

ENABLING TRANSFER LEARNING FOR USER EQUIPMENT ANALYTICS IN A WIRELESS COMMUNICATION NETWORK

TECHNICAL FIELD

[0001] The subject matter disclosed herein relates generally to the field of implementing enabling transfer learning for user equipment analytics in a wireless communication network. This document defines a user equipment entity for wireless communication, a processor for wireless communication, a network entity for wireless communication, a method performed by a user equipment entity, a method performed by a processor, and a method performed by a network entity.

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] Transfer Learning (TL) is a technique in machine learning where a model trained on one task is used as the starting point for a model on a second task. This can be useful when the second task has similarities with the first task, or when there is limited data available for the second task. By using the learned features from the first task as a starting point, the model can learn more quickly and effectively on the second task. This can also

help to prevent overfitting, as the model will have already learned general features that are likely to be useful in the second task.

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] There is provided a user equipment ‘UE’ entity for wireless communication, comprising: at least one memory; and at least one processor coupled with the at least one memory and configured to cause the UE entity to: determine a requirement for one or more pre-trained machine learning ‘ML’ models for transfer learning, the transfer learning for a UE analytics event; obtain, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; select a pre-trained ML model from the one or more available pre-trained ML models; and train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0006] There is further provided a processor for wireless communication, comprising: at least one controller coupled with at least one memory and configured to cause the processor to: input a requirement for one or more pre-trained ML models for transfer

learning, the transfer learning for a UE analytics event; input, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; output, a selection of a pre-trained ML model from the one or more available pre-trained ML models; and train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0007] There is further provided a network entity for wireless communication, 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 a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtain a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; send a first response to the UE entity, wherein the first response comprises the first information.

[0008] There is further provided a method performed by a UE entity, comprising: determining a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtaining, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; selecting a pre-trained ML model from the one or more available pre-trained ML models; and training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0009] There is further provided a method performed by a processor, comprising: inputting a requirement for one or more pre-trained machine learning 'ML' models for transfer learning, the transfer learning for a UE analytics event; inputting, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML

models trained for one or more network analytics events; outputting a selection of a pre-trained ML model from the one or more available pre-trained ML models; and training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0010] There is further provided a method performed by a network entity, comprising: receiving a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtaining a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; and sending a first response to the UE entity, wherein the first response comprises the first information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 illustrates an example of a wireless communications system in accordance with aspects of the present disclosure.

[0012] Figure 2 illustrates an example of high-level transfer learning operation in accordance with aspects of the present disclosure.

[0013] Figure 3 illustrates an example of a high-level architecture in accordance with aspects of the present disclosure.

[0014] Figure 4 illustrates an example of transfer learning for UE QoS analytics in accordance with aspects of the present disclosure.

[0015] Figure 5 illustrates an example of transfer learning for VAL session performance analytics in accordance with aspects of the present disclosure.

[0016] Figure 6 illustrates an example of a user equipment (UE) 600 in accordance with aspects of the present disclosure.

[0017] Figure 7 illustrates an example of a processor 700 in accordance with aspects of the present disclosure.

[0018] Figure 8 illustrates an example of a network equipment (NE) 800 in accordance with aspects of the present disclosure.

[0019] Figure 9 illustrates a flowchart of a method performed by a UE in accordance with aspects of the present disclosure.

[0020] Figure 10 illustrates a flowchart of a method performed by a processor in accordance with aspects of the present disclosure.

[0021] Figure 11 illustrates a flowchart of a method performed by a NE in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

[0022] In the third-generation partnership project (3GPP) the employment of artificial intelligence (AI) has been discussed in different domains (RAN, core, service enabler layer) either as a tool to optimize existing features and procedures by making them more intelligent, or for assisting AI-enabled apps to communicate via the mobile communications system.

[0023] The work of the 3GPP in core and application layers has been performed in 3GPP SA2 which started from Rel-16 to investigate AI/ML support. This focused on two main aspects: 1) AI/ML support in the networks data analytics function (NWDAF); and 2) network assistance for AI/ML services.

[0024] Regarding the first main aspect, network analytics and AI/ML is deployed in the 5G core network via the introducing of NWDAF which considers the support of various analytics types that can be distinguished using different Analytics IDs, e.g., “UE Mobility”, “NF Load”, etc. as elaborated in the 3GPP Technical Specification (TS) 23.288, titled “Architecture enhancements for 5G System to support network data analytics services”. Each NWDAF may support one or more Analytics IDs and may have the role of: (i) AI/ML inference called NWDAF AnLF, or (ii) AI/ML training called NWDAF MTLF or (iii) both. For federated learning (FL) use cases, in 3GPP Release 18, 3GPP defined federated learning amongst different NWDAF MTLFs where ML model training is running in multiple local MTLFs.

[0025] Regarding the second main aspect, as part of AIMLSys, enhancements to 5GC (as specified in the 3GPP TS 23.501 clause 5.46, titled “System Architecture for the 5G system”) are specified for assisting the AI/ML operations in the application layer (between one or more AI/ML users and AI/ML server). In this direction, a network exposure function (NEF) may assist the AI/ML application server in scheduling available UE(s) to participate in the AI/ML operation (e.g. Federated Learning). Also, 5GC may assist the selection of UEs to serve as FL clients, by providing a list of target member UE(s), then subscribing to the NEF to be notified about the subset list of UE(s) (i.e. list of candidate UE(s)) that fulfil certain filtering criteria.

[0026] The 3GPP SA4 continues to study AI and ML for Media (see 3GPP Technical Report (TR) 26.927 titled, “Study on Artificial Intelligence and Machine Learning in 5G media services”). This study aims to identify relevant interoperability requirements and implementation constraints of AI/ML in 5G media services. This study includes media-based AI/ML use cases and architecture considerations related to media services.

[0027] The 3GPP SA5 in Rel-17 has also provided a work on AI/ML management (see 3GPP TS 28.105 titled, “Management and orchestration; Artificial Intelligence/Machine Learning management”) focusing on AI/ML capabilities (e.g., ML training MnS/MF) at the operations administration and maintenance (OAM) side. The 3GPP TS 28.105 specifies the AI/ML management capabilities and services for 5GS where AI/ML is used, including management and orchestration (e.g., MDA, see the 3GPP TS 28.104, titled “Management and orchestration; management data analytics (MDA)). In this work, an ML training Function (MLT) playing the role of ML training MnS producer is introduced which may consume various data from 5GS for ML training purpose and provides the training outputs to other management functions in OAM.

[0028] The 3GPP SA6 in Rel-19 continues to study an AIML Enabler (which can be logically placed in ADAES or a new SEAL server) to support AI/ML services via the service enablement layer. This study is discussed in the 3GPP TR 23.700-82 titled, “*Study on Application Layer support for AI/ML services*”.

[0029] In 3GPP TR 23.700-82 (AIMLAPP study in Rel-19 of 3GPP SA6), a key issue is presented in respect of support for transfer learning in 3GPP systems. The key issue that arises is how to support transfer learning at application enablement layers.

[0030] In 3GPP TS 22.261 titled "Service requirements for the 5G system" at clause 6.40 it is stated that the 5G system needs to support at least three types of AI/ML operations: AI/ML operation splitting between AI/ML endpoints; Distributed/Federated Learning over 5G system; and AI/ML model/data distribution and sharing over 5G system.

[0031] In relation to AI/ML model/data distribution and sharing over the 5G system, it is expected to let the AI/ML consumers use an ML model (out-of-a-set of candidate models available in a NW endpoint) for better adapting to task and environment variations.

[0032] Nevertheless, in practical deployments involving AI/ML operations, the number of available models is not that large to cover all the possible environmental conditions in a network. Therefore out-of-the-box usage of the model will imply a lack of accuracy/precision during the inference stage unless a model fine tuning stage is present to ensure that the baseline model is quickly adapted to the current conditions using lesser amount of data (than the one used for training of the baseline model). On that note, Transfer learning (TL) is conceived as a technique used to tackle the lack of labelled data for training a model which is the first limitation that is faced when (re-)training a model. In TL, the training of a model (to solve a particular task) is carried out using information from a "similar" domain in which enough information is available (the so-called source domain). With TL, some parts of the model trained in the source domain are fine-tuned with scarce information available in the target domain. This way, the model reuses the understanding of the structure of the problem and does not have to learn from scratch all the low-level features/structure of the problem: it will only have to learn the higher-level structures/relationships which usually requires less labelled data. TL assumes a baseline model is already available (in a NW endpoint) and can be fine-tuned to quickly perform the same or similar tasks (e.g., prediction, classification, etc.) in a target domain with lesser amount of training data.

[0033] Considering firstly that: in 3GPP TS 23.288 clauses 6.2B.5, 6.2B.6 and 6.2B.7 some operations to allow the storage/retrieval of ML models to/from the Analytical Data

Repository Function (ADRF) are available to be consumed by a network function containing an instance of the Model Training Logical Function (MTLF); and as from clause 5.3 in 3GPP TS 23.436 (titled “Functional architecture and information flows for application data analytics enablement service”), the internal architecture of ADAE (Application Data Analytics Enabler) can include an instance of ADRF, the so-called A-ADRF (Application layer - Analytical Data Repository Function), it is possible to enable inter- and intra-VAL domain transfer learning scenarios.

[0034] As mentioned above, Transfer Learning is used to support an entity to ease the training process (if not possible or if the data and time to train is not sufficient). When applying Transfer Learning (TL) to features in 5G and beyond systems, there can be different use cases which can be assumed for the operations to be impacted.

[0035] A first use-case is where TL is employed in AI/ML enabled analytics between network analytics functions in the Core (between NWDAFs / MTLFs).

[0036] A further use-case is where TL is employed in AI/ML enabled analytics between SEAL ADAESs (ADAES as defined in SA6, 3GPP TS 23.436)

[0037] A further use-case is where TL is employed between an AF and a NF, for instance in cross domain analytics of AI-assisted operations.

[0038] A further use-case is where TL is employed to assist AI/ML operations between VAL servers or Afs.

[0039] A further use-case is where TL is employed to assist AI/ML operations between AF/VAL servers and VAL UEs or among VAL UEs.

[0040] A particularly relevant use case where the network/server trains the ML model for predicting the network QoS for a certain cell or network area and then this is used at the UE side to predict the QoS for a particular session when traversing the cell / area of interest.

[0041] For TL enablement in 3GPP systems focusing more on application layer, a problem that can be identified is how to deal with the scenario where the VAL UE utilizes

the pre-trained model to perform local training with minimum processing. Furthermore, a related problem relates to the impact at the UE side and the interfaces.

[0042] This disclosure herein provides a mechanism to allow local UE performance prediction (e.g., UE QoS prediction) with the support of AI/ML, by utilizing transfer learning from ML tasks related to network QoS / performance for a target area of interest (e.g., cell area). Such solutions enable the benefits of transfer learning as hereinbefore discussed, within 3GPP systems.

[0043] Aspects of the present disclosure are described in the context of a wireless communications system.

[0044] 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.

[0045] 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 signalling, transmit signalling) over a Uu interface.

[0046] 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 associated with the same or different radio access technologies may overlap, but the different geographic coverage areas may be associated with different NE 102.

[0047] 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.

[0048] 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 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.

[0049] The NE 102 may comprise application enablement or vertical enablement or edge enablement entities (SEAL, EDGEAPP, CAPIF) which are specified in 3GPP SA6. 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, N2, 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 or 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).

[0050] 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.

[0051] The CN 106 may communicate with a packet data network over one or more backhaul links (e.g., via an S1, N2, N2, 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).

[0052] 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.

[0053] 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., $\mu=1$) may be associated with a second subcarrier spacing (e.g., 30 kHz) and a normal cyclic prefix. A third numerology (e.g., $\mu=2$) 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., $\mu=3$) may be associated with a fourth subcarrier spacing (e.g., 120 kHz) and a normal cyclic prefix. A fifth numerology (e.g., $\mu=4$) may be associated with a fifth subcarrier spacing (e.g., 240 kHz) and a normal cyclic prefix.

[0054] 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.

[0055] 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., $\mu=0$, $\mu=1$, $\mu=2$, $\mu=3$, $\mu=4$) 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., $\mu=0$) associated with a first subcarrier spacing (e.g., 15 kHz) may be used interchangeably between subframes and slots.

[0056] 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.

[0057] 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., $\mu=0$), which includes 15 kHz subcarrier spacing; a second numerology (e.g., $\mu=1$), which includes 30 kHz subcarrier spacing; and a third numerology (e.g., $\mu=2$), 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., $\mu=2$), which includes 60 kHz subcarrier spacing; and a fourth numerology (e.g., $\mu=3$), which includes 120 kHz subcarrier spacing.

[0058] Figure 2 illustrates an example of high-level transfer learning operation 200 in accordance with aspects of the present disclosure.

[0059] Transfer learning starts with a model 220 that has previously been trained for a certain task 210 using a large set of data. Frequently trained on extensive datasets, this model 220 has identified general features and patterns relevant to numerous related jobs. This model 220 is known as a pre-trained model or pre-trained ML model.

[0060] The model 220 that has been pre-trained is also known as the base model. It is made up of layers that have utilized the incoming data to learn hierarchical feature representations. This base model forms the starting point for re-training for a second ML task 250.

[0061] In the pre-trained model 220, a set of layers is found that capture generic information and knowledge 230 relevant to the new task 250 as well as the previous task 210. Because they are prone to learning low-level information, these layers are frequently found near the top of the trained ML network.

[0062] Using the dataset from the new task/challenge 250 to retrain the chosen layers is a procedure known as fine-tuning. The goal is to preserve the knowledge from the pre-training while enabling the new model 240 to modify its parameters to better suit the demands of the current assignment.

[0063] Figure 3 illustrates an example 300 of a high-level architecture in accordance with aspects of the present disclosure. The example 300 shows a first VAL UE 310 having an associated Application/Enabler Client (i.e., MTME functionality) 312. The first VAL UE 310 interfaces with an AI enabler server (i.e., an application function) 320. The AI enabler server 320 interfaces with an ML model repository/ML model providers 330, and a 5GC 350 and gNB(s) 360. The first VAL UE 310 is within coverage of the gNB(s) 360. Furthermore, a second VAL UE 340 is within coverage of the gNB(s) 360. The second

VAL UE 340 has an associated application/enabler client (i.e., MTME functionality) 342. The first VAL UE 310 and the second VAL UE 340 are able to interface over sidelink. The application/enabler clients 312, 342 may be able to interface with the ML model repository/ML model providers 330. For completeness, the AI Enabler Server 320 may be equivalent to an AI/ML Enablement Server as reported in 3GPP TR 23.700-82, or an ADAES as specified in 3GPP TS 23.436 or any other SEAL or edge enablers as defined in 3GPP TS 23.434 and 3GPP TS 23.558. Such an AI Enabler Server may include or can be represented as an AF functionality. AF is defined in 3GPP TS 23.501 as an application function which is part of 5G system (part of the service-based 5G core architecture); however such an AF can be either deployed at the MNO domain or at a trusted or non-trusted 3rd party (e.g., vertical) which has established agreement with the MNO for consuming core network services. The high-level operational procedures of the example 300 will now be described.

[0064] In a first step 301, the UE#1 310 wants to predict the QoS for a Uu session in a given cell area. The UE #1 310 has local performance data for this area and service, and has also a ML training capability. However, the UE#1 310 has limited processing capacity to perform extensive training. The UE#1 310 does not want to obtain the trained data to do inference, but wants to train the ML model by itself with its own data. The UE#1 310 can act on behalf a group of UEs (as a group lead) and may be able to delegate /offload/split some training at further UEs of the group. This first step 301 can be expressed differently as detecting the need for transfer learning (TL) for local analytics.

[0065] In certain embodiments, the criteria for triggering the use of TL can be the energy constraints or the high expected energy utilization for the training process (i.e., if the training were to be performed from scratch). In certain embodiments, the criteria for triggering the use of TL can be the processing latency and load for training the ML model from scratch and/or the lack of data for doing this.

[0066] In a further step 302, the UE#1 310 via the enabler client 312 requests the AI Enabler Server (AF) 320 to find sources and to get a pre-trained ML model for any ML model task related to QoS / performance analytics for the target area of interest (i.e., a cell or list of cells). The sources can be either at the server side or at the network side (from one

or more network operators). This step 302 can be expressed differently as requesting the availability/information of pre-trained models for UE/group analytics.

[0067] In a further step 303, the AI Enabler Server (AF) 320 fetches the pre-trained models for all analytics IDs related to performance prediction in the given cell area (i.e., QoS sustainability, network congestion, UE congestion, VAL server performance, Edge Load / Performance analytics).

[0068] In a further step 304, the AI Enabler Server (AF) 320 sends to the UE#1 310 information of one or more pre-trained models as candidates (this may include the features, data set requirements, environment, context information). The list may be acquired by multiple domains (edge, cloud, PLMN1, PLMN2) - hence it may include all the registered models from multiple vendors or operators. This step 304 can be expressed differently as providing a list of pre-trained model information for all relevant analytics IDs.

[0069] In a further step 305, the UE#1 310 via the Enabler Client 312 selects one of the candidate pre-trained models and downloads the pre-trained model using the repository 330 address (or via the enabler client 320).

[0070] The criteria for selecting the best model as a pre-trained ML model are one or more of the following: similarity of analytics event; similarity of context information for the model (features, data sets); assistance information from AI Enabler 320 on the rating/evaluation of the model as applicable to the analytics ID; whether the listed pre-trained model has been used in the past for analytics task and under which conditions (time of the day, load); whether the listed pre-trained model has been tested in a simulation platform (e.g., digital twin) for analytics task and under certain hypothetical what-if scenarios (time of the day, load); expected time, energy and/or processing drain to train based on pre-trained model status; the model which optimizes a given network utility function which can be based on KPIs (i.e., latency, energy, rate) and pricing values; whether the pre-trained model resides at the edge or cloud or a given domain with higher preference/trustfulness (e.g. from MNO this can be easier trusted); or a combination of the above.

[0071] In a further step 306, the UE#1 310 trains the model using the pre-trained model as a basis (i.e., as a base ML model) and predicts the QoS for the session for the given area and time. The UE#1 310 may then use the re-trained ML model to derive UE analytics.

[0072] In a further step 307, the UE#1 310 may also distribute via side-link to other UEs 340 in the group to split the training process based on the pre-trained model. Alternatively, the UE#1 310 may indicate to the AI Enabler Server 320 to provide the pre-trained model information to all the UEs 340 in the group or to broadcast this information.

[0073] Figure 4 illustrates an example 400 of transfer learning for UE QoS analytics in accordance with aspects of the present disclosure. In this example 400, the pre-trained ML model is used for QoS prediction at the UE side for a Uu/ PC5 session. Such QoS prediction happens at the application enabler client using a ML model trained for network QoS sustainability analytics.

[0074] The example 400 shows a first UE#1 410 having an AI enabler client 412 and VAL client 414. Also shown is an AI enabler server 420, an ML model repository 430, sources of pre-trained models 432 and a second UE#2 440. The AI Enabler Client 412 is connected to the AI Enabler Server 420. The AI Enabler Server 420 is connected to either or both of the ML model repository 430 or the at least one source of ML models 432 (which may be OAM, AF, NWDAF, or a 3rd party ML server). The various messaging flows will now be described.

[0075] In a first step 401, the VAL client 414 in VAL UE#1 410 requests from AI Enabler Client 412 to support training a ML model for deriving local analytics for predicting QoS for a given session (Uu or PC5) for a given area and time of interest (i.e., cell x, time window y). This step 401 can be expressed as identifying the need for using transfer learning for UE QoS prediction.

[0076] In a further step 402, the AI Enabler Client 412 having identified that the UE 410 is not capable of locally training the ML model from scratch due to time/data/processing/energy limitations, sends a request to AI enabler server 420 to find the available pre-trained ML models which can be applicable for the given analytics task (i.e., the local UE QoS analytics for either Uu or PC5 session). This request includes one or

more of the following parameters: Analytics ID; Requestor ID (i.e., app ID, VAL client ID, UE ID, group ID); app session /service ID or profile or session type or vertical identifier / consumer identifier to help in identifying the key performance indicators, KPIs, for which the analytics service applies; filter information for the request; requirements related to the required features/datasets for the pre-trained ML models; area of applicability and time of validity for the request; preferred confidence level; cause for requesting pre-trained ML models (energy, processing limitations, time or lack of data); requirement for when to send the requested information (for example only in good channel conditions, low load); whether analytics apply for Uu or PC5 session and/or transmission mode (i.e., unicast, broadcast, groupcast); one or more PLMN/NPN IDs and vendor IDs for which the request is allowable to be made (i.e., based on UE capabilities and service level agreements, SLAs).

[0077] In a further step 403a, the AI Enabler Server 420 collects information for base ML models for analytics related to QoS prediction (e.g., QoS sustainability analytics) from the repository 430 or from the sources 432 directly (e.g., from NWDAF or ADRF, AF). If the source 432 is NWDAF or ADRF, this requires the enhancement of core interfaces (via NEF or directly) to allow for exposing the model information from NWDAF or ADRF. This step 403a can be expressed as discovering pre-trained models.

[0078] The collection for multi-operator scenarios (i.e., if UE 410 supports dual radio) may require collecting information from multiple NWDAF/ADRF/AF corresponding to different mobile network operators (MNOs).

[0079] The information collected may include one or more of the following parameters: model ID and/or profile; analytics ID (or IDs) for which the model has been used or matched; context information including features, environment, data set requirements, type of ML methods/types used (e.g., reinforcement learning, federated learning); vendor ID and vendor interoperability information; permissions for exposure and being used as a pre-trained ML model (i.e., excluded list of users / services / application types allowed to use it); allowable CRUD operations; URL/address to fetch the model and optionally an ML repository ID/address to fetch the model; whether the model can be directly fetched by the UE 410 or via enabler 420; area and time of validity; source 432 (ML provider)

requirement for when the model is available for download (i.e., during low load scenarios or good channel conditions).

[0080] In a further step 403b, the AI Enabler Server 420 optionally filters the received list of candidate pre-trained models based on the UE request. This step includes also possible processing for formatting the message and sending as a bulk message including a structured list of sources, pre-trained ML models, and candidate partner analytics IDs. Such analytics IDs which are different from UE QoS prediction analytics are defined herein as “partner analytics ID”. For completeness, a partner analytics service or event or ID may be an analytics service/event/ID which is paired with the analytics service/event/ID for which the transfer learning applies. For example, a partner analytics event may be the event for which the base model is originally trained and based on this it is used for the target analytics event (UE-driven analytics).

[0081] In a further step 404, the AI Enabler Server 420 sends the collected information to the VAL UE #1 410 (i.e., to enabler client 412) as a response to step 402. This message may include the information from step 403a or a subset of it or any abstraction of it. The information can be bundled in one message or can be sent as independent messages from either the AI Enabler Server 420 and/or the one or more sources 432 of the base models.

[0082] In a further step 405, the AI Enabler Client 412 interacts with the VAL client 414 to select one of the models for the analytics task (this can be done jointly or at the enabler client 412 or at the VAL client 414 with the recommendation from enabler client 412).

[0083] In a further step 406, the AI Enabler client 412 requests to download the model from the enabler server 420 or from the repository 430 or from the source 432 of the ML model directly.

[0084] In a further step 407, the AI Enabler client 412 then receives the requested model for the partner analytics ID/ task ID.

[0085] In a further step 408, the VAL UE#1 410, if the training involves more UEs 440 (in group-based communications), further sends a request to further UEs 440 to perform part of the ML model training and also sends the ML model info (ID/address) to allow the

other UEs 440 to download. Or the VAL UE#1 410 may also send the model itself via side-link. This step 408 can be expressed as exchanging pre-trained model with other UEs in the training process.

[0086] In further step 4099, the one or more VAL UEs 410, 440 train the model based on the pre-trained model. The trained model is then used to derive UE/session QoS analytics.

[0087] In case of multiple entities involved, the other VAL UEs 440 may send the trained model to VAL UE #1 410 to derive analytics for the QoS prediction. The derivation of local analytics can be performed either in UE#1 410 or in parallel in multiple UEs 410, 440.

[0088] Figure 5 illustrates an example 500 of transfer learning for VAL session performance analytics in accordance with aspects of the present disclosure.

[0089] In this example 500, transfer learning is used when a ADAEC is using AI/ML methods to derive VAL session performance analytics (as specified in 3GPP TS 23.436 clause 8.2.3). Transfer learning can be used by other ADAE analytics tasks (like VAL server performance or Edge Load analytics) so as to get the pre-trained model and use it at the ADAEC/VAL client to locally train the model for another task (i.e., the VAL session performance analytics). Expressed differently, for TL purposes the destination analytics ID is “VAL session perf” whereas the partner analytics ID is for example: “VAL server #1 perf”, “VAL server #2 perf”, “EAS perf #1”, “EDN ID”.

[0090] In 3GPP TS 23.436 clause 8.2.2 and 8.8 example analytics services are identified. In clause 8.2.2 the VAL server performance analytics ID / event is defined to provide insight on the operation and performance of an application service (i.e., of VAL server or EAS), and in particular statistics or prediction on parameters related to e.g., VAL server number of connections for a given time and area, VAL server rate of connection requests, connection probability failure rates, RTT and deviations for a VAL server, packet loss rates etc. In clause 8.8 the edge load analytics provide insight on the operation and performance of an edge data network (EDN) and in particular statistics or prediction on parameters related to: the edge application server (EAS) / edge enablement server (EES)

load for one or more EAS/EES; edge platform load parameters, which include the aggregated load per EDN or per data network access identifier (DNAI) due to the edge support services and e.g., load level of edge computational resources.

[0091] The example 500 shows a UE#1 510 comprising an ADAEC (with MTME functionality) 512 and a AI enabler client 514. Also shown is an ADAES (with MTME functionality) 520, an AI enabler server 522, sources of pre-trained models and ML repository 530, and a UE#2 540. The various messaging flows will now be described.

[0092] In a first step 501, the ADAEC 512 in VAL UE#1 510 requests from AI Enabler Client 514 to support training a model for deriving local analytics for predicting performance for a given session (UE to UE or UE to Server) for a given area and time of interest (VAL/edge service area #x, time window y).

[0093] If the ADAEC 512 includes MTME or AI Enabler client 514 is not supported, then this step 501 may be omitted.

[0094] If the training happens at a VAL client then this step 501 is between ADAEC 512 and VAL client in VAL UE#1 510.

[0095] This step 501 may be also between ADAEC 512 and AI Enabler Client/ VAL clients belonging to different VAL UEs within a group.

[0096] In a further step 502, the ADAEC 512 or AI Enabler Client 514/VAL client identifies that the UE 510 is not capable of locally training the model from scratch due to time/data/processing/energy limitations and sends a request to ADAES 520 to find the available pre-trained models which can be applicable for the given analytics task (i.e., VAL session performance analytics for either Uu or PC5 session). This request may include some of the following parameters: analytics ID; requestor ID (i.e., app ID, VAL client ID, UE ID, group ID, ADAEC ID); list of EAS/EESs to be considered as sources; VAL session ID or profile or session type or vertical identifier / consumer identifier to help identifying the KPIs for which the analytics service applies; requirements related to the required features/datasets for the pre-trained models; area of applicability and time of validity for the request; preferred confidence level; cause for requesting pre-trained models (i.e., energy, processing limitations, time or lack of data); requirement for when to send the requested

information (for example only in good channel conditions, low load); whether analytics apply for Uu or PC5 session and/or transmission mode (unicast, broadcast, groupcast); one or more PLMN/NPN IDs and vendor IDs for which the request is allowable to be made (based on UE capabilities and SLAs); and edge provider information / EDN ID / DNAI or DNN.

[0097] In a further step 503, the ADAES 520 with the optional support of AI Enabler Server 522 collects information for base models for analytics related to VAL/Edge performance prediction from the model repository (e.g., A-ADRF) or from the sources directly (e.g. VAL server, other ADAES, AI enablers) 530. The information may include one or more of the following parameters: model ID and/or profile; analytics ID (or IDs) for which the model has been used or matched (candidate partner analytics IDs); context information including features, environment, data set requirements, type of ML methods/types used (e.g., reinforcement learning, federated learning); vendor ID and vendor interoperability information; permissions for exposure and being used as pre-trained model (excluded list of users / services / application types to use it); allowable CRUD operations; URL/address to fetch the model and optionally an ML repository ID/address to fetch the model; whether the model can be directly fetched by the UE 510 or via enabler 522; area and time of validity; source (ML provider) requirement for when the model is available for download (low load scenarios, good channel conditions). This step 503 may be expressed differently as discover pre-trained models via ADAES or AI enabler server.

[0098] In further step 504, if AI Enabler Server 522 is collecting the information this step 503 includes providing the pre-trained model information from AI Enabler Server 522 to ADAES 520.

[0099] In a further step 505, the ADAES 520 sends the collected information to the ADAEC 512 of the VAL UE #1 510 as response to step 502. This message may include the information of step 503 of a subset of it or any abstraction of it.

[0100] In certain embodiments, this information can be bundled in one message or can be sent as independent messages from either the AI Enabler Server 522 and/or the one or more sources 530 of the base models.

[0101] This step 505 may be expressed as providing information for candidate pre-trained models for partner analytics ID.

[0102] In further step 506, the ADAEC 512 interacts with the VAL client or AI Enabler client 514 (depending on which is going to train the model) to select one of the models for the analytics task (this can be done jointly or at the enabler client 514 or at the VAL client with the recommendation from enabler client 514). This step can be expressed as selecting the pre-trained model and matching of VAL session performance analytics to partner task/analytics ID.

[0103] In a further step 507a, m the ADAEC 512 or the VAL client or AI Enabler client 514 requests to download the model from the AI enabler server 522 or from the repository or from the source 530 of ML model directly.

[0104] In the step 507b, the AI Enabler client 514 then receives the requested model for the partner analytics ID/ task ID.

[0105] In the step 508, the VAL UE#1 510, if the training involves more UEs 540 (in group-based communications), further sends a request to further UEs 540 to perform part of the ML model training and also sends the ML model info (ID/address) to allow the other UEs 540 to download. Or the VAL UE#1 510 may also send the model itself via side-link. This step can be expressed as exchanging pre-trained models with other UEs in the training process.

[0106] In the step 509, the one or more VAL UEs 510, 540 train the model based on the pre-trained model. The one or more VAL UEs 510, 540 derive, using the re-trained model, VAL session performance analytics.

[0107] Figure 6 illustrates an example of a UE 600 in accordance with aspects of the present disclosure. The UE 600 may include a processor 602, a memory 604, a controller 606, and a transceiver 608. The processor 602, the memory 604, the controller 606, or the transceiver 608, 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.

[0108] The processor 602, the memory 604, the controller 606, or the transceiver 608, 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.

[0109] The processor 602 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 602 may be configured to operate the memory 604. In some other implementations, the memory 604 may be integrated into the processor 602. The processor 602 may be configured to execute computer-readable instructions stored in the memory 604 to cause the UE 600 to perform various functions of the present disclosure.

[0110] The memory 604 may include volatile or non-volatile memory. The memory 604 may store computer-readable, computer-executable code including instructions when executed by the processor 602 cause the UE 600 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 604 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.

[0111] In some implementations, the processor 602 and the memory 604 coupled with the processor 602 may be configured to cause the UE 600 to perform one or more of the functions described herein (e.g., executing, by the processor 602, instructions stored in the memory 604). For example, the processor 602 may support wireless communication at the UE 600 in accordance with examples as disclosed herein. The UE 600 may be configured to support a means for performing aspects of the methods disclosed herein. The UE 600 may be a UE entity such as UE 310, 340 or App/Enabler 312, 342 or Figure 3. The UE 600 may be UE 410, 440 or AI enabler 412 of Figure 4. The UE 600 may be UE 510, 540, or AI enabler 514 or ADAEC 512 of Figure 5.

[0112] The controller 606 may manage input and output signals for the UE 600. The controller 606 may also manage peripherals not integrated into the UE 600. In some implementations, the controller 606 may utilize an operating system such as iOS®, ANDROID®, WINDOWS®, or other operating systems. In some implementations, the controller 606 may be implemented as part of the processor 602.

[0113] In some implementations, the UE 600 may include at least one transceiver 608. In some other implementations, the UE 600 may have more than one transceiver 608. The transceiver 608 may represent a wireless transceiver. The transceiver 608 may include one or more receiver chains 610, one or more transmitter chains 612, or a combination thereof.

[0114] A receiver chain 610 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 610 may include one or more antennas for receive the signal over the air or wireless medium. The receiver chain 610 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 610 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 610 may include at least one decoder for decoding the processing the demodulated signal to receive the transmitted data.

[0115] A transmitter chain 612 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 612 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 612 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 612 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0116] Figure 7 illustrates an example of a processor 700 in accordance with aspects of the present disclosure. The processor 700 may be an example of a processor configured to

perform various operations in accordance with examples as described herein. The processor 700 may include a controller 702 configured to perform various operations in accordance with examples as described herein. The processor 700 may optionally include at least one memory 704, which may be, for example, an L1/L2/L3 cache. Additionally, or alternatively, the processor 700 may optionally include one or more arithmetic-logic units (ALUs) 706. 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).

[0117] The processor 700 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 700) 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).

[0118] The controller 702 may be configured to manage and coordinate various operations (e.g., signalling, receiving, obtaining, retrieving, transmitting, outputting, forwarding, storing, determining, identifying, accessing, writing, reading) of the processor 700 to cause the processor 700 to support various operations in accordance with examples as described herein. For example, the controller 702 may operate as a control unit of the processor 700, generating control signals that manage the operation of various components of the processor 700. These control signals include enabling or disabling functional units, selecting data paths, initiating memory access, and coordinating timing of operations.

[0119] The controller 702 may be configured to fetch (e.g., obtain, retrieve, receive) instructions from the memory 704 and determine subsequent instruction(s) to be executed to cause the processor 700 to support various operations in accordance with examples as described herein. The controller 702 may be configured to track memory address of

instructions associated with the memory 704. The controller 702 may be configured to decode instructions to determine the operation to be performed and the operands involved. For example, the controller 702 may be configured to interpret the instruction and determine control signals to be output to other components of the processor 700 to cause the processor 700 to support various operations in accordance with examples as described herein. Additionally, or alternatively, the controller 702 may be configured to manage flow of data within the processor 700. The controller 702 may be configured to control transfer of data between registers, arithmetic logic units (ALUs), and other functional units of the processor 700.

[0120] The memory 704 may include one or more caches (e.g., memory local to or included in the processor 700 or other memory, such RAM, ROM, DRAM, SDRAM, SRAM, MRAM, flash memory, etc. In some implementations, the memory 704 may reside within or on a processor chipset (e.g., local to the processor 700). In some other implementations, the memory 704 may reside external to the processor chipset (e.g., remote to the processor 700).

[0121] The memory 704 may store computer-readable, computer-executable code including instructions that, when executed by the processor 700, cause the processor 700 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 702 and/or the processor 700 may be configured to execute computer-readable instructions stored in the memory 704 to cause the processor 700 to perform various functions. For example, the processor 700 and/or the controller 702 may be coupled with or to the memory 704, the processor 700, the controller 702, and the memory 704 may be configured to perform various functions described herein. In some examples, the processor 700 may include multiple processors and the memory 704 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.

[0122] The one or more ALUs 706 may be configured to support various operations in accordance with examples as described herein. In some implementations, the one or more

ALUs 706 may reside within or on a processor chipset (e.g., the processor 700). In some other implementations, the one or more ALUs 706 may reside external to the processor chipset (e.g., the processor 700). One or more ALUs 706 may perform one or more computations such as addition, subtraction, multiplication, and division on data. For example, one or more ALUs 706 may receive input operands and an operation code, which determines an operation to be executed. One or more ALUs 706 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 706 may support logical operations such as AND, OR, exclusive-OR (XOR), not-OR (NOR), and not-AND (NAND), enabling the one or more ALUs 706 to handle conditional operations, comparisons, and bitwise operations.

[0123] The processor 700 may support wireless communication in accordance with examples as disclosed herein. The processor 700 may be configured to or operable to support a means for performing aspects methods disclosed herein. The processor 700 may be a processor 602 of Figure 6.

[0124] Figure 8 illustrates an example of a NE 800 in accordance with aspects of the present disclosure. The NE 800 may comprise application enablement or vertical enablement or edge enablement entities (SEAL, EDGEAPP, CAPIF) which are specified in 3GPP SA6. The NE 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.

[0125] 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.

[0126] 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 NE 800 to perform various functions of the present disclosure.

[0127] 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 NE 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.

[0128] In some implementations, the processor 802 and the memory 804 coupled with the processor 802 may be configured to cause the NE 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 NE 800 in accordance with examples as disclosed herein. The NE 800 may be configured to support a means for performing aspects of the methods disclosed herein. The NE 800 may be an AI enabler server 320 or ML model repository 330 of Figure 3. The NE 800 may be an AI enabler server 420, ML model repository 430 or source of pre-trained models 432 of Figure 4. The NE 800 may be an ADAES 520, AI enabler server 522 or source of pre-trained models/ML repository 530 of Figure 5.

[0129] The controller 806 may manage input and output signals for the NE 800. The controller 806 may also manage peripherals not integrated into the NE 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.

[0130] In some implementations, the NE 800 may include at least one transceiver 808. In some other implementations, the NE 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.

[0131] 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 processing the demodulated signal to receive the transmitted data.

[0132] 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.

[0133] Figure 9 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.

[0134] At 902, the method may include determining a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event. The operations of 902 may be performed in accordance with examples as described herein. In

some implementations, aspects of the operations of 902 may be performed by a UE as described with reference to Figure 6.

[0135] At 904, the method may include obtaining, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events. The operations of 904 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 904 may be performed by a UE as described with reference to Figure 6.

[0136] At 906, the method may include selecting a pre-trained ML model from the one or more available pre-trained ML models. The operations of 906 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 906 may be performed a UE as described with reference to Figure 6.

[0137] At 908, the method may include training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model. The operations of 908 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 908 may be performed a UE as described with reference to Figure 6.

[0138] 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 processor as described herein.

[0139] At 1002, the method may include inputting a requirement for one or more pre-trained machine learning 'ML' models for transfer learning, the transfer learning for a UE analytics event. 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 processor as described with reference to Figure 7.

[0140] At 1004, the method may include inputting, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for

one or more network analytics events. 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 processor as described with reference to Figure 7.

[0141] At 1006, the method may include outputting a selection of a pre-trained ML model from the one or more available pre-trained ML models. 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 processor as described with reference to Figure 7.

[0142] At 1008, the method may include training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model. 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 processor as described with reference to Figure 7.

[0143] 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.

[0144] At 1102, the method may include receiving a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event. 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 8.

[0145] At 1104, the method may include obtaining a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events. 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 8.

[0146] At 1106, the method may include sending a first response to the UE entity, wherein the first response comprises the first information. The operations of 1106 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1106 may be performed a NE as described with reference to Figure 8.

[0147] It should be noted that the methods described herein describe possible implementations, and that the operations and the steps may be rearranged or otherwise modified and that other implementations are possible.

[0148] There is provided a user equipment 'UE' entity for wireless communication, comprising: at least one memory; and at least one processor coupled with the at least one memory and configured to cause the UE entity to: determine a requirement for one or more pre-trained machine learning 'ML' models for transfer learning, the transfer learning for a UE analytics event; obtain, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; select a pre-trained ML model from the one or more available pre-trained ML models; and train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0149] Transfer learning is a technique in ML where a model trained on a first ML task is used as the starting point for a model on a second ML task. This can be useful when the second ML task has similarities with the first ML task and/or when there is limited data available for training an entirely new ML model for the second ML task. By using the learned features from the first ML task as a starting point, a ML model can learn more quickly and effectively on the second ML task. This can also help to prevent overfitting, as the ML model will have already learned general features that are likely to be useful in the second ML task.

[0150] A pre-trained ML model is a ML model that has previously been trained for a certain ML task. The pre-trained ML model may have been trained on extensive training datasets. The pre-trained ML model may have identified general features and patterns relevant to numerous related ML tasks. A base ML model or base model is the pre-trained model chosen to form the starting point for re-training for a subsequent ML task. It may be made up of layers that have utilized the training data from a first ML task to learn hierarchical feature representations. In transfer learning, a set of layers of the base model can be found that capture generic information related to a second ML task. Using available training data for the second ML task, the chosen layers of the base model can be retrained – a process known as ‘fine tuning’. This preserves the knowledge from the pre-training (for the first ML task) while enabling the ML model to be modified to better suit the demands of the second ML task.

[0151] In the 3GPP Technical Specification 22.261, titled “Service Requirements for the 5G System”, it is stated that the 5G system needs to support AI/ML model/data distribution and sharing.

[0152] When applying transfer learning to features in 5G and beyond systems, there can be different use cases which can be assumed for the operations to be impacted. One of them is when transfer learning is employed to assist AI/ML operations between AF/VAL servers and VAL UEs, or among VAL UEs.

[0153] A particular use case is where the network/server trains the model for predicting the network QoS for a certain cell or network area, and this is then used at the UE side to predict QoS for a particular session when traversing the cell/area of interest.

[0154] For transfer learning enablement in 3GPP systems focusing more on the application layer, a problem arises with how to deal with the scenario where the VAL UE utilizes the pre-trained model to perform local training with minimum processing. Furthermore, current 3GPP systems do not address how a UE must be configured to enable transfer learning, let alone how such a UE should interface with the wider 3GPP system.

[0155] The disclosure herein provides a mechanism to allow local UE performance prediction (i.e., UE QoS prediction) with the support of AI/ML by using transfer learning

from ML tasks related to network performance prediction (i.e., QoS) for a target area of interest (i.e., cell area).

[0156] Such a solution provides enhancement of existing 3GPP systems. Specifically, such a solution enables the benefits of transfer learning for UEs in 3GPP systems.

[0157] The UE analytics event may be referred to herein as a UE driven analytics event. The “select for the transfer learning” includes determining a selection of the one or more available pre-trained ML models for the UE analytics event.

[0158] The UE entity may comprise one or more UE/device application entities. The one or more UE/device application entities may be configured to operate independently or in combination to perform the methods described herein.

[0159] The first ML model being based on the selected pre-trained ML model may comprise the first ML model being equivalent to the selected pre-trained ML model or to a portion thereof or to an abstraction thereof.

[0160] The at least one processor may be configured to cause the UE entity to obtain the first information by causing the UE entity to: send a first request to the at least one network entity, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and receive a first response from the at least one network entity, the first response comprising the first information.

[0161] The first response may be a single response i.e., one message. However, the first response may alternatively be provided as a plurality of messages from the at least one network entity. For instance, a plurality of messages may be provided by an AI enabler server and/or the sources of the pre-trained ML models.

[0162] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model; an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model;

one or more required conditions for providing the first information; a transmission mode of the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the first request is allowable; one or more key performance indicators 'KPI' applicable to the UE analytics event; an edge provider information; and one or more edge application servers 'EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

[0163] The requestor may be the UE entity. The identifier for the requestor may comprise an application ID, a VAL client ID, a UE ID, a group ID, or an ADAE client ID. The identifier for a session or service may be an identifier for a VAL session or service.

[0164] The cause of the requirement for a pre-trained ML model may be a constraint relating to energy consumption, processing limitation, time or lack of data.

[0165] The one or more required conditions may be good channel conditions or a low data load.

[0166] The transmission mode may be unicast, broadcast, or groupcast.

[0167] A vendor may be deemed allowable or not-allowable based on UE capabilities and/or service level agreements 'SLA'.

[0168] The edge provider information may comprise an identifier for an edge data network 'EDN', a data network access identifier 'DNAI', or a data network name 'DNN'.

[0169] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics associated with the respective pre-trained ML model; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete 'CRUD' operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or

time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0170] The first information may be acquired by one or more vendors or via one or more vendors (edge, cloud, PLMNs, for instance).

[0171] The one or more permissions may include an excluded list of users/services/application types.

[0172] The address from which the respective pre-trained ML model can be fetched may comprise an address for a ML repository. The address may be a URL.

[0173] The source requirement may comprise a time or date.

[0174] The context information may comprise one or more features of the respective pre-trained ML model, an environment in which the respective pre-trained ML model was trained/used or intended to be used, one or more dataset requirements, and/or a type of ML method used for training the respective pre-trained ML model (i.e., reinforcement learning, federated learning).

[0175] The at least one processor may be configured to cause the UE entity to determine the requirement by causing the UE entity to: obtain a notification from a logical ML model training entity, wherein the logical ML model training entity is for training an ML model for the UE analytics event.

[0176] The notification may comprise at least one of: an energy consumption restriction; a processing capability restriction; and an ML model training time restriction.

[0177] The UE entity may comprise at least one of: a vertical application layer 'VAL' client; an artificial intelligence 'AI' enabler client; an application data analytics enabler 'ADAE' client; and an edge enabler client.

[0178] The at least one network entity may comprise at least one of: an ML model repository entity, wherein the ML model repository entity is optionally an application layer analytics data repository function 'AADRF', an operations administration and maintenance 'OAM' entity or an edge repository; a logical ML model training entity, wherein the logical ML model training entity is optionally a model training and management entity 'MTME';

an ML model provider entity, wherein the ML model provider entity is optionally a VAL server, an ADAE server, an AI enabler server, a networks data analytics function 'NWDAF', or another application function 'AF'.

[0179] The UE analytics event may comprise at least one of: Quality of Service 'QoS' analytics for a UE; QoS analytics for a group of UEs; QoS analytics for a user session; QoS analytics for an application session; performance analytics for a sidelink session; and performance analytics for an uplink/downlink session. For completeness, sidelink tends to enable the direct communication between proximal UEs using PC5/ProSE interfaces. The sidelink interface refers to the communication over PC5/ProSE as well as the communication over the application enablement layer for direct UE to UE communications. A sidelink session can be a UE-to-UE session in the application enablement layer and/or lower layers (i.e., UE modem to UE modem or L1/L2 session).

[0180] The 'session' may be for a given area and/or time of interest (i.e., for a cell, group of cells, of time window). The area and time of interest may be a VAL/edge service area and time window.

[0181] The network analytics event may comprise at least one of: QoS analytics for a network; QoS analytics for a VAL server; analytics for performance/load of an edge server; analytics for an edge platform.

[0182] The at least one processor may be configured to cause the UE entity to select the pre-trained ML model based on one or more criteria, wherein the one or more criteria comprise for a respective available pre-trained ML model, at least one of: a rating of the respective available pre-trained ML model, the rating being provided by the first network entity and indicating a suitability of the respective available pre-trained ML model for the transfer learning; a historical usage of the respective available pre-trained ML model; a simulated usage of the respective available pre-trained ML model; an expected time, energy and/or processing requirement associated with training the respective available pre-trained ML model; an optimization of a network utility function; a location of the respective available pre-trained ML model; a similarity between the UE analytics event and the networks analytics event for which the respective available pre-trained ML model was trained or used; a similarity between context information of the UE analytics event and the

respective available pre-trained ML model. For completeness, a rating can be in form of a weight or percentage or relative/actual value or qualitative parameter (i.e., “good”, “average”) which is used to evaluate the applicability of the model for being used for the transfer learning. Such a rate can be derived based on feedback received from previous training from the consumer or from the provider of the service, or from a further verification entity which verifies the rating of the model.

[0183] The historical usage may be usage for other analytics tasks. The network utility function may be based on KPIs for latency, energy rate, or pricing values. The location may be an edge location, cloud location or domain with different preferences/trustworthiness.

[0184] The at least one processor may be further configured to cause the UE entity to obtain the selected pre-trained ML model, preferably by causing the UE entity to: send, to the at least one network entity, a second request to download the selected pre-trained ML model; and receive the selected pre-trained ML model.

[0185] The UE entity may obtain the selected pre-trained ML model. This may comprise obtaining from a repository of ML models or obtaining from a source of an ML model or a combination thereof. For example, the first information may be obtained from the first network entity (which may be a repository entity). The UE entity may then download the ML model from a separate server using an address provided by the first network entity.

[0186] The at least one processor may be configured to cause the UE entity to: distribute the selected pre-trained ML model to at least one second UE entity.

[0187] Alternatively, the UE entity may provide send an address or URL to the at least one second UE entity such that the at least one second UE entity can access and download the selected pre-trained ML model directly.

[0188] The at least one processor may be configured to cause the UE entity to train the first ML model by causing the UE entity to: receive the first ML model from the at least one second UE entity.

[0189] The at least one processor may be configured to cause the UE entity to: derive, using the first ML model, UE analytics for the UE analytics event.

[0190] There is provided a processor for wireless communication, comprising: at least one controller coupled with at least one memory and configured to cause the processor to: input a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; input, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; output, a selection of a pre-trained ML model from the one or more available pre-trained ML models; and train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0191] The at least one controller may be configured to cause the processor to input the first information by causing the processor to: output a first request, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and input a first response, the first response comprising the first information.

[0192] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model; an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model; one or more required conditions for providing the first information; a transmission mode of the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the first request is allowable; one or more key performance indicators 'KPI' applicable to the UE analytics event; an edge provider information; and one or more edge application servers EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

[0193] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an

identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics associated with the respective pre-trained ML model; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete 'CRUD' operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0194] The at least one controller may be configured to cause the processor to input the requirement by causing the processor to: input a notification from a logical ML model training entity, wherein the logical ML model training entity is for training an ML model for the UE analytics event.

[0195] The notification may comprise at least one of: an energy consumption restriction; a processing capability restriction; and an ML model training time restriction.

[0196] The UE analytics event may comprise at least one of: Quality of Service 'QoS' analytics for a UE; QoS analytics for a group of UEs; QoS analytics for a user session; QoS analytics for an application session; performance analytics for a sidelink session; and performance analytics for an uplink/downlink session.

[0197] The network analytics event may comprise at least one of: QoS analytics for a network; QoS analytics for a VAL server; analytics for performance/load of an edge server; analytics for an edge platform.

[0198] The at least one controller may be configured to cause the processor to output the selection of the pre-trained ML model based on one or more criteria, wherein the one or more criteria comprise for a respective available pre-trained ML model, at least one of: a rating of the respective available pre-trained ML model, the rating being provided by the first network entity and indicating a suitability of the respective available pre-trained ML model for the transfer learning; a historical usage of the respective available pre-trained ML model; a simulated usage of the respective available pre-trained ML model; an expected

time, energy and/or processing requirement associated with training the respective available pre-trained ML model; an optimization of a network utility function; a location of the respective available pre-trained ML model; a similarity between the UE analytics event and the networks analytics event for which the respective available pre-trained ML model was trained or used; a similarity between context information of the UE analytics event and the respective available pre-trained ML model.

[0199] The at least one controller may be configured to cause the processor to input the selected pre-trained ML model, preferably by causing the processor to: output a second request to download the selected pre-trained ML model; and input the selected pre-trained ML model.

[0200] The at least one controller may be configured to cause the processor to: output, using the first ML model, UE analytics for the UE analytics event.

[0201] There is provided a network entity for wireless communication, 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 a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtain a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; send a first response to the UE entity, wherein the first response comprises the first information.

[0202] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model; an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model; one or more required conditions for providing the first information; a transmission mode of the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the

first request is allowable; one or more key performance indicators ‘KPI’ applicable to the UE analytics event; an edge provider information; and one or more edge application servers EAS’ or edge enabler servers ‘EES’ to be considered as sources for pre-trained ML models.

[0203] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics for which the respective pre-trained ML model has been used; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete ‘CRUD’ operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0204] There is provided a method performed by a UE entity, comprising: determining a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtaining, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; selecting a pre-trained ML model from the one or more available pre-trained ML models; and training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0205] The obtaining the first information may comprise: sending a first request to the at least one network entity, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and receiving a first response from the at least one network entity, the first response comprising the first information.

[0206] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a

session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model; an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model; one or more required conditions for providing the first information; a transmission mode of the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the first request is allowable; one or more key performance indicators 'KPI' applicable to the UE analytics event; an edge provider information; and one or more edge application servers 'EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

[0207] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics for which the respective pre-trained ML model has been used; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete 'CRUD' operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0208] The determining the requirement may comprise obtaining a notification from a logical ML model training entity, wherein the logical ML model training entity is for training an ML model for the UE analytics event.

[0209] The notification may comprise at least one of: an energy consumption restriction; a processing capability restriction; and an ML model training time restriction.

[0210] The UE entity may comprise at least one of: a vertical application layer 'VAL' client; an artificial intelligence 'AI' enabler client; an application data analytics enabler 'ADAE' client; and an edge enabler client.

[0211] The at least one network entity may comprise at least one of: an ML model repository entity, wherein the ML model repository entity is optionally an application layer analytics data repository function 'AADRF', an operations administration and maintenance 'OAM' entity or an edge repository; a logical ML model training entity, wherein the logical ML model training entity is optionally a model training and management entity 'MTME'; an ML model provider entity, wherein the ML model provider entity is optionally a VAL server, an ADAE server, an AI enabler server, a networks data analytics function 'NWDAF' another application function 'AF'.

[0212] The UE analytics event may comprise at least one of: Quality of Service 'QoS' analytics for a UE; QoS analytics for a group of UEs; QoS analytics for a user session; QoS analytics for an application session; performance analytics for a sidelink session; and performance analytics for an uplink/downlink session.

[0213] The network analytics event may comprise at least one of: QoS analytics for a network; QoS analytics for a VAL server; analytics for performance/load of an edge server; analytics for an edge platform.

[0214] The selecting may comprise selecting the pre-trained ML model based on one or more criteria, wherein the one or more criteria comprise for a respective available pre-trained ML model, at least one of: a rating of the respective available pre-trained ML model, the rating being provided by the first network entity and indicating a suitability of the respective available pre-trained ML model for the transfer learning; a historical usage of the respective available pre-trained ML model; a simulated usage of the respective available pre-trained ML model; an expected time, energy and/or processing requirement associated with training the respective available pre-trained ML model; an optimization of a network utility function; a location of the respective available pre-trained ML model; a similarity between the UE analytics event and the networks analytics event for which the respective available pre-trained ML model was trained or used; a similarity between context information of the UE analytics event and the respective available pre-trained ML model.

[0215] The method may comprise obtaining the selected pre-trained ML model, preferably by: sending, to the at least one network entity, a second request to download the selected pre-trained ML model; and receiving the selected pre-trained ML model.

[0216] The method may comprise distributing the selected pre-trained ML model to at least one second UE entity.

[0217] The method may comprise training the first ML model by causing the UE entity to: receive the first ML model from the at least one second UE entity.

[0218] The method may comprise deriving, using the first ML model, UE analytics for the UE analytics event.

[0219] There is further provided a method performed by a processor, comprising: inputting a requirement for one or more pre-trained machine learning 'ML' models for transfer learning, the transfer learning for a UE analytics event; inputting, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; outputting a selection of a pre-trained ML model from the one or more available pre-trained ML models; and training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.

[0220] The method may comprise outputting a first request, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and inputting a first response, the first response comprising the first information.

[0221] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model; an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model; one or more required conditions for providing the first information; a transmission mode of

the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the first request is allowable; one or more key performance indicators 'KPI' applicable to the UE analytics event; an edge provider information; and one or more edge application servers 'EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

[0222] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics for which the respective pre-trained ML model has been used; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete 'CRUD' operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0223] The inputting the requirement may comprise: inputting a notification from a logical ML model training entity, wherein the logical ML model training entity is for training an ML model for the UE analytics event.

[0224] The notification may comprise at least one of: an energy consumption restriction; a processing capability restriction; and an ML model training time restriction.

[0225] The UE analytics event may comprise at least one of: Quality of Service 'QoS' analytics for a UE; QoS analytics for a group of UEs; QoS analytics for a user session; QoS analytics for an application session; performance analytics for a sidelink session; and performance analytics for an uplink/downlink session.

[0226] The network analytics event may comprise at least one of: QoS analytics for a network; QoS analytics for a VAL server; analytics for performance/load of an edge server; analytics for an edge platform.

[0227] The outputting the selection may comprise outputting the selection of the pre-trained ML model based on one or more criteria, wherein the one or more criteria comprise for a respective available pre-trained ML model, at least one of: a rating of the respective available pre-trained ML model, the rating being provided by the first network entity and indicating a suitability of the respective available pre-trained ML model for the transfer learning; a historical usage of the respective available pre-trained ML model; a simulated usage of the respective available pre-trained ML model; an expected time, energy and/or processing requirement associated with training the respective available pre-trained ML model; an optimization of a network utility function; a location of the respective available pre-trained ML model; a similarity between the UE analytics event and the networks analytics event for which the respective available pre-trained ML model was trained or used; a similarity between context information of the UE analytics event and the respective available pre-trained ML model.

[0228] The method may further comprise inputting the selected pre-trained ML model, preferably by: outputting a second request to download the selected pre-trained ML model; and inputting the selected pre-trained ML model.

[0229] The method may comprise outputting, using the first ML model, UE analytics for the UE analytics event.

[0230] There is provided a method performed by a network entity, comprising: receiving a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event; obtaining a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events; and sending a first response to the UE entity, wherein the first response comprises the first information.

[0231] The one or more first parameters may comprise at least one of: an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer; one or more required features of a pre-trained ML model; one or more required attributes of a dataset of a pre-trained ML model;

an area of applicability and/or time of validity of the first request; a preferred confidence level for a pre-trained ML model; a cause of the requirement for a pre-trained ML model; one or more required conditions for providing the first information; a transmission mode of the UE analytics event; an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable; an identifier for one or more vendors for which the first request is allowable; one or more key performance indicators 'KPI' applicable to the UE analytics event; an edge provider information; and one or more edge application servers 'EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

[0232] The first information may comprise one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of: an identifier for the respective pre-trained ML model or pre-trained ML model profile; an identifier for analytics for which the respective pre-trained ML model has been used; an identifier for a vendor or vendor interoperability; a context information; one or more permissions for exposure of the respective pre-trained ML model for transfer learning; one or more allowable create read update delete 'CRUD' operations; an address from which the respective pre-trained ML model can be fetched; an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity; an area and/or time of validity; and a source requirement for when the respective pre-trained ML model is available.

[0233] Transfer Learning is used to support an entity to ease the training process (i.e., if conventional training is not possible or if the data and time to train is not sufficient). When applying Transfer Learning (TL) to features in 5G and beyond systems, there can be different use cases which can be assumed for the operations to be impacted. One of them is when TL is employed to assist AI/ML operations between AF/VAL servers and VAL UEs or among VAL UEs. More specifically there is the use case where the network/server trains a ML model for predicting a network QoS for a certain cell or network area and then this is used at the UE side to predict the QoS for a particular session when traversing the cell / area of interest. The disclosure herein tends to address the problem of how to deal with the scenario where the VAL UE utilizes the pre-trained ML model to perform local training

with minimum processing. The disclosure herein further tends to address the impact at the UE side and the interfaces.

[0234] This disclosure herein presents solutions that provide mechanisms to allow local UE performance prediction (e.g., UE QoS prediction) with the support of AI/ML, by utilizing transfer learning from ML tasks related to network QoS / performance for a target area of interest (e.g., cell area).

[0235] Current 3GPP solutions do not consider transfer learning techniques (i.e., using a pre-trained ML model from one task to another) for AI enabled analytics. More specifically, current 3GPP solutions do not address the local training at the UE side..

[0236] More specifically, the disclosure herein provides a first example wherein the pre-trained ML model is used for QoS prediction at the UE side for a Uu/ PC5 session. Such QoS prediction happens at the application enabler client using the ML model trained for network QoS sustainability analytics.

[0237] More specifically, the disclosure herein provides a second example wherein transfer learning is used when ADAEC is using AI/ML methods to derive VAL session performance analytics (as specified in 3GPP TS 23.436 clause 8.2.3). Transfer learning can be used by other ADAE analytics tasks (like VAL server performance or Edge Load analytics) so as to get the pre-trained model and use it at the ADAEC/VAL client to locally train the model for another task (i.e., the VAL session performance analytics).

[0238] There is provided, a method, at a device application, for supporting transfer learning for a UE driven analytics event, the method comprising: detecting a requirement for utilizing a pre-trained ML model for the UE driven analytics event, wherein the pre-trained ML model is trained for one or more network analytics events; obtaining information from at least one network entity, wherein the information is related to at least one available ML model related to the one or more network analytics events based on the requirement; determining the selection of a pre-trained ML model from the at least one available ML models for the UE driven analytics event; obtaining the selected pre-trained ML model from the at least network entity; and training the ML model for deriving the UE driven analytics using the selected pre-trained ML model.

[0239] The device application may comprise a vertical application layer client, an AI enabler client, an ADAEC, an edge enabler client or a combination thereof.

[0240] The detecting the requirement may comprise obtaining a notification from a logical ML model training entity, wherein the logical ML model training entity is expected to train the ML model for the UE driven analytics event.

[0241] The obtained notification may comprise an energy consumption restriction, a processing capability restriction, an ML model training time limitation or a combination thereof.

[0242] The network entity may be a ML model repository function, a further logical ML model training entity, an ML model provider function or a combination thereof.

[0243] The obtaining information from the at least one network entity may comprise: requesting or subscribing to receive at least one parameter related to the at least one available ML model; and receiving the requested at least one parameter.

[0244] The information and/or parameter from the at least one network entity may comprise information and/or parameters as described in respect of the examples herein.

[0245] The UE driven analytics event may comprise per UE QoS analytics, per group of UEs QoS analytics, per user session QoS analytics, per application session QoS analytics, performance analytics for a sidelink session, performance analytics for an UL/DL session or a combination thereof.

[0246] The network analytics event may comprise per network QoS analytics, per VAL server analytics, per edge server performance and/or load analytics, per edge platform analytics or a combination thereof.

[0247] The obtaining the selected pre-trained ML model may comprise receiving from an ML model repository, an application server, a network function, a management function, an enablement server or a combination thereof.

[0248] The determination of the selection may be based on one or more criteria as described in respect of the examples herein.

[0249] The obtaining the model information may comprise: requesting to download the ML model from the network entity and/or model repository; and receiving the ML model.

[0250] The method may further comprise sending the selected pre-trained ML model information to at least one further device application.

[0251] The method may further comprise receiving the trained ML model based on the pre-trained ML model from further device applications within a group.

[0252] The method may further comprise deriving UE driven analytics based on the selected pre-trained model.

[0253] 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.

[0254] 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.

[0255] As used herein, an “ML model” may comprise a mathematical algorithm that can be "trained" by data and human expert input as examples to replicate a decision an expert would make when provided that same information (see 3GPP TS 28.105). As used herein, “ML model training” may include capabilities of an ML training function or service to take data, run it through an ML model, derive the associated loss and adjust the parameterization of that ML model based on the computed loss (see 3GPP TS 28.105). As used herein, “ML model inference may include capabilities of an ML model inference function that employs an ML model and/or AI decision entity to conduct inference (see 3GPP TS 28.105). As used herein “AI/ML enablement” may comprise an application enablement framework consisting of one or more AI/ML enabler capabilities based on the SA6 provider implementation. Such function can be deployed as (or within) an enablement layer server (e.g. SEAL or ADAES) or client. As used herein the term “repository” may be

used interchangeably with the term “model repository”. The term “model” may be used interchangeably with the term “ML model”.

[0256] The following abbreviations are used herein: TL, Transfer Learning; MDA, Management Domain Analytics; AIML, Artificial Intelligence / Machine Learning; ADAES, Application Data Analytics Enabler Server; ADAEC, Application Data Analytics Enabler Client; SEAL, Service Enabler Architecture Layer; NW, Network; MTME, Model Training and Management Entity; CCF, CAPIF (Common API Framework) Core Function; CRUD, Create-Read-Update-Delete; A-ADRF, Application layer – Analytics Data Repository Function; PMP, Pre-trained model producer; PMC, Pre-trained model consumer; TLSSP, TL Support Service Producer; TLSSC, TL Support Service Consumer; EDN, Edge Data Network; EES / EAS, Edge Enabler Server / Edge Application Server; and DNAI / DNN, Data Network Access Identifier / Data Network Name.

CLAIMS

1. A user equipment 'UE' entity for wireless communication, comprising:
 - at least one memory; and
 - at least one processor coupled with the at least one memory and configured to cause the UE entity to:
 - determine a requirement for one or more pre-trained machine learning 'ML' models for transfer learning, the transfer learning for a UE analytics event;
 - obtain, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events;
 - select a pre-trained ML model from the one or more available pre-trained ML models; and
 - train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.
2. The UE entity of claim 1, wherein the at least one processor is configured to cause the UE entity to obtain the first information by causing the UE entity to:
 - send a first request to the at least one network entity, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and
 - receive a first response from the at least one network entity, the first response comprising the first information.
3. The UE entity of claim 2, wherein the one or more first parameters comprise at least one of:
 - an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer;
 - one or more required features of a pre-trained ML model;
 - one or more required attributes of a dataset of a pre-trained ML model;

- an area of applicability and/or time of validity of the first request;
- a preferred confidence level for a pre-trained ML model;
- a cause of the requirement for a pre-trained ML model;
- one or more required conditions for providing the first information;
- a transmission mode of the UE analytics event;
- an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable;
- an identifier for one or more vendors for which the first request is allowable;
- one or more key performance indicators 'KPI' applicable to the UE analytics event;
- an edge provider information; and
- one or more edge application servers 'EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

4. The UE entity of any preceding claim, wherein the first information comprises one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of:

- an identifier for the respective pre-trained ML model or pre-trained ML model profile;
- an identifier for analytics associated with the respective pre-trained ML model;
- an identifier for a vendor or vendor interoperability;
- a context information;
- one or more permissions for exposure of the respective pre-trained ML model for transfer learning;
- one or more allowable create read update delete 'CRUD' operations;
- an address from which the respective pre-trained ML model can be fetched;
- an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity;
- an area and/or time of validity; and
- a source requirement for when the respective pre-trained ML model is available.

5. The UE entity of any preceding claim, wherein the at least one processor is configured to cause the UE entity to determine the requirement by causing the UE entity to:
obtain a notification from a logical ML model training entity, wherein the logical ML model training entity is for training an ML model for the UE analytics event.
6. The UE entity of claim 5, wherein the notification comprises at least one of:
an energy consumption restriction;
a processing capability restriction; and
an ML model training time restriction.
7. The UE entity of any preceding claim, wherein the UE entity comprises at least one of:
a vertical application layer 'VAL' client;
an artificial intelligence 'AI' enabler client;
an application data analytics enabler 'ADAE' client; and
an edge enabler client.
8. The UE entity of any preceding claim, wherein the at least one network entity comprises at least one of:
an ML model repository entity, wherein the ML model repository entity is optionally an application layer analytics data repository function 'AADRF', an operations administration and maintenance 'OAM' entity or an edge repository;
a logical ML model training entity, wherein the logical ML model training entity is optionally a model training and management entity 'MTME';
an ML model provider entity, wherein the ML model provider entity is optionally a VAL server, an ADAE server, an AI enabler server, a networks data analytics function 'NWDAF', or another application function 'AF'.
9. The UE entity of any preceding claim, wherein the UE analytics event comprises at least one of:
Quality of Service 'QoS' analytics for a UE;

- QoS analytics for a group of UEs;
- QoS analytics for a user session;
- QoS analytics for an application session;
- performance analytics for a sidelink session; and
- performance analytics for an uplink/downlink session.

10. The UE entity of any preceding claim, wherein the network analytics event comprises at least one of:

- QoS analytics for a network;
- QoS analytics for a VAL server;
- analytics for performance/load of an edge server;
- analytics for an edge platform.

11. The UE entity of any preceding claim, wherein the at least one processor is configured to cause the UE entity to select the pre-trained ML model based on one or more criteria, wherein the one or more criteria comprise for a respective available pre-trained ML model, at least one of:

- a rating of the respective available pre-trained ML model, the rating being provided by the first network entity and indicating a suitability of the respective available pre-trained ML model for the transfer learning;
- a historical usage of the respective available pre-trained ML model;
- a simulated usage of the respective available pre-trained ML model;
- an expected time, energy and/or processing requirement associated with training the respective available pre-trained ML model;
- an optimization of a network utility function;
- a location of the respective available pre-trained ML model;
- a similarity between the UE analytics event and the networks analytics event for which the respective available pre-trained ML model was trained or used;
- a similarity between context information of the UE analytics event and the respective available pre-trained ML model.

12. The UE entity of any preceding claim, wherein the at least one processor is further configured to cause the UE entity to obtain the selected pre-trained ML model, preferably by causing the UE entity to:
 - send, to the at least one network entity, a second request to download the selected pre-trained ML model; and
 - receive the selected pre-trained ML model.
13. The UE entity of any preceding claim, wherein the at least one processor is configured to cause the UE entity to:
 - distribute the selected pre-trained ML model to at least one second UE entity.
14. The UE entity of any preceding claim, wherein the at least one processor is configured to cause the UE entity to:
 - derive, using the first ML model, UE analytics for the UE analytics event.
15. A processor for wireless communication, comprising:
 - at least one controller coupled with at least one memory and configured to cause the processor to:
 - input a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event;
 - input, based on the requirement, a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events;
 - output, a selection of a pre-trained ML model from the one or more available pre-trained ML models; and
 - train a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.
16. The processor of claim 15, wherein the at least one controller is configured to cause the processor to input the first information by causing the processor to:

output a first request, wherein the first request comprises one or more first parameters for identifying the one or more available pre-trained ML models; and input a first response, the first response comprising the first information.

17. The processor of claim 16, wherein the one or more first parameters comprise at least one of:

- an identifier for one or more of analytics, a requestor, a session/service, a session/service profile, a session/service type, a vertical or a consumer;
- one or more required features of a pre-trained ML model;
- one or more required attributes of a dataset of a pre-trained ML model;
- an area of applicability and/or time of validity of the first request;
- a preferred confidence level for a pre-trained ML model;
- a cause of the requirement for a pre-trained ML model;
- one or more required conditions for providing the first information;
- a transmission mode of the UE analytics event;
- an identifier for one or more public land mobile networks 'PLMN' for which the first request is allowable;
- an identifier for one or more vendors for which the first request is allowable;
- one or more key performance indicators 'KPI' applicable to the UE analytics event;
- an edge provider information; and
- one or more edge application servers EAS' or edge enabler servers 'EES' to be considered as sources for pre-trained ML models.

18. The processor of any one of claims 15-17, wherein the first information comprises one or more second parameters, the one or more second parameters comprising for a respective pre-trained model, at least one of:

- an identifier for the respective pre-trained ML model or pre-trained ML model profile;
- an identifier for analytics associated with the respective pre-trained ML model;
- an identifier for a vendor or vendor interoperability;
- a context information;

one or more permissions for exposure of the respective pre-trained ML model for transfer learning;

one or more allowable create read update delete 'CRUD' operations;

an address from which the respective pre-trained ML model can be fetched;

an indication as to whether the respective pre-trained ML model can be directly or indirectly fetched by the UE entity;

an area and/or time of validity; and

a source requirement for when the respective pre-trained ML model is available.

19. A network entity for wireless communication, 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 a first request from a UE entity, wherein the first request comprises one or more first parameters for identifying one or more available pre-trained ML models for transfer learning, the transfer learning for a UE analytics event;

obtain a first information of one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events;

send a first response to the UE entity, wherein the first response comprises the first information.

20. A method performed by a UE entity, comprising:

determining a requirement for one or more pre-trained ML models for transfer learning, the transfer learning for a UE analytics event;

obtaining, based on the requirement, a first information from at least one network entity, wherein the first information indicates one or more available pre-trained ML models for the transfer learning, wherein the one or more available pre-trained ML models comprise ML models trained for one or more network analytics events;

selecting a pre-trained ML model from the one or more available pre-trained ML models; and

training a first ML model for deriving UE analytics for the UE analytics event, wherein the first ML model is based on the selected pre-trained ML model.