GROUPING A PLURALITY OF ENTITIES FOR PARTICIPATING IN A MACHINE LEARNING OPERATION

TECHNICAL FIELD

[0001] The present disclosure relates to wireless communications, and more specifically to grouping a plurality of entities for participating in a machine learning (ML) operation.

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, subframes, 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] Use cases and requirements have been identified for the support of Artificial intelligence (AI) / Machine Learning (ML) model distribution, transfer, and training for various applications (e.g., video/speech recognition, robot control, automotive control), and also covering support for Distributed AI training/inference based on direct device connection.

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 further include a network entity for grouping a plurality of entities for participating in a machine learning (ML) operation, wherein the plurality of entities are coupled to a wireless communication system, the network 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: obtain information relating to the ML operation; create a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation; determine a unique group identifier for the group; and transmit the unique group identifier for delivery to each of the plurality of entities.

[0006] The information relating to the ML operation may comprise one or any combination of: (i) an ML model identifier; (ii) a service identifier or service area or service profile; (iii) at least one target user equipment identifier; and (iv) an analytics identifier.

[0007] The information relating to the ML operation may comprise one or any combination of: (v) one or more KPIs for the ML operation; (vi) a data requirement for the ML operation; (vii) one or more entity types; (viii) an indication as to whether model training and/or inference is needed; (ix) model vendor information; and (x) a training job ID.

[0008] The at least one processor may be configured to cause the network entity to: transmit a request message to a ML model repository, the request message comprising the information relating to the ML operation; and receive a response message from the ML model repository, the response message comprising a list of candidate entities to perform the ML operation.

[0009] The at least one processor may be configured to cause the network entity to: select the plurality of entities from the list of candidate entities to create the group.

[0010] The at least one processor may be configured to cause the network entity to determine the unique group identifier randomly.

[0011] The at least one processor may be configured to: determine a mapping of the unique group identifier to a set of member identifiers of the plurality of entities in the group; and store the mapping in the at least one memory.

[0012] The at least one processor may be configured to: detect a trigger event for modifying the group; based on the trigger event, update the group to create an updated group of a plurality of entities to perform the ML operation; and modify the mapping so that the unique group identifier is associated with a set of member identifiers of the plurality of entities in the updated group.

[0013] The trigger event may be: an entity joining the group; one of the plurality of entities leaving the group; one of the plurality of entities being unavailable to perform the ML operation; an expected or predicted failure to meet the one or more KPIs for the ML operation; one of the plurality of entities being incapable to perform the ML operation; or an expected or predicted failure or congestion of a wireless interface coupled with the at least one of the plurality of the entities.

[0014] The at least one processor may be configured to monitor a status of the plurality of entities in the group to detect the trigger event.

[0015] The at least one processor may be configured to: periodically transmit a status request to each of the plurality of entities in the group; and receive a status response from each of the plurality of entities in the group.

[0016] The at least one processor may be configured to configure the plurality of entities in the group to periodically transmit a status update to the network entity.

[0017] The at least one processor may be configured to transmit the unique group identifier and identities of members of the updated group for delivery to at least one of the plurality of entities in the updated group.

[0018] The network entity may be a Service Enabler Architecture Layer (SEAL) group management server. The plurality of entities in the group may comprise one or any combination of: at least one FL client, at least one FL server, at least one FL aggregator or collaborator, at least one ML client, and at least one ML server.

[0019] The network entity may be a Service Enabler Architecture Layer (SEAL) group management server, an AI enabler server or an Application Data Analytics Enabler Server or an Edge Enabler Server.

[0020] The plurality of entities in the group may comprise at least one Edge Application Server and/or at least one UE application client.

[0021] The at least one processor may be configured to transmit the identities of the plurality of entities of the group for delivery to at least one of the plurality of entities in the group.

[0022] Some implementations of the method and apparatuses described herein may further include a method for grouping a plurality of entities for participating in a machine learning (ML) operation, wherein the plurality of entities are coupled to a wireless communication system, the method comprising: obtaining information relating to the ML operation; creating a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation; determining a unique group identifier for the group; and transmitting the unique group identifier for delivery to each of the plurality of entities.

[0023] Some implementations of the method and apparatuses described herein may further include a processor for grouping a plurality of entities for participating in a machine learning (ML) operation, the processor comprising: at least one controller coupled with at least one memory and configured to cause the processor to: obtain information relating to the ML operation; create a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation; determine a unique group identifier for the group; and output the unique group identifier for delivery to each of the plurality of entities.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Figure 1 illustrates an example of a wireless communications system in accordance with aspects of the present disclosure.

[0025] Figure 2 illustrates a high level Service Enabler Architecture Layer (SEAL) architecture in accordance with aspects of the present disclosure.

[0026] Figure 3 illustrates an architecture for enabling edge applications in accordance with aspects of the present disclosure.

[0027] Figures 4a and 4b illustrate Edge Application Server (EAS) bundling use cases.

[0028] Figure 5 illustrates an architecture to enable the group management of the entities in accordance with aspects of the present disclosure.

[0029] Figure 6 is a flow diagram of a first embodiment for grouping a plurality of entities for participating in a ML operation.

[0030] Figure 7 is a flow diagram of a second embodiment for grouping a plurality of entities for participating in a ML operation.

[0031] Figure 8 illustrates an example of a processor 800 in accordance with aspects of the present disclosure.

[0032] Figure 9 illustrates an example of a network equipment (NE) 900 in accordance with aspects of the present disclosure.

[0033] Figure 10 illustrate a flowcharts of method performed by a NE in accordance with aspects of the present disclosure.

[0034] Figure 11 illustrate how the architecture of Figure 5 may be coupled to a wireless communications system in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

[0035] 3GPP has identified use cases and requirements (in TS 22.261) for the support of Artificial intelligence (AI)/ Machine Learning (ML) model distribution, transfer, and training for various applications (e.g., video/speech recognition, robot control, automotive control),

and also covering support for Distributed AI training/inference based on direct device connection.

[0036] Relating to AI/ML support, the 3GPP SA2 (system architecture group) has investigated the following two aspects:

- Enhancements to a Network Data Analytics Function (NWDAF) for supporting Federated learning (FL) enabled analytics to improve an analytics output. In this enhancement, NWDAFs acting as FL members (servers or clients) register to a Network Repository Function (NRF) and can be discovered by other Network Functions (NFs) (currently only by other NWDAFs).
- Enhancements to 5G Core (5GC) (as specified in TS 23.501 clause 5.46) for assisting AI/ML (including FL) operations in an application layer (between one or more AI/ML users and an AI/ML server). In this enhancement, a Network Exposure Function (NEF) may assist the AI/ML application server in scheduling available UE(s) to participate in the AI/ML Federated Learning. Also, 5GC may assist in the selection of UEs to serve as FL clients, by providing a list of target member UE(s), to an Application Function (AF) that is subscribed to the NEF, to be notified about a subset list of UE(s) (i.e., a list of candidate UE(s)) that fulfil certain filtering criteria).

[0037] In addition, 3GPP SA6 has specified an application layer architecture to enable data analytics as a new Service Enabler Architecture Layer (SEAL) service, also known as an Application Data Analytics Enablement Service (ADAES), in 3GPP TS 23.436. Such architecture (illustrated in Figure 2) provides an application layer analytics framework which offers generic analytics exposure and value-added services for verticals and ASPs. This includes application layer analytics related to end-to-end application performance, edge load, service API availability, location accuracy, and slice-related performance and fault analysis.

[0038] Group management is defined in 3GPP SA6. One aspect of the group management defined in 3GPP SA6 is the SEAL group management service. The group management (GM) service is a SEAL service that offers the group management related capabilities to one or more vertical applications. It consists of a GM server (GMS) and one or more GM clients (GMC) and Figure 2 illustrates the on-network functional model for group management in the SEAL group management service.

[0039] The group management client communicates with the group management server over the GM-UU reference point. The group management client provides the support for group management functions to the Vertical Application Layer (VAL) client(s) over GM-C reference point. The VAL server(s) communicate with the group management server over the GM-S reference point. The group management server interacts with the NEF of the underlying 3GPP network system via N33 reference point to perform group management procedures for 5G Virtual Network (5GVN) groups.

[0040] There is also a service-based representation as well as an off-network deployment model (GMC to GMC interactions) specified in TS 23.434 10.2. The capabilities provided in group management are presented in procedures in TS 23.434 10.3 which comprise the:

- Group creation
- Group information query
- Group membership update by authorized user/UE/VAL server
- Group configuration management
- Location-based group creation
- Group member leave
- Temporary group formation within a VAL system

[0041] Such capabilities are generic and aim to support vertical applications (e.g., automotive, factories, uncrewed aerial systems (UAS), ...); however the group creation and management is applicable to the grouping of VAL UEs or VAL entities in general, and not on the enablers of different types as well as the grouping for a particular purpose (e.g. ML support).

[0042] Another aspect of the group management defined in 3GPP SA6 is EDGEAPP which specifies an application layer architecture for edge services. In this architecture, as illustrated in Figure 3, the following entities can be defined:

- Edge Enabler Server (EES) provides supporting functions needed for Edge Application Servers and Edge Enabler Client, such as:

a) provisioning of configuration information to Edge Enabler Client, enabling exchange of application data traffic with the Edge Application Server;

c) interacting with 3GPP Core Network for accessing the capabilities of network functions either directly (e.g. via PCF) or indirectly (e.g. via SCEF/NEF/SCEF+NEF);

e) supports external exposure of 3GPP network capabilities to the Edge Application Server(s) over EDGE-3

- Edge Enabler Client (EEC) provides supporting functions needed for Application Client(s), such as retrieval and provisioning of configuration information to enable the exchange of Application Data Traffic with the Edge Application Server; and discovery of Edge Application Servers available in the Edge Data Network.

- Edge Configuration Server (ECS) provides supporting functions needed for the Edge Enabler Client to connect with an Edge Enabler Server. These functionalities of Edge Configuration Server are related to the provisioning of edge configuration information to the EEC, which are used for establishing connection with EES.

- Edge Application Server (EAS) is the application server resident in the Edge Data Network, performing the server functions.

- Application Client (AC): Application Client is the application resident in the UE performing the client function.

[0043] One of the features in EDGEAPP Rel-18 is the support for EAS bundles. Such bundles are mainly supported for cases where the EAS(s) are grouped together due to dependencies or need for operating together even if these may belong to different edge/cloud platforms.

Two use cases have been identified in 3GPP TR 23.700-98 which are illustrated in Figures 4a and 4b.

[0044] Figure 4a illustrates an EAS bundle example. To provide services to the end user a typical AC communicates with multiple endpoints i.e. multiple EASs. In an online game where to support large number of users, different game functions are split across multiple servers: (i) a game engine (e.g. EAS1) for game state and user input management, (ii) in-game chat server(e.g. EAS2) for communication between players, and (iii) a capture server (e.g. EAS3) for capturing rendered images, encoding, and transporting them to the player's device. If each of these EASs are discovered, controlled, and relocated individually, it may impact the overall quality of service.

[0045] Figure 4b illustrates a composite EAS example. To provide services (weather, transportation, maps, etc.) in partnership with other EASs, EAS context processing and composite EAS support may be required at edge-compatible layers. When UE mobility occurs, a method of rearranging the composite EAS context may be required to provide continuous service of the composite EASs. In addition, there may be a need for a method for finding an EAS that provides services to composite EASs within the EDN in which the UE has moved.

[0046] One aspect not covered in the 3GPP specifications is how to support the management of ML/FL groups (e.g., how to create and manage a group of ML/FL members) during ML/FL operations (e.g., training).

[0047] Federated Learning (or more generally an ML workflow) involves a plurality of ML/FL members which may act as one or more of a FL server, FL aggregator, FL client to perform jointly or coordinately ML training and/or inference processes. The grouping of these entities for a particular ML task (e.g. training) may be important for optimizing performance, especially for cases where FL members reside in external systems and end devices (with different availabilities and capabilities). Embodiments of the present disclosure address how to support the grouping of application layer entities to perform a ML operation (e.g. to serve as FL participants) and how to create and manage the groups for a particular ML/FL task involving plurality of members. In particular, embodiments of the present disclosure provide a mechanism for creating and managing groups of entities to serve as ML/FL members for a particular vertical/ Application Service Provider (ASP) requirement. Such grouping is supported by a new enablement functionality which generates group identities and may provide the mapping to member IDs dynamically.

[0048] Federated Learning (FL) can be defined as a machine learning technique that enables multiple FL clients to train a model by exchanging the parameters instead of exchanging/sharing local data set. FL servers perform management of FL operations by maintaining and updating a global ML model, selecting and managing FL clients, performing aggregation strategies, scheduling training in a federated manner, and communicating with FL clients. FL clients provide various aspects of data for FL operations, such as data collection, data preparation, and using data for training and/or inferencing while communicating updates of ML models to FL servers.

[0049] Aspects of the present disclosure are described in the context of a wireless communications system.

[0050] 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 include one or more NE 102, one or more UE 104, and a core network (CN) 106. 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.

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 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.

[0055] 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).

[0056] 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.

[0057] 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).

[0058] 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.

[0059] 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., $\mu=0$) 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., $\mu=0$) 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.

[0060] 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.

[0061] 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.

[0062] 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.

[0063] 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.

[0064] Figure 5 illustrates an example of an architecture 500 in accordance with aspects of the present disclosure to enable the group management of the entities (e.g. entities serving as ML entities or FL clients) at the application enablement layer. Such group management comprises the creation, monitoring and update of the groups. This may be based on the AI/ML lifecycle/workflows/pipeline, which are based on 1) the analytics event/service by ADAES or 2) the VAL requirement for ML/FL support services. The plurality of entities formed within a group in accordance with embodiments of the invention are coupled to a wireless communication system, for example the wireless communication system 100 described above. This is illustrated in Figure 11. That is, an entity within a group may for example be a UE 104, a client or application running on a UE 104, a server, or a network element e.g. a network element in the core network 106.

[0065] As shown in Figure 5, the architecture 500 comprises a ML model producer 502, a ML model repository 504, a ML model consumer 506, a ML model management service consumer 508, a ML model management service producer 510. The architecture 500 additionally comprises at least one ML model training service producer 512 and/or at least one ML model inference service producer(s) 514.

[0066] The ML model producer 502 is an entity that is providing the ML model. Such entity can be the VAL server or EAS or Application specific client at a UE 104. Such entity can belong to a non-3GPP system or to an edge/cloud platform provider. In certain implementations, the ML model producer 502 can be the mobile network operator (MNO) itself.

[0067] The ML model repository 504 may be a Common API Framework (CAPIF) Core Function (CCF) as defined in 3GPP TS 23.222, edge or central platform repository, Application layer – Analytics Data Repository Function (A-ADRF), VAL User Database (UDB) as defined in 3GPP TS 23.434, and application layer registry, an embedded registry at ADAES, or a Network Repository Function (NRF). The ML model repository 504 is configured to store data. For example the ML model repository 504 may store ML model(s), ML model IDs, and/or information on candidate participants (e.g. ML or FL candidate participants) which may perform a ML operation.

[0068] The ML model consumer 506 is an entity which consumes the ML model after agreement with the ML model producer 502. The agreement can be in-direct. For example, a 3rd party registers the ML model at the ML model repository 504 of the MNO (e.g. edge or cloud registry), and another trusted 3rd party consumes it after agreement with the MNO. The ML model consumer 506 may be a VAL server. In embodiment of the present disclosure a VAL server may be a gaming server or AI App Server or XR server or V2X server.

[0069] A ML model training service producer 512 is an entity which undertakes the task of ML model training for the identified model. In case of Federated Learning (FL) multiple producers are being identified. Such a ML model training service producer 512 in the context of SA6, can be an enablement functionality at the server or client side. In certain scenarios, the ML model training service producer 512 may be an external application or system (e.g. ETSI MEC, O-RAN,..) after agreement with the MNO or SA6 service provider.

[0070] An ML model inference service producer 514 is an entity which offers the ML model inference. Such entity can be an entity with access for instance to real time data and can improve the trained ML model with more accurate/granular data. An example of such entity can be the VAL UE (enabler client or VAL client or AC).

[0071] The architecture 500 further comprises a ML model management service consumer ML-MMSC 508 (which may for example be a FL aggregator, an application data analytics enablement server, or Edge Enabler Server (EES)), and a ML model management service producer ML-MMSP 510 (which may for example be a SEAL group management server, an AI/ML enabler server, an Application Data Analytics Enabler Server or an Edge Enabler Server). Whilst the ML-MMSC 508 and the ML-MMSP 510 have been shown as separate entities in Figure 5, in some embodiment of the present disclosure the functionality of the ML-MMSC 508 and the ML-MMSP 510 may be performed by a single entity.

[0072] In the architecture 500 an entity takes on the role of a ML model training service consumer. The ML model training service consumer is an entity which consumes the ML model training service provided by one or multiple producers (e.g. ML model training service producer(s) 512). Such consumer may be an FL server /aggregator in case of FL. Such consumer may be a network function (e.g. ANLF NWDAF) or can be a VAL server or any other enabler (SEAL, EDGE). Such entity can be a UE functionality. The ML model consumer 506 and/or the ML-MMSC 508 may take on the role of ML model training service consumer.

[0073] In the architecture 500 an entity takes on the role of a ML model inference service consumer. The ML model inference service consumer is an entity which consumes the ML model inference services. Such entity can be the ML model training service producer 512 at the enablement or application specific layer or even at a network function e.g. a NWDAF, or Model Training Logical Function (MTLF).

[0074] In the architecture 500 an entity takes on the role of a ML group member management service producer. This service can be a standalone service or part of the ML model management services. That is, the ML-MMSP 510 may perform the ML group member management service producer role. The producer of such service provides the group management for the ML model participants (ML clients / ML server or FL server and list of FL clients), as part of the group configuration for a given ML model and use case. The use

case may be defined either by the application service, or service area, or service type, or slice, or analytics event or VAL requirement.

[0075] In the architecture 500 an entity takes on the role of a ML group member management service consumer: the consumer of the group management service can be the ML model training service producer 512, or any other entity involved in the process of ML model lifecycle. The ML-MMSC 508 may perform the ML group member management service consumer role.

[0076] At step S502, the ML model producer 502 registers a ML model at the ML model repository 504. At step S502, a ML model identifier (ID) associated with the ML model is stored in the ML model repository 504. In addition, ML model information, context, and/or parameters may also be provided for storage in the ML model repository 504. It will be appreciated that multiple models may be registered at the repository 504 by one or more ML model producers 502.

[0077] At step S504, candidate participants able participate in performing a ML operation register at the ML model repository 504 such that information identifying the candidate participants is stored at the ML model repository 504. Each candidate participant stores information at the ML model repository 504 which indicates what ML operations they may be involved in. For example, the ML model repository 504 may store each candidate participant in association with (i) an ML model identifier; (ii) a service identifier or service area or service profile (e.g. type of service, e.g. URLLC), (iii) at least one target user equipment identifier, (iv) an analytics identifier, and/or capabilities of the candidate participant. The service identifier can be an application service ID, a vertical service ID, an application ID, or an application server ID. The candidate participants may comprise at least one ML model training service producer 512 and/or at least one ML model inference service producer(s) 514.

[0078] At step S506, the ML model consumer 506 transmits a request to the ML-MMSC 508 for providing AI/ML support services. The request may be a VAL request when the ML model consumer 506 is a VAL server. The request transmitted at step S506 request can be a request for ADAES analytics or for support for the training/inference of an ML model. This request may include the ML model ID, the application service ID or profile, the vertical service requirements, a service area, a list of VAL UEs for which this request applies, a

PLMN/NPN ID, and/or the data requirements for the training (e.g. a requirement to handle real-time data).

[0079] At step S508, based on the request the ML-MMSC 508 determines a need to consume group management services (e.g. a need for FL or ML model training split, or for multiple ML participants for the ML training job).

[0080] At step S510, the ML-MMSC 508 transmits a ML member group management request to the ML-MMSP 510 for creating or configuring a group comprising the ML or FL participants for the given task (which can for example be per service or service area or model ID or analytics ID). The ML member group management request comprises information relating to a ML operation (which may be a FL operation). The information relating to the ML operation may comprise one or more of: (i) an ML model identifier; (ii) a service identifier (can be an application service ID, a vertical service ID, an application ID, or an application server ID) or service area or service profile; (iii) at least one target user equipment identifier; (iv) an analytics identifier; (v) one or more KPIs for the ML operation; (vi) a data requirement for the ML operation; (vii) one or more entity types; (viii) an indication as to whether model training and/or inference is needed; (ix) model vendor information; and (x) a training job ID.

[0081] The ML-MMSP 510 is configured to create a group of a plurality of entities to perform the ML operation based on the information relating to the ML operation. This may be achieved by the ML-MMSP 510 by transmitting, at step S512, a request message to the ML model repository 504, the request message comprising the information relating to the ML operation. At step S512 the ML-MMSP 510 receive a response message from the ML model repository 504, the response message comprising a list of candidate entities to perform the ML operation. In particular, the list of candidate entities are associated with the information relating to the ML operation in the ML model repository 504. In particular, the list of candidate entities may be associated with a given ML model ID, service ID or analytics ID.

[0082] At step S514, the ML-MMSP 510 is configured to determine a group of a plurality of entities to perform the ML operation using the list of candidate entities. The ML-MMSP 510 may determine the members of the group from the list of candidate entities based on policy information or based on some predefined criteria. The ML-MMSP 510 may determine

the members of the group based on the capabilities of the candidate participants (computational resources, maximum load, thresholds), whether the candidate participants are fixed or mobile nodes and their availabilities, the proximity of the candidate participants among them and any interface limitations, the dependencies among candidate participants, how physically close to the data producers (based on the data requirements) are the candidate participants, and/or the energy status/constraints for the candidate participants.

[0083] At step S514, the ML-MMSP 510 is configured to determine a unique group identifier for the group.

[0084] At step S514, the ML-MMSP 510 may determine a mapping of the unique group identifier to a set of member identifiers of the plurality of entities in the group, e.g. mapping of a group ID N to members #1, #2, #3.

[0085] The ML-MMSP 510 may also obtain at least one of: a ML correlation ID, a process ID, and a ML training job ID. The ML-MMSP 510 may generate the ML correlation ID / process ID / ML training job ID. Alternatively the ML correlation ID / process ID / ML training job ID may be provided by the ML-MMSC 508. A ML correlation ID identifies the Federated Learning procedure for training the ML model, this parameter may be included when the service is used for Federated Learning. The ML training job or ML process ID identifies the job/process for training the ML model. For FL use cases the ML correlation ID may comprise one or more ML training job IDs. The ML process ID may also identify the ML inference process.

[0086] At step S514, the ML-MMSP 510 may also determine a mapping of the unique group identifier to the ML correlation ID / process ID / ML training job ID. For example, the group ID of a group consisting of member #1, #2, #3 may map to job#x, job#y or map to correlation ID #z.

[0087] A ML training job ID can be for a given ML task, but the group may comprise entities performing multiple ML tasks, for example multiple training jobs happen by 2 servers in the group for 2 app services (e.g. analytics service #1 and analytics service #2).

[0088] At step S514, the ML-MMSP 510 may generate the unique group identifier for the group randomly, or the unique group identifier may enclose context information indicating one of the mappings referred to above.

[0089] The ML-MMSP 510 is configured to transmit the unique group identifier for delivery to each of the plurality of entities selected for the group. The transmission of the unique group identifier may be direct to the plurality of entities or via another entity (e.g. the ML-MMSC 508). Based on the type of the message transmitted, this may be followed by an ACK/NACK or response.

[0090] For example, at step S516 the ML-MMSP 510 may transmit the unique group identifier for delivery to one or more ML model training service producers 512 and/or at step S518 the ML-MMSP 510 may transmit the unique group identifier for delivery to one or more ML model inference service producers 514.

[0091] The ML-MMSP 510 may be further configured to transmit the identities of the plurality of entities of the group for delivery to at least one of the plurality of entities (e.g. a ML server or FL server) in the group.

[0092] After the inclusion of ML model training service producers 512 to the determined group, at step S520 the ML model training service producers 512 send the trained models with the unique group identifier to the ML-MMSC 508. Whilst not shown in Figure 5, the ML model inference service producer(s) 514 may send inference results with the unique group identifier to the ML training service producer(s) 512 or to the ML-MMSC 508 (depending on the deployment).

[0093] The ML-MMSP 510 may additionally monitor the created group status and in particular the group members status periodically to detect a trigger event for modifying the group. The trigger event may be an entity joining the group; one of the plurality of entities leaving the group; one of the plurality of entities being unavailable to perform the ML operation; an expected or predicted failure to meet the one or more KPIs for the ML operation; one of the plurality of entities being incapable to perform the ML operation; or an expected or predicted failure or congestion of the wireless interface coupled with the at least one of the plurality of the entities.

[0094] The ML-MMSP 510 may monitor the members of a group by subscribing to the ML model repository 504 in order to receive notifications on updates on the members (e.g. entering / leaving the group).

[0095] The ML-MMSP 510 may monitor the members of a group by periodically transmitting a status request to each of the plurality of entities in the group; and receiving a status response from each of the plurality of entities in the group.

[0096] The ML-MMSP 510 may monitor the members of a group configure them to periodically transmit a status update to the network entity. In particular, the messages transmitted at step S516 and S518 comprising the unique group identifier may comprise information to configure the participants to periodically transmit a status update.

[0097] The ML-MMSP 510 may monitor the members of a group by transmitting an on-demand status request to each of the plurality of entities in the group; and receiving a status response from each of the plurality of entities in the group. The on-demand status request may be transmitted based on one or more of:

- on new VAL request or ML member group management request being received for adapting the enablement service;
- ML member change of status or predicted change of status (e.g., FL member moving out of coverage);
- UE mobility or energy or capability changes;
- finding more optimal grouping based on a KPI;
- if the KPI for the enablement service cannot be fulfilled (e.g., confidence level not reached).

[0098] The monitoring performed by the ML-MMSP 510 can also include the constant checking of the group participants' (e.g. FL clients) status within the group and whether the local ML models are aligned or not. This step may include the interaction between the ML-MMSP 510 and the ML training service producer(s) 512 directly (via an API) or between the ML-MMSP 510 and a FL server/aggregator (if the alignment issues are captured in the FL server). Possible misalignment of local ML models may lead to triggering a change in the group members (enter of new members or leave of members with misaligned model outputs).

[0099] In response to detecting a trigger event, the ML-MMSP 510 may update the group to create an updated group of a plurality of entities to perform the ML operation, and modify the mapping so that the unique group identifier is associated with a set of member identifiers

of the plurality of entities in the updated group. That is, the unique group identifier stays the same, but the member list changes.

[0100] The ML-MMSP 510 may to transmit the unique group identifier and identities of members of the updated group for delivery to at least one of the plurality of entities in the updated group. In some embodiments, the ML-MMSP 510 transmits the unique group identifier and identities of members of the updated group for delivery to each of the plurality of entities in the updated group. The transmission of the unique group identifier and identities of members of the updated group may be direct to the plurality of entities or via another entity (e.g. the ML-MMSC 508).

[0101] A first embodiment will now be described with reference to Figure 6, whereby the group functionality is at a GMS (i.e. the ML-MMSP 510 is a SEAL GMS), the ML model consumer 506 is a VAL server (operating as a ML/FL server), and the ML-MMSC 508 is an AI enabler server (it can also be an application data analytics enablement server). In particular, the consumer may be the AI enabler server or the VAL server which can be for example the FL server/aggregator. We refer herein to an AI enabler server (or AI/ML enabler server) as a middleware server for supporting AI/ML services. Such server provides middleware services as NetApps on top of core network, to allow the AI/ML applications to integrate with 3GPP systems. The role of the AI enabler server may comprise the selection and grouping of the FL members or the support for discovery and registration of ML participants. The role of the AI enabler server may include the unified exposure of AI/ML capabilities from the 3GPP core network 106, acting as an application Function (AF), as well as configuration of AI enabler client applications at the mobile devices. In certain implementations the AI enabler server may be deployed as a SEAL application data analytics enablement server as specified in TS 23.436.

[0102] At step S602, the VAL server 506 sends a request to the AI enabler server 508 for supporting an ML/FL workflow. The request may include information relating to a ML operation including any of the examples described herein. For example, the request may include a ML model ID, a list of target UEs, a data requirement, a FL/ML training or inference job description, a list of candidate ML/FL participants (if known), and/or a list of selected ML/FL participants (if the VAL server 506 is a ML/FL aggregator) etc. The AI

enabler server 508 can alternatively receive from the VAL server 506 a request for grouping the FL/ML members.

[0103] At step S604, the AI enabler server 508, based on the received request, determines the need for creating a group consisting of the needed ML/FL members. In case of FL, the FL aggregator can be either the VAL server 506 or the AI enabler server 508, based on the request.

[0104] At step S606, the AI enabler server 508, sends a request to an FL group functionality at the Group Management Server (GMS) 510 to undertake the group management for the ML/FL workflow/process based on the request of step S602. The request transmitted at step S606 may include one or more of: the ML model ID, the service ID or service area or service profile, one or more KPIs for the ML/FL process, a data requirement, the types of ML/FL entities required, whether model training and/or inference is needed, model vendor information, and a training job ID (if known) etc. It should be noted that the FL group functionality can be a new service or function or part of any existing enablement server. The request transmitted at step S606 can alternatively be in the form of subscription request to the GMS 510 (e.g. a subscribe-notify model).

[0105] At step S608, the GMS 510 authorizes the request and at step S610 fetches the ML / FL candidate or available ML/FL members for the given model ID/type/profile or for the given service or service area or for the given training/inference job ID etc. from the ML model repository 504. The GMS 510 may also request and receive the capabilities of ML/FL members at the VAL UEs via GM clients for a given service area. For example the GMS 510 may query all GM clients within the service area on the status of the VAL UEs and their availability/capability to serve as ML/FL members.

[0106] At step S611, the GMS 510 determines a group of a plurality of entities to perform the ML operation either by selecting the ML/FL member or by configuring the group based on the selected ML/FL members by the ML/FL server/aggregator. The GMS 510 generates a unique group identifier (group ID) and may also create and stores a mapping of the generated group ID to a set of ML/FL member IDs within the group. The determination of the group can be alternatively jointly decided by the GMS 510 and the AI enabler server 508 or confirmed by the AI enabler server 508.

[0107] The GMS 510 transmits the group ID (and optionally the group member identities) to the members of the group. This transmission of the group ID may be direct to the plurality of entities in the group. Alternatively, the transmission may be indirect. For example the GMS 510 may inform the AI enabler server 508, and the AI enabler server 508 may inform the ML/FL group members of the group ID. Such a transmission can be in form of request or command to the candidate ML/FL members to serve as selected ML/FL members for the given task (ML operation). Such a transmission may include a training job ID or a correlation ID or an FL process / workflow ID.

[0108] As shown in Figure 6, at step S612 the GMS 510 may transmit the group ID (and optionally the group member identities) to one or more candidate entities (candidate ML/FL members) 602 selected for inclusion in the group. Furthermore, at step S614, the GMS 510 may transmit the group ID (and optionally the group member identities) to an AI enabler client (e.g. a GM client) 606 on one or more UEs selected for inclusion in the group. The AI enabler client is the counterpart of the AI enabler server at the VAL UE side. The role of the AI enabler client is to get configured by the AI enabler server via a standardized API and to interact with the VAL applications (or apps or app clients) to confirm the grouping. The AI enabler client may for example be a SEAL client (as specified in TS 23.434) or an EEC capability (as specified in TS 23.558).

[0109] Whilst not shown in Figure 6, a notification may also be sent (directly or indirectly via the AI enabler server 508) to the ML model repository 504 to store the group ID and mapping to the ML/FL members.

[0110] In an alternative embodiment the GMS 510 may send a request to candidate entities (e.g. clients) to join the group for an ML/FL process. Based on the reply from each candidate the GMS 510 creates a Group ID and optionally a training job and/or ML correlation ID for the FL service.

[0111] At step S616, for a VAL client 604 (which may be a ML or FL client) on a UE 104 selected as ML/FL member, the AI enabler client 606 passes the information relating to the ML operation to the VAL client 604 and may also request its consent/confirmation.

[0112] At step S618, the AI enabler server 508 sends a response to the VAL request transmitted at step S602 indicating the group creation and the information relating to the ML operation. For example the response transmitted at step S618 may comprise one or any

combination of: a ML model ID, an analytics ID, a service ID, group ID, bundle ID, ML correlation ID, a process ID, a ML training job ID, member IDs, and member type information.

[0113] At step S620a and S620b the members of the group (e.g. ML/FL clients) send the locally trained ML model outputs to the VAL server 506 using the group ID. The group ID can be equivalent to a ML correlation ID or a bundled training job ID.

[0114] At step S622, the GMS 510 (FL group functionality) continuously monitors the group status and the members' availability / conditions. Furthermore, the GMS 510 detects a trigger for changing the group (new member joins or existing member leaves or existing member problematic/unavailable). As shown in Figure 6, the initial trigger detection may be performed by the VAL server 506 and communicated at step S621 to the GMS 510 which detects the trigger based on the received communication. Similarly, the initial trigger detection may be performed by the AI enabler server 508 and communicated to the GMS 510 which detects the trigger based on the received communication.

[0115] The GMS 510 then updates the group by changing the mapping of the group ID to the member list (the group ID stays the same, but the member list changes).

[0116] At steps S624 and S626, the GMS 510 sends notifications for the updated group to the involved entities in a similar manner to steps S612 and S614 (either direct or via the AI enabler server 508).

[0117] The GMS 510 or the AI enabler server 508 may also notify the ML model consumer (VAL server) 506 and/or the ML model repository 504 of the change of members of the group.

[0118] A second embodiment will now be described with reference to Figure 7. In EDGEAPP, EAS bundle scenario involves the grouping of EASs with dependencies. In the scenario of EAS grouping for an ML / FL process (e.g. training), considering also EES or other enablers (e.g. EEC) and also the device apps (AC) in the groups, it is possible that an ASP doesn't provide the grouping since the group members may change as part of the AI/ML support service.

[0119] In this example embodiment the ML-MMSC 508 is an EES, the ML-MMSP 510 is a SEAL server, and the ML model consumer 506 is an EAS or an AI enabler server (which may operate as an FL server/aggregator). The EES or AI Enabler 506 triggers the EAS

grouping and utilizes the ML-MMSP 510 (which can be a logical functionality of other enablers e.g. EES or ADAES or AI enabler), and notifies the EASs 702 as well as the ACs 704 to be grouped together. So, the bundle ID is provided by the SA6 functionality and not by the ASP. Such requirement is unique for AI/ML since the grouping is based on the selection of the ML training / inference entities which can change dynamically. The ML-MMSP 510 with the group functionality can be a new AI enabler server, an application data analytics enablement server, a GMS, or even an EES.

[0120] At step S702, the EES 508 together with the EAS (FL/ML server) 506 detects that 1) a bundling of EASs or 2) a bundling of a combination of EAS and ACs, or 3) a bundling of ACs, is required for supporting an ML/FL workflow. Such detection can be based on an EAS request, can be initiated from the EES 508 or can be jointly triggered.

[0121] At step S704, the EES 508 determines the need for grouping the ML/FL members (candidate or selected based on FL server).

[0122] At step S706, the EES 508 sends an ML group management request to the corresponding SEAL server 510 with the ML/FL group functionality. Such request includes the information relating to a ML operation (which may be a FL operation) the examples of which are described herein.

[0123] At step S708, the SEAL server 510 authorizes the request and at step S710 fetches the ML / FL candidate or available ML/FL members from the ML model repository 504 for the given model ID/type/profile, or for the given service or service area, or for the given training/inference job ID etc.

[0124] The ML-MMSP 510 may also request and receive the capabilities of ML/FL members at the VAL UEs via GM clients for a given service area. For example a GMS 510 may query all GMCs within the service area on the status of the VAL UEs and their availability/capability to serve as ML/FL members.

[0125] At step S712, the SEAL server 510 determines the group either by selecting the ML/FL member or by configuring the group based on the selected ML/FL members by the ML/FL server/aggregator 506. The SEAL server 510 generates a group identity and may also create/store a mapping of the generated group ID to a set of ML/FL member IDs within the group. The determination of the group can be alternatively jointly decided or confirmed with the EES 508. The generated group ID can be the “bundle ID” as defined in 3GPP TS 23.558.

[0126] At step S712, the SEAL server 510 transmits the group ID (and optionally the group member identities) to the members of the group. This transmission of the group ID (and optionally the group member identities) may be direct to the plurality of entities in the group. Alternatively, the transmission may be indirect. For example at step S714 the SEAL server 510 may transmit a notification to the EES 508, and the EES 508 may inform the EASs/ACs acting as ML/FL members of the group ID. Such a transmission can be in form of request or command to the candidate ML/FL members to serve as selected ML/FL members for the given task (ML operation). Such a transmission may include a training job ID or a correlation ID or an FL process / workflow ID.

[0127] As shown in Figure 7, at step S716 the SEAL server 510 may transmit the group ID (and optionally the group member identities) to an EEC 706 on a UE 104 on which there is an AC 704 selected for inclusion in the group.

[0128] Whilst not shown in Figure 7, a notification may also be sent (directly or indirectly via the EES 508) to the ML model repository 504 to store the group ID and mapping to the ML/FL members.

[0129] At step S718, for UEs selected as ML/FL members, the EEC passes the information relating to the ML operation to the AC 704 and may also request its consent/confirmation.

[0130] At step S720, the SEAL server 510 transmits a response to the EES 508 indicating the group creation and the information relating to the ML operation.

[0131] Alternatively, if the SEAL server 510 is not directly informing the EAS(s), at step S722 the EES 508 may notify the EAS(s) 702 acting as ML/FL member with the information relating to the ML operation.

[0132] Thus in the first embodiment the FL group functionality is an enhanced capability of SEAL GMS, and the consumer is the AI enabler server or the VAL server who can be for example the FL server/aggregator. In the second embodiment, the FL group functionality is applicable for EDGE scenarios where the groups comprise EASs and/or ACs, and the role of the EES is to coordinate the group management (either by jointly supporting the group management or by utilizing SEAL capability and exposing to the EASs).

[0133] In embodiments of the present disclosure, an entity selected for a group may for example be a FL client, a FL server, a FL aggregator or collaborator, a ML client, or a ML server.

[0134] We refer herein to a FL client as an entity which is serving as a ML model training service producer 512 when FL is used. The FL client(s) provide the local ML model outputs to the FL server/aggregator. We refer herein to a FL server as an entity which is configuring and globally training the model based on the local outputs from FL clients. A FL aggregator or collaborator can be similar to FL server, but in certain scenarios (e.g. hierarchical FL), the aggregator or collaborator may only aggregate the local ML model trained data from the FL clients/FL server. We refer herein to a ML server as an AI/ML endpoint which can be an application server providing the ML model or configuring and training the ML model. We refer herein to a ML client as an AI/ML endpoint which is configured by the AI/ML server to perform an ML operation. For example an ML client is at the UE side and receives the ML training output from the ML server, and based on this it uses the output as input for ML model inference (e.g. based on UE local data).

[0135] As shown in the example of Figure 6, a ML client or FL client may be an application running on a UE 104. As shown in the example of Figure 7, a ML client or FL client may be an application client on a UE 104, or an Edge Application Server (EAS).

[0136] Figure 8 illustrates an example of a processor 800 in accordance with aspects of the present disclosure. The processor 800 may be an example of a processor configured to perform various operations in accordance with examples as described herein. The processor 800 may include a controller 802 configured to perform various operations in accordance with examples as described herein. The processor 800 may optionally include at least one memory 804, which may be, for example, an L1/L2/L3 cache. Additionally, or alternatively, the processor 800 may optionally include one or more arithmetic-logic units (ALUs) 806. 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).

[0137] The processor 800 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 800)) 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).

[0138] The controller 802 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 800 to cause the processor 800 to support various operations in accordance with examples as described herein. For example, the controller 802 may operate as a control unit of the processor 800, generating control signals that manage the operation of various components of the processor 800. These control signals include enabling or disabling functional units, selecting data paths, initiating memory access, and coordinating timing of operations.

[0139] The controller 802 may be configured to fetch (e.g., obtain, retrieve, receive) instructions from the memory 804 and determine subsequent instruction(s) to be executed to cause the processor 800 to support various operations in accordance with examples as described herein. The controller 802 may be configured to track memory address of instructions associated with the memory 804. The controller 802 may be configured to decode instructions to determine the operation to be performed and the operands involved. For example, the controller 802 may be configured to interpret the instruction and determine control signals to be output to other components of the processor 800 to cause the processor 800 to support various operations in accordance with examples as described herein. Additionally, or alternatively, the controller 802 may be configured to manage flow of data within the processor 800. The controller 802 may be configured to control transfer of data between registers, arithmetic logic units (ALUs), and other functional units of the processor 800.

[0140] The memory 804 may include one or more caches (e.g., memory local to or included in the processor 800) or other memory, such as RAM, ROM, DRAM, SDRAM, SRAM, MRAM, flash memory, etc. In some implementations, the memory 804 may reside within or

on a processor chipset (e.g., local to the processor 800). In some other implementations, the memory 804 may reside external to the processor chipset (e.g., remote to the processor 800).

[0141] The memory 804 may store computer-readable, computer-executable code including instructions that, when executed by the processor 800, cause the processor 800 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 802 and/or the processor 800 may be configured to execute computer-readable instructions stored in the memory 804 to cause the processor 800 to perform various functions. For example, the processor 800 and/or the controller 802 may be coupled with or to the memory 804, the processor 800, the controller 802, and the memory 804 may be configured to perform various functions described herein. In some examples, the processor 800 may include multiple processors and the memory 804 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.

[0142] The one or more ALUs 806 may be configured to support various operations in accordance with examples as described herein. In some implementations, the one or more ALUs 806 may reside within or on a processor chipset (e.g., the processor 800). In some other implementations, the one or more ALUs 806 may reside external to the processor chipset (e.g., the processor 800). One or more ALUs 806 may perform one or more computations such as addition, subtraction, multiplication, and division on data. For example, one or more ALUs 806 may receive input operands and an operation code, which determines an operation to be executed. One or more ALUs 806 may 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 806 may support logical operations such as AND, OR, exclusive-OR (XOR), not-OR (NOR), and not-AND (NAND), enabling the one or more ALUs 806 to handle conditional operations, comparisons, and bitwise operations.

[0143] The processor 800 may support wireless communication in accordance with examples as disclosed herein. The processor 800 may be configured to or operable to support a means to obtain information relating to the ML operation; create a group of a plurality of

entities to perform the ML operation, based on the information relating to the ML operation; determine a unique group identifier for the group; and transmit the unique group identifier for delivery to each of the plurality of entities.

[0144] Figure 9 illustrates an example of a NE 900 in accordance with aspects of the present disclosure. The NE 900 may include a processor 902, a memory 904, a controller 906, and a transceiver 908. The processor 902, the memory 904, the controller 906, or the transceiver 908, 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. The NE 900 may correspond to the ML-MMSP 510 and perform the functionality of the ML-MMSP 510 described herein.

[0145] The processor 902, the memory 904, the controller 906, or the transceiver 908, 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.

[0146] The processor 902 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 902 may be configured to operate the memory 904. In some other implementations, the memory 904 may be integrated into the processor 902. The processor 902 may be configured to execute computer-readable instructions stored in the memory 904 to cause the NE 900 to perform various functions of the present disclosure.

[0147] The memory 904 may include volatile or non-volatile memory. The memory 904 may store computer-readable, computer-executable code including instructions when executed by the processor 902 cause the NE 900 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such as the memory 904 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.

[0148] In some implementations, the processor 902 and the memory 904 coupled with the processor 902 may be configured to cause the NE 900 to perform one or more of the functions described herein (e.g., executing, by the processor 902, instructions stored in the memory 904). For example, the processor 902 may support wireless communication at the NE 900 in accordance with examples as disclosed herein. The NE 900 may be configured to support a means for obtaining information relating to the ML operation; creating a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation; determining a unique group identifier for the group; and transmitting the unique group identifier for delivery to each of the plurality of entities.

[0149] The controller 906 may manage input and output signals for the NE 900. The controller 906 may also manage peripherals not integrated into the NE 900. In some implementations, the controller 906 may utilize an operating system such as iOS®, ANDROID®, WINDOWS®, or other operating systems. In some implementations, the controller 906 may be implemented as part of the processor 902.

[0150] In some implementations, the NE 900 may include at least one transceiver 908. In some other implementations, the NE 900 may have more than one transceiver 908. The transceiver 908 may represent a wireless transceiver. The transceiver 908 may include one or more receiver chains 910, one or more transmitter chains 912, or a combination thereof.

[0151] A receiver chain 910 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 910 may include one or more antennas for receiving the signal over the air or wireless medium. The receiver chain 910 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 910 may include at least one demodulator configured to demodulate the received signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal. The receiver chain 910 may include at least one decoder for decoding and/or processing the demodulated signal to receive the transmitted data.

[0152] A transmitter chain 912 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 912 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 912 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 912 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0153] Figure 10 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 performing the functionality of a ML-MMSP 510. In some implementations, the NE may execute a set of instructions to control the function elements of the NE to perform the described functions.

[0154] At 1002, the method may include obtaining information relating to a ML operation. The operations of 1002 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1002 may be performed by a NE as described with reference to Figure 9.

[0155] At 1004, the method may include creating a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation. The operations of 1004 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1004 may be performed by a NE as described with reference to Figure 9.

[0156] At 1006, the method may include determining a unique group identifier for the group. The operations of 1006 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1006 may be performed a NE as described with reference to Figure 9.

[0157] At 1008, the method may include transmitting the unique group identifier for delivery to each of the plurality of entities. The operations of 1008 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1008 may be performed a NE as described with reference to Figure 9.

[0158] Figure 11 illustrates how the components described herein may be coupled to the wireless communication system 100 comprising a SA6 enablement layer.

[0159] In particular, Figure 11 illustrates an architecture for supporting vertical FL, the architecture comprising the relevant AI-enabled analytics entities in the 5G core (e.g. the NWDAF), and the enablement layer (in particular the SEAL ADAE layer or an AI enabler) which sits on top of the core network 106 and operations and management (OAM) function as AFs. Such an enablement layer spans across the server side as well as the device side (named as "enabler client"), and has interfaces with the application specific layer both at the server side and client side. The AI enabler servers and AI enabler clients may correspond to those described in 3GPP TS 23.434, TS 23.558, and TS 23.222.

[0160] 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.

[0161] 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.

CLAIMS

1. A network entity for grouping a plurality of entities for participating in a machine learning (ML) operation, wherein the plurality of entities are coupled to a wireless communication system, the network 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:

obtain information relating to the ML operation;

create a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation;

determine a unique group identifier for the group; and

transmit the unique group identifier for delivery to each of the plurality of entities.

2. The network entity of claim 1, wherein the information relating to the ML operation comprises one or any combination of: (i) an ML model identifier; (ii) a service identifier or service area or service profile; (iii) at least one target user equipment identifier; and (iv) an analytics identifier.

3. The network entity of claim 1 or 2, wherein the information relating to the ML operation comprises one or any combination of: (v) one or more KPIs for the ML operation; (vi) a data requirement for the ML operation; (vii) one or more entity types; (viii) an indication as to whether model training and/or inference is needed; (ix) model vendor information; and (x) a training job ID.

4. The network entity of any preceding claim, wherein the at least one processor is configured to cause the network entity to:

transmit a request message to a ML model repository, the request message comprising the information relating to the ML operation; and

receive a response message from the ML model repository, the response message comprising a list of candidate entities to perform the ML operation.

5. The network entity of claim 4, wherein the at least one processor is configured to cause the network entity to:

select the plurality of entities from the list of candidate entities to create the group.

6. The network entity of any preceding claim, wherein the at least one processor is configured to cause the network entity to determine the unique group identifier randomly.

7. The network entity of any preceding claim, wherein the at least one processor is configured to:

determine a mapping of the unique group identifier to a set of member identifiers of the plurality of entities in the group; and

store the mapping in the at least one memory.

8. The network entity of claim 7, wherein the at least one processor is configured to:

detect a trigger event for modifying the group;

based on the trigger event, update the group to create an updated group of a plurality of entities to perform the ML operation; and

modify the mapping so that the unique group identifier is associated with a set of member identifiers of the plurality of entities in the updated group.

9. The network entity of claim 8, wherein the trigger event is:

an entity joining the group;

one of the plurality of entities leaving the group;

one of the plurality of entities being unavailable to perform the ML operation;

an expected or predicted failure to meet the one or more KPIs for the ML operation;

one of the plurality of entities being incapable to perform the ML operation; or

an expected or predicted failure or congestion of a wireless interface coupled with the at least one of the plurality of the entities.

10. The network entity of claim 8 or 9, wherein the at least one processor is configured to monitor a status of the plurality of entities in the group to detect the trigger event.

11. The network entity of claim 10, wherein the at least one processor is configured to:

periodically transmit a status request to each of the plurality of entities in the group; and

receive a status response from each of the plurality of entities in the group.

12. The network entity of claim 10, wherein the at least one processor is configured to configure the plurality of entities in the group to periodically transmit a status update to the network entity.

13. The network entity of any of claims 8 to 12, wherein the at least one processor is configured to transmit the unique group identifier and identities of members of the updated group for delivery to at least one of the plurality of entities in the updated group.

14. The network entity of any preceding claim, wherein the network entity is a Service Enabler Architecture Layer (SEAL) group management server.

15. The network entity of claim 14, wherein the plurality of entities in the group comprise one or any combination of: at least one FL client, at least one FL server, at least one FL aggregator or collaborator, at least one ML client, and at least one ML server.

16. The network entity of any of claims 1 to 13, wherein the network entity is Service Enabler Architecture Layer (SEAL) group management server, an AI enabler server or an Application Data Analytics Enabler Server or an Edge Enabler Server.

17. The network entity of claim 14, wherein the plurality of entities in the group comprises at least one Edge Application Server and/or at least one UE application client.

18. The network entity of any preceding claim, wherein the at least one processor is configured to transmit the identities of the plurality of entities of the group for delivery to at least one of the plurality of entities in the group.

19. A method for grouping a plurality of entities for participating in a machine learning (ML) operation, wherein the plurality of entities are coupled to a wireless communication system, the method comprising:

obtaining information relating to the ML operation;

creating a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation;

determining a unique group identifier for the group; and

transmitting the unique group identifier for delivery to each of the plurality of entities.

20. A processor for grouping a plurality of entities for participating in a machine learning (ML) operation, the processor comprising:

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

obtain information relating to the ML operation;

create a group of a plurality of entities to perform the ML operation, based on the information relating to the ML operation;

determine a unique group identifier for the group; and

output the unique group identifier for delivery to each of the plurality of entities.