytics task, and upon identifying the requirement for machine learning-enabled analytics, obtaining machine learning model information from an application layer machine learning model repository, and selecting at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter, and at least one application entity parameters, and outputting an indication of the at least one application entity.

[0143] Figure 10 illustrates an example of a NE 1000 in accordance with aspects of the present disclosure. The NE 1000 may include a processor 1002, a memory 1004, a controller 1006, and a transceiver 1008. The processor 1002, the memory 1004, the controller 1006, or the transceiver 1008, or various combinations thereof or various components thereof may be examples of means for performing various aspects of the present disclosure as described herein. These components may be coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces.

[0144] The processor 1002, the memory 1004, the controller 1006, or the transceiver 1008, or various combinations or components thereof may be implemented in hardware (e.g., circuitry). The hardware may include a processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), or other programmable logic device, or any combination thereof configured as or otherwise supporting a means for performing the functions described in the present disclosure.

[0145] The processor 1002 may include an intelligent hardware device (e.g., a general-purpose processor, a DSP, a CPU, an ASIC, an FPGA, or any combination thereof). In some implementations, the processor 1002 may be configured to operate the memory 1004. In some other implementations, the memory 1004 may be integrated into the processor 1002. The processor 1002 may be configured to execute computer-readable instructions stored in the memory 1004 to cause the NE 1000 to perform various functions of the present disclosure.

[0146] The memory 1004 may include volatile or non-volatile memory. The memory 1004 may store computer-readable, computer-executable code including instructions when executed by the processor 1002 cause the NE 1000 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 1004 or another type of memory. Computer-readable media includes both non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A non-transitory storage medium may be any available medium that may be accessed by a general-purpose or special-purpose computer.

[0147] In some implementations, the processor 1002 and the memory 1004 coupled with the processor 1002 may be configured to cause the NE 1000 to perform one or more of the functions described herein (e.g., executing, by the processor 1002, instructions stored in the memory 1004). For example, the processor 1002 may support wireless communication at the NE 1000 in accordance with examples as disclosed herein. The NE 1000 may be configured to support a means for receiving a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task, identifying a requirement for providing machine learning-enabled analytics for the application layer data analytics task, and upon identifying the requirement for providing machine learning-enabled analytics, obtaining machine learning model information from an application layer machine learning model repository, and selecting at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter and at least one application entity parameter. The controller 1006 may manage input and output signals for the NE 1000. The controller 1006 may also manage

peripherals not integrated into the NE 1000. In some implementations, the controller 1006 may utilize an operating system such as iOS®, ANDROID®, WINDOWS®, or other operating systems. In some implementations, the controller 1006 may be implemented as part of the processor 1002.

[0148] In some implementations, the NE 1000 may include at least one transceiver 1008. In some other implementations, the NE 1000 may have more than one transceiver 1008. The transceiver 1008 may represent a wireless transceiver. The transceiver 1008 may include one or more receiver chains 1010, one or more transmitter chains 1012, or a combination thereof.

[0149] A receiver chain 1010 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 1010 may include one or more antennas for receive the signal over the air or wireless medium. The receiver chain 1010 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 1010 may include at least one demodulator configured to demodulate the receive signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal. The receiver chain 1010 may include at least one decoder for decoding the demodulated signal to receive the transmitted data.

[0150] A transmitter chain 1012 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 1012 may include at least one modulator for modulating data onto a carrier signal, preparing the signal for transmission over a wireless medium. The at least one modulator may be configured to support one or more techniques such as amplitude modulation (AM), frequency modulation (FM), or digital modulation schemes like phase-shift keying (PSK) or quadrature amplitude modulation (QAM). The transmitter chain 1012 may also include at least one power amplifier configured to amplify the modulated signal to an appropriate power level suitable for transmission over the wireless medium. The transmitter chain 1012 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0151] Figure 11 illustrates a flowchart of a method in accordance with aspects of the present disclosure. The operations of the method may be implemented by a UE as described

herein. In some implementations, the UE may execute a set of instructions to control the function elements of the UE to perform the described functions.

[0152] Figure 11 illustrates a flowchart of a method in accordance with aspects of the present disclosure. The operations of the method may be implemented by a NE as described herein. In some implementations, the NE may execute a set of instructions to control the function elements of the NE to perform the described functions.

[0153] At 1102, the method may include receiving a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task. The operations of 1102 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1102 may be performed by a NE as described with reference to Figure 10.

[0154] At 1104, the method may include identifying a requirement for providing machine learning-enabled analytics for the application layer data analytics task. The operations of 1104 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1104 may be performed by a NE as described with reference to Figure 10.

[0155] At 1106, the method may include upon identifying the requirement for providing machine learning-enabled analytics, at 1108, obtaining machine learning model information from an application layer machine learning model repository and, at 1110, selecting at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter and at least one application entity parameter. The operations of 1106, 1108 and 1110 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1106, 1108 and 1110 may be performed by a NE as described with reference to Figure 10.

[0156] Figure 12 illustrates a flowchart of a method in accordance with aspects of the present disclosure, as a further feature of the embodiment of Figure 11. The operations of the method may be implemented by a NE as described herein. In some implementations, the NE

may execute a set of instructions to control the function elements of the NE to perform the described functions.

[0157] At 1202, the method of Figure 11 may include configuring the at least one selected application entity using the machine learning model information. The operations of 1202 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1202 may be performed by a NE as described with reference to Figure 10.

[0158] At 1204, the method may further include, in an embodiment, receiving derived analytics output based on trained and/or inferred machine learning model data from the at least one selected application. The operations of 1204 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1204 may be performed by a NE as described with reference to Figure 10.

[0159] At 1206, the method may, optionally, further include, in an embodiment, processing the received derived analytics output, in particular by aggregating or filtering the data based on the vertical service requirement. The operations of 1206 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1206 may be performed by a NE as described with reference to Figure 10.

[0160] At 1208, the method may, optionally, further include, in an embodiment, sending the processed derived analytics output to the analytics consumer. The operations of 1208 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1208 may be performed by a NE as described with reference to Figure 10.

[0161] It should be noted that the method described herein describes a possible implementation, and that the operations and the steps may be rearranged or otherwise modified and that other implementations are possible.

[0162] The description herein is provided to enable a person having ordinary skill in the art to make or use the disclosure. Various modifications to the disclosure will be apparent to a person having ordinary skill in the art, and the generic principles defined herein may be applied to other variations without departing from the scope of the disclosure. Thus, the

disclosure is not limited to the examples and designs described herein but is to be accorded the broadest scope consistent with the principles and novel features disclosed herein.

Abbreviations

AF	Application Function
NF	Network Function
NWDAF	Network Data Analytics Function
OAM	Operations and Maintenance
UE	User Equipment
MDAS	Management Domain Analytics Service
C-ADAES	Centralized-Application Data Analytics Enabler Service/Server
ANLF	Analytics Logical Function
MTLF	Model Training Logical Function
DNAI	Data Network Access Identifier
ADAEC	Application Data Analytics Enabler Client
TRLF	Trusted Rating Logical Function
ML	Machine Learning
ADAE	Application Data Analytics Enablement
VAL	Vertical Application Layer
A-ADRF	Application layer – Analytics Data Repository Function
RTT	Round Trip Time
EES	Edge Enabler Server
EAS	Edge Application Server
EDN	Edge Data Network

DNN Data Network Name

WHAT IS CLAIMED IS:

1) An analytics enabler entity for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, the analytics enabler entity comprising:

at least one memory; and

at least one processor coupled with the at least one memory and configured to cause the analytics enabler entity to:

receive a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task;

identify a requirement for providing machine learning-enabled analytics for the application layer data analytics task, and

upon identifying the requirement for providing machine learning-enabled analytics:

obtain machine learning model information from a machine learning model repository; and

select at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter, the obtained machine learning model information, and at least one application entity parameter.

2) The analytics enabler entity according to claim 1, further configured to:

send the machine learning model information to the at least one selected application entity;

configure the at least one selected application entity using the machine learning model information; and

receive derived application data analytics output based on trained and/or inferred machine learning model data from the at least one selected application entity.

3) The analytics enabler entity according to claim 2, further configured to:

process the received derived application data analytics output based on at least one of the at least one analytics parameter of the service requirement, wherein the processing comprises one or more of aggregating or filtering data from the received derived analytics output; and

send the processed derived analytics output to the analytics consumer.

4) The analytics enabler entity according to any preceding claim, wherein the machine learning model processing comprises at least one ML model lifecycle operation, wherein the model lifecycle operation is one of: a machine learning model inference or machine learning training operation.

5) The analytics enabler entity according to any preceding claim, configured to identify at least one candidate entity, wherein, optionally, at least one of the at least one candidate entities is an application enablement server and/or client.

6) The analytics enabler entity according to any preceding claim, further configured to:

receive in the service requirement an analytics parameter comprising an indication of consumer type, wherein, optionally, the consumer type is one of: a Vertical Application Layer server, an Edge Application Servers, an Edge Enabler Server Application Client, an Edge Enabler Client, a Vertical Application Layer client, a Network Function, an Application Function, a management function or server, or an external application; and

select the at least one application entity based at least partly on the consumer type, and

wherein, optionally, the analytics enabler entity is further configured to identify the requirement for machine learning-enabled analytics for the application layer data analytics task by either: receiving from the analytics consumer an indication, or determining based on the at least one analytics parameter, that machine learning-enabled analytics is required.

7) The analytics enabler entity according to any preceding claim, wherein the at least one application entity parameter is one of: a proximity to data producers, a signaling cost, a latency, a data accessibility, a trust level, a reliability, an energy consumption, a cost, or, a location of and applications supported in an edge network.

8) The analytics enabler entity according to any preceding claim, configured to select at least one application entity for performing the machine learning model processing, wherein selecting at least one application entity comprises selecting one or both of an application layer machine model training functionality or application layer machine learning model inference functionality.

9) A method for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, the method comprising:

receiving a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task;

identifying a requirement for providing machine learning-enabled analytics for the application layer data analytics task; and

upon identifying the requirement for providing machine learning-enabled analytics:

obtaining machine learning model information from an application layer machine learning model repository; and

selecting at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter and at least one application entity parameter.

10) The method according to claim 9, further comprising:

configuring the at least one selected application entity using the machine learning model information; and

receiving derived analytics output based on trained and/or inferred machine learning model data from the at least one selected application entity.

11) The method according to claim 10, further comprising:

processing the received derived analytics output, in particular by aggregating or filtering the data based on the vertical service requirement; and

sending the processed derived analytics output to the analytics consumer.

12) The method according to any of claims 9 to 11, wherein the machine learning model processing comprises at least one ML model lifecycle operation, wherein the model lifecycle operation is one of: a machine learning model inference or machine learning training operation.

13) The method according to any of claims 9 to 12, further comprising identifying at least one candidate entity, wherein, optionally, at least one of the at least one candidate entities is an application enablement server and/or client.

14) The method according to any of claims 9 to 13, further comprising:

receiving in the service requirement an analytics parameter comprising an indication of consumer type, wherein, optionally, the consumer type is one of: a Vertical Application Layer server, an Edge Application Servers, an Edge Enabler Server Application Client, an Edge Enabler Client, a Vertical Application Layer client, a Network Function, an Application Function, a management function or server, or an external application; and

selecting the at least one application entity based at least partly on the consumer type, and

wherein, optionally, identifying the requirement for machine learning-enabled analytics for the application layer data analytics task comprises either receiving from the analytics consumer an indication, or determining based on the at least one analytics parameter, that machine learning-enabled analytics is required.

15) The method according to any of claims 9 to 14, wherein the at least one application entity parameter is one of: a proximity to data producers, a signaling cost, a latency, a data accessibility, a trust level, a reliability, an energy usage, a cost, or, a location of and applications supported in an edge network.

16) The method according to any of claims 9 to 15, wherein selecting at least one application entity comprises selecting one or both of an application layer machine model training functionality or application layer machine learning model inference functionality.

17) A processor for selecting one or more application entities for implementing machine learning model processing for application layer data analytics, comprising:
at least one controller coupled with at least one memory and configured to cause the processor to:

obtain a service requirement for an application layer data analytics task from an analytics consumer, wherein the service requirement comprises at least one analytics parameter for the application layer data analytics task;

identify a requirement for machine learning-enabled analytics for the application layer data analytics task, and

upon identifying the requirement for machine learning-enabled analytics:

obtain machine learning model information from an application layer machine learning model repository; and

select at least one application entity for performing the machine learning model processing from one or more candidate entities, based on the at least one analytics parameter, and at least one application entity parameters; and

output an indication of the at least one application entity.

18) The processor according to claim 17, further configured to:

configure the at least one selected application entity using the machine learning model information; and

receive derived analytics output based on trained and/or inferred machine learning model data from the at least one selected entity;

19) The processor according to claim 18, further configured to:

process the received derived analytics output based on at least one of the at least one analytics parameter of the service requirement, wherein the processing comprises one or more of aggregating or filtering data from the received derived analytics; and

output the processed derived analytics to the analytics consumer.

20) A network entity comprising an application entity for implementing machine learning model processing for application layer data analytics, the application entity comprising:

at least one memory; and

at least one processor coupled with the at least one memory and configured to cause the network entity to:

receive from an analytics enabler entity machine learning model information;

be configured, based on the machine learning model information to implement one or more of: a machine learning training task, a machine learning inference task or a data collection task; and

send derived analytics output and/or collected data to the analytics enabler entity.