TRANSFER LEARNING IN APPLICATION LAYER MACHINE LEARNING TASKS IN A WIRELESS COMMUNICATION NETWORK

TECHNICAL FIELD

[0001] The subject matter disclosed herein relates generally to the field of implementing transfer learning in application layer machine learning tasks in a wireless communication network. This document defines a first network entity for wireless communication, a second network entity for wireless communication and a third network entity for wireless communication. This document further defines a method performed by a first network entity, a method performed by a second network entity and a method performed by a third 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 first 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 first network entity to: receive, from a second network entity, a first request to identify at least one pre-trained machine learning ‘ML’ model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; discover, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models; obtain, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and provide, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0006] There is further provided a second 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 second network entity to: send, to a first network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; and receive, from the first network entity, a second information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0007] There is further provided a third 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 third network entity to: receive, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at least one pre-trained ML model; determine, based on the one or more criteria, a first information of one or more available pre-trained ML models; and send, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models.

[0008] There is further provided a method performed by a first network entity, comprising: receiving, from a second network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; discovering, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models; obtaining, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and providing, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0009] There is further provided a method performed by a second network entity, comprising: sending, to a first network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; and receiving, from the first network entity, a second

information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0010] There is further provided a method performed by a third network entity, comprising: receiving, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at least one pre-trained ML model; determining, based on the one or more criteria, a first information of one or more available pre-trained ML models; and sending, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models.

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 messaging flow for ADAE triggered TL enablement in accordance with aspects of the present disclosure.

[0015] Figure 5 illustrates an example of messaging flow for VAL triggered TL enablement 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 NE in accordance with aspects of the present disclosure.

[0020] Figure 10 illustrates a flowchart of a method performed by a NE 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] For TL enablement in 3GPP systems focusing more on application layer, the following problems can be identified: how to identify whether a pre-trained model can be used for another ML model task (i.e., a new ML job) and how to ensure that such solution will have benefits; and how to configure the TL-enabled entities for the distribution of the pre-trained ML model.

[0041] This disclosure herein describes the identification of candidate pre-trained ML models and their selection based on criteria. Further described herein is the configuration of involved entities for the distribution of pre-trained ML models (i.e., the source of pre-trained ML models, the destination, the coordinator/registry). The disclosure herein also

includes the authorization and charging aspects. Such solutions enable the benefits of transfer learning as hereinbefore discussed, within 3GPP systems.

[0042] Aspects of the present disclosure are described in the context of a wireless communications system.

[0043] 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.

[0044] The NE 102 may comprise application enablement or vertical enablement or edge enablement entities (SEAL, EDGEAPP, CAPIF) which are specified in 3GPP SA6. 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[0057] Figure 2 illustrates an example of high-level transfer learning operation 200 in accordance with aspects of the present disclosure.

[0058] 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.

[0059] 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.

[0060] 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.

[0061] 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.

[0062] Figure 3 illustrates an example of a high-level architecture 300 in accordance with aspects of the present disclosure. Prior to describing the high-level architecture 300 in Figure 3, a number of roles will be introduced.

[0063] A pre-trained Model Producer (PMP) is an entity that can be the ML model provider or the ML model training entity that has produced the pre-trained ML model. Such an entity can be a vertical application layer (VAL) server or an AI/ML Enablement Server or an application data analytics enabler (ADAE) server with model training and management entity (MTME) capability. In certain embodiments, this entity is a service module or capability or functionality of the above entities.

[0064] A pre-trained Model Consumer (PMC) is an entity which consumes the pre-trained ML model. This entity will use the pre-trained model to perform its own training on top and can be a VAL server or a further AIML Enablement Server or a further ADAES with MTME capability, or even a 3GPP network analytics function.

[0065] A pre-trained model Registry (PMR) is optional and can be implemented as an A-ADRF, edge or cloud repository, or a CAPIF Core Function (CCF).

[0066] A TL Support Service Producer (TLSSP) may be the same entity as the PMC; however it can be different if for example the PMC is the VAL server (TLSSC) and it requires from AIML Enablement Server (TLSSP) to provide support for identifying a pre-trained ML model for a given task. So, the TLSSP tends to apply for the case where the pre-trained ML model training entity which consumes the ML model is not the decision entity, but delegates this to the enabler layer. This entity can be for instance AIML Enablement Server or ADAES with MTME functionality.

[0067] A TL Support Service Consumer (TLSSC) is an entity with the role which involves requesting the support service for deciding on the candidate and selected pre-trained ML model using certain criteria. Such an entity may be a VAL server or it can be also a 3GPP SA6 enabler functionality. The reason for requiring such a service is the fact that this entity doesn't necessarily have the permission and required capability and knowledge to judge what is the best pre-trained ML model to be used for a certain task, and consumes the service from the producer which is authorized to perform the selection.

[0068] A Charging Functionality (ChaF) is for TLSS and/or for pre-trained ML model utilization (allowing CRUD operations in a pre-trained ML model). Such functionality can be part of operations administration and maintenance (OAM) or part of the service enabler architecture layer (SEAL)/or CAPIF platform provider.

[0069] With regard to the example 300 of Figure 3, a consumer of pre-trained models 310 is shown. The consumer of pre-trained models 310 may be a PMC or TLSSC. Also shown in the example 300 is a transfer learning enabler 320. The transfer learning enabler 320 may be a TLSSP. The transfer learning enabler 320 interfaces with a model repository 330 (which may be a PMR), a source of base models 340 (which may be a PMP), with

enabler clients 382, 392 or UEs 380, 390 and ML enhanced application tasks 384 or 394. Also shown in the example 300 is a ChaF 350 that interfaces with model repository 330, a 5GC 360 and a RAN 370. The UEs 380, 390 interface with RAN 370 over Uu. Various high-level procedural steps for transfer learning in the example 300 will now be described.

[0070] In a first step 301, the TLSSC 310 detects the need for using transfer learning for a certain ML task / job ID or for a given Analytics ID.

[0071] In a further step 302, the TLSSC 310 subscribes or requests TLSSP 320 to determine one or more available pre-trained ML models for the given ML task/job or analytics ID. In this step, TLSSC 310 provides all the necessary context information including set of features, dataset requirement, model information, required confidence level in case of analytics event, UE or list of UEs, service or service profile, area and time of interest.

[0072] In a further step 303, the TLSSP 320 (acting as PMC 384) discovers the PMR 330 or the list of PMP 340 entities providing the pre-trained ML models in a given service area.

[0073] Without the existence of the PMR 330, the entities providing pre-trained ML models needs to be discovered at the TLSSP 320 based on a list of pre-configured entities/platforms or based on querying PMPs 340 which in the past have provided pre-trained ML models for other tasks, or via querying the PMP entities 340 with MTME functionality in other enablers.

[0074] In a further step 304, the TLSSP 320 (acting as PMC 384) sends to the PMR 330 a request for provide all pre-trained ML model information which are allowed to be exposed. Such a request can be for all or a subset of ML models in a given area or for a given analytics task or for a given set of ML model task/job IDs, or for a given set of features/ context information or for a given service/service profile or set of UEs.

[0075] In a further step 305, the PMR 330 authorizes the request and may also interact with ChaF 350 to check the charging information for each of the pre-trained ML models.

[0076] In a further step 306, the PMR 330 responds to the TLSSP 320 (acting as PMC 384) with information for the list of pre-trained ML models based on the request.

[0077] In a further step 307, the TLSSP 320 identifies the similar models to be used as pre-trained ML models for transfer learning and may also rate them and evaluate the applicability and possible benefit.

[0078] The criteria for the identification may include the following: similar features / data requirements with the requested task; similarity of tasks (e.g., similar analytics type); similarity of ML methods used; whether a pre-trained ML model has been used in the past for similar tasks/purposes as the requested task; previous rating/ evaluation of the ML model based on feedback from other PMCs; a reputation/rating of the source of the pre-trained model (i.e., or the vendor, server, operator, edge provider).

[0079] In a further step 308, the TLSSP 320 sends one or more proposed pre-trained ML model information to the TLSSC 310, optionally with some rating or some context information to either allow the consumer to select based on the proposal or to just notify on the selected model(s).

[0080] Figure 4 illustrates an example 400 of messaging flow for ADAE triggered TL enablement in accordance with aspects of the present disclosure.

[0081] In this example, the entities involved are entities defined in 3GPP SA6TR 23.700-82 and SEAL ADAES (in 3GPP TS 23.434 titled “service enabler architecture layer for verticals (SEAL); functional architecture and information flows”, and 3GPP TS 23.436). The mapping with the above roles will be briefly introduced prior to a discussion of the messaging flow.

[0082] The TLSSC 410 is ADAES with MTME capability who is requiring to identify from where and how to get the pre-trained ML model.

[0083] The TLSSP 420 is an AI/ML Enablement Server or functionality/service module in the server for providing TL enablement.

[0084] The PMP 440 can be a VAL server or another AI/ML Enablement Server or another platform capability with MTME capability (e.g., another ADAES).

[0085] The PMC is the same as TLSSC or the TLSSP and can be ADAES or AI/ML Enablement Server or VAL server.

[0086] The PMR 430 is an A-ADRF or CCF or an edge repository.

[0087] In a first step 401, the ADAES with MTME capability 410 sends a request to the AIML Enablement Server (with TL enabler capability) 420 to provide support for discovering and selecting the appropriate pre-trained ML model for an analytics ID. This request may include the analytics ID of interest, and also the application service ID/profile for which this applies, the UE/list of UEs for which this applies, the area and time of interest and validity, information to assist the selection such as the required features/ data requirements, environment, confidence level, type of TL, method / algorithms to be used, permissions and restrictions for the model to be selected (e.g., from other vendors, operators), the KPI for the delivery of pre-trained model (e.g. maximum delay) and if there is some preference on a list of ML models or sources / platforms to be used.

[0088] In a further step 402, the AI/ML Enablement Server with TL enabler capability 420 discovers the repository 430 or directly the possible MTME sources 440 which can provide a pre-trained ML model for this request. Such entities can be VAL servers or other ADAES or other AI/ML Enablement Servers from the same or other platforms.

[0089] The discovery of the entities 430, 440 can happen using CAPIF (interaction with CCF to discover A-ADRF or the other Servers).

[0090] In a further step 403, the AI/ML Enablement Server 420 sends a request message to receive information on the pre-trained ML models available in the given service area. This request message is sent to the repository 430 or directly to the sources of the base models 440. This request may include the criteria of step 401 or a subset thereof.

[0091] In a further step 404, the AI/ML Enablement Server 420 receives, based on the request, the information on the available pre-trained ML models together with context information to assist the selection.

[0092] In a further step 405, the AI/ML Enablement Server 420 evaluates with the optional support of A-ADRF 430 whether the pre-trained ML models are applicable to the

analytics ID using certain criteria (as hereinbefore described with respect to Figure 3). This can be assisted using data/stats from previous utilization of these models for the certain analytics type.

[0093] The AI/ML Enablement Server 420 may rate or set a weight to the pre-trained ML model or the source of the pre-trained ML model. This rate may be for the applicability to a certain analytics ID (i.e., <analytics ID, rate 90%>). 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 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 or the source.

[0094] Based on the evaluation (which can be based on the rating), the AI/ML Enablement Server 420 determines one or more pre-trained ML models to be used for the analytics ID.

[0095] In a further step 406, the AI/ML Enablement Server 420 sends to the ADAES 410 the information for the selected/candidate pre-trained ML models (ID, profile, address to download) and optionally the pre-trained ML models themselves if this entity has it. Also, this may include the rating/weight for the pre-trained ML model if the ADAES 410 needs to decide among a list of them.

[0096] In a further step 407, the ADAES 410 may fetch the selected models from the repository 430 or directly from the corresponding MTME entities 440.

[0097] Figure 5 illustrates an example 500 of messaging flow for VAL triggered TL enablement in accordance with aspects of the present disclosure.

[0098] In this example 500, the entities involved are entities defined in 3GPP SA6 TR 23.700-82 and SEAL ADAES (3GPP TS 23.434, 3GPP TS 23.436). The mapping with the above roles will be briefly introduced before discussing the messaging flows.

[0099] The TLSSC 510 is a VAL server that is requiring to identify from where and how to get the pre-trained ML model.

[0100] The TLSSP 520 is an AI/ML Enablement Server or functionality/service module in the server for providing TL enablement.

[0101] The PMP 540 can be a VAL server or another AI/ML Enablement Server or another platform capability with MTME capability (e.g., another ADAES).

[0102] The PMR 530 is an A-ADRF or CCF or an edge repository.

[0103] In a first step 501, the VAL server 510 sends a request to the AI/ML Enablement Server (with TL enabler capability) 520 to provide support for discovering and selecting the appropriate pre-trained ML model for a given ML model task or for a given ML model profile. This request may include the task ID/profile or the model profile, and also the application service ID/type for which this applies, the UE/list of UEs for which this applies, the area and time of interest and validity, information to assist the selection such as the required features/ data requirements, environment, confidence level, type of TL, method / algorithms to be used, permissions and restrictions for the model to be selected (e.g., from other vendors, operators), the KPI for the delivery of pre-trained model (e.g., maximum delay) and if he has some preference on a list of ML models or sources / platforms to be used. The ML model profile can be a set of attributes which identify similar models.

[0104] In a further step 502, the AI/ML Enablement Server with TL enabler capability 520 discovers the repository 530 or directly the possible MTME sources 540 which can provide a pre-trained ML model for this request. Such entities can be VAL servers or other ADAES or other AIML Enablement Servers from the same or other platforms.

[0105] This step 502 also includes the mapping of the requested ML model profile to a list of ML task IDs which are applicable, so as to request the availability of pre-trained ML models only for the selected list of tasks under this profile.

[0106] If the ML model profile is not used, AI/ML Enablement Server 520 searches for all ML models in a given service area or based on the requested characteristics.

[0107] In a further step 503, the AI/ML Enablement Server 520 sends a request message to receive information on the pre-trained ML models available in the given service area or under the same ML model profile or list of ML tasks. This request message is sent

to the repository 530 or directly to the sources 540 of the base ML models. This request includes the criteria of step 501 or a subset thereof.

[0108] In a further step 504, the AI/ML Enablement Server 520 receives based on the request the information on the available pre-trained ML models together with context information to assist the selection.

[0109] In a further step 505, the AI/ML Enablement Server 520 evaluates with the optional support of A-ADRF 530 whether the pre-trained ML models are applicable to the ML model task or the ML model profile using certain criteria (mentioned in high level solution). This can be assisted using data/stats from previous utilization of these models for the certain model profiles or tasks.

[0110] The AI/ML Enablement Server 520 may rate or set a weight to the pre-trained ML model or the source of the ML model. This rate may be for the applicability to a certain analytics ID (i.e., <model profile ID, rate 90%> or <task ID, rate 0.98>).

[0111] Based on the evaluation (which can be based on the rating), the AI/ML Enablement Server 520 determines one or more candidate/selected pre-trained ML models to be used for the task or profile.

[0112] In a further step 506, the AI/ML Enablement Server 520 sends to the VAL server 510 the information for the selected pre-trained ML models (ID, profile, address to download) and optionally the trained ML models themselves if this entity has it. Also, this may include the rating/weight for the pre-trained ML model if the VAL server 510 needs to decide among a list of them.

[0113] In a further step 507, the VAL server 510 may fetch the selected pre-trained ML models from the repository 530 or directly from the corresponding MTME entities 540.

[0114] 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.

[0115] 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.

[0116] 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.

[0117] 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 as 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.

[0118] 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 methods disclosed herein. The UE 600 may be a UE 380 or 390 of Figure 3.

[0119] 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.

[0120] 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.

[0121] 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 and/or processing the demodulated signal to receive the transmitted data.

[0122] 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.

[0123] 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).

[0124] 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).

[0125] 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.

[0126] 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.

[0127] The memory 704 may include one or more caches (e.g., memory local to or included in the processor 700) or other memory, such as 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).

[0128] 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.

[0129] 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.

[0130] 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 of the methods disclosed herein.

[0131] 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.

[0132] 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.

[0133] 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.

[0134] 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 as 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.

[0135] 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 here. The NE 800 may be a NE 310, 320, 330, 340, 350, 360, 370 of Figure 3. The NE 800 may be a NE 410, 420, 430, 440 of Figure 4. The NE 800 may be an NE 510, 520, 530, 540 of Figure 5.

[0136] 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.

[0137] 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.

[0138] 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 and/or processing the demodulated signal to receive the transmitted data.

[0139] 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.

[0140] 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 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.

[0141] At 902, the method may include receiving, from a second network entity, a first request to identify at least one pre-trained machine learning ‘ML’ model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model. 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 NE as described with reference to Figure 8.

[0142] At 904, the method may include discovering, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models. 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 NE as described with reference to Figure 8.

[0143] At 906, the method may include obtaining, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria. 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 NE as described with reference to Figure 8.

[0144] At 908, the method may include providing, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task. 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 NE as described with reference to Figure 8.

[0145] Figure 10 illustrates a flowchart of a method in accordance with aspects of the present disclosure. The operations of the method may be implemented by a NE as described herein. In some implementations, the NE may execute a set of instructions to control the function elements of the NE to perform the described functions.

[0146] At 1002, the method may include sending, to a first network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model. The operations of 1002 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1002 may be performed by a NE as described with reference to Figure 8.

[0147] At 1004, the method may include receiving, from the first network entity, a second information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task. The operations of 1004 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1004 may be performed by a NE as described with reference to Figure 8.

[0148] 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.

[0149] At 1102, the method may include receiving, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at least one pre-trained ML model. 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.

[0150] At 1104, the method may include determining, based on the one or more criteria, a first information of one or more available pre-trained ML models. 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.

[0151] At 1106, the method may include sending, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models. 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.

[0152] 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.

[0153] The disclosure herein provides a first 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 first network entity to: receive, from a second network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; discover, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models; obtain, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and provide, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0154] 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.

[0155] 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. The pre-trained ML model may also be referred to as a base ML model. 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.

[0156] 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. A key issue becomes how to support transfer learning at application enablement layers. More specifically, a problem arises with how to identify whether a pre-trained model can be used for another ML task and how to ensure that such a solution will be beneficial. A further problem arises with how to configure transfer learning enabled entities for the distribution of pre-trained models.

[0157] Current 3GPP solutions do not consider transfer learning techniques, hence the identification and selection of transfer learning entities and pre-trained models is not supported.

[0158] The disclosure herein provides for the identification of available pre-trained models for transfer learning for an ML model task. The disclosure herein further provides for selection of the available pre-trained models based on criteria. The disclosure herein further provides for configuration of the entities involved in the distribution of pre-trained ML models for transfer learning (i.e., the source of the pre-trained model, the destination of the pre-trained model, and the coordinator/registry). Authorization and charging aspects are also provided.

[0159] The solutions described herein utilize a middleware transfer learning support service producer (TLSSP) to provide the identification and selection of the pre-trained ML model for the ML model task.

[0160] The solutions described herein provide enhancement of existing application layer registries for ML models. Such solutions enable the benefits of transfer learning within 3GPP systems.

[0161] The first, second and third network entities may be of a wireless communication system. For instance, the first network entity may be an application entity of a wireless communication system; the second network entity may be an application entity of a

wireless communication system; and/or the third network entity may be an application entity of a wireless communication system.

[0162] The third network entity may be a Pre-trained Model Producer (PMP). This entity can be the ML model provider or the ML model training entity that has produced the pre-trained model. Examples of such an entity include a VAL server or an AIML Enablement Server or an ADAES with MTME capability. This entity may be a service module or capability or functionality of the above entities.

[0163] The third network entity may be a pre-trained model Registry (PMR). This registry can be optional and can be implemented as A-ADRF, an edge or cloud repository, or a CAPIF Core Function (CCF).

[0164] A Pre-trained Model Consumer (PMC) is also introduced herein. This entity consumes the pre-trained ML model. This entity will use the pre-trained model to perform its own training on top and can operate as a VAL server or a further AIML Enablement Server or a further ADAES with MTME capability, or even a 3GPP network analytics function.

[0165] The first network entity may be a Transfer Learning Support Service Producer (TLSSP). The TLSSP may be the same entity as the PMC (however it can be different if, for example, the PMC is the VAL server (TLSSC) and it requires from AIML Enablement Server (TLSSP) to provide support for identifying the pre-trained model for a given task). The TLSSP can be for instance an AIML Enablement Server or an ADAES with MTME functionality.

[0166] The second network entity may be referred to herein as a Transfer Learning Support Service Consumer (TLSSC). The TLSSC has the role of requesting the support service (TLSSP) for deciding on the candidate available pre-trained ML models and selecting pre-trained ML model using certain criteria. Such an entity may operate as a VAL server. Such an entity may be an 3GGP SA6 enabler function. The reason for requiring such a service is the fact that this entity doesn’t have the permission and required capability and knowledge to judge what is the best pre-trained model to be used for a certain task and

consumes the service from the producer (TLSSP) which is authorized to perform the selection.

[0167] Also referred to herein is Charging Functionality (ChaF) for TLSS and/or Pre-trained Model utilization (allowing CRUD operations in pre-trained model). Such functionality can be part of OAM or part of the SEAL/EDGE/CAPIF platform provider.

[0168] The second information may be based at least in part on the first information. The receiving of the first request may be referred to herein as receiving of an application requirement.

[0169] The one or more criteria may comprise context information for identifying the at least one pre-trained ML model, wherein the context information comprises at least one of: one or more required features of a pre-trained ML model; one or more training data requirements; a required confidence level for analytics; a required type of transfer learning; one or more user equipment 'UE' associated with the ML model task; a service or service type associated with the ML model task; an area and/or time of interest associated with the ML model task; an environment associated with the ML model task; a required type of ML algorithm; one or more permissions for a pre-trained ML model; one or more restrictions for a pre-trained ML model; a key performance indicator 'KPI' for delivery of a pre-trained ML model.

[0170] The one or more criteria may comprise at least one of: an identifier for analytics; an identifier for the ML model task; an identifier for an application service; an identifier for a service profile; an ML model profile; or an ML model task profile.

[0171] The one or more criteria may comprise the ML model profile, wherein the at least one processor may be configured to cause the first network entity to discover, based on the one or more criteria, the third network entity by causing the first network entity to: map the ML model profile to a list of identifiers of ML model tasks; and discover the third network entity based on the mapping.

[0172] The one or more criteria may comprise the ML model task profile, wherein the at least one processor may be configured to cause the first network entity to discover, based on the one or more criteria, the third network entity, by causing the first network entity to:

map the ML model task profile to a list of identifiers of pre-trained ML models; and discover the third network entity based on the mapping.

[0173] The at least one processor may be configured to cause the first network entity to obtain, from the third network entity, the first information, by causing the first network entity to: send, to the third network entity, a second request for available pre-trained ML models, wherein the second request comprises the one or more criteria for identifying the at least one pre-trained ML model; and receive, from the third network entity, a second response comprising the first information.

[0174] The at least one processor may be configured to cause the first network entity to evaluate the one or more available pre-trained ML models as pre-trained ML models for the ML model task.

[0175] The at least one processor may be configured to cause the first network entity to evaluate the one or more available pre-trained ML models by causing the first network entity to: rate the one or more available pre-trained ML models, thereby generating a rating parameter.

[0176] The at least one processor may be configured to cause the first network entity to rate the one or more available pre-trained ML models based on one or more similarities to the ML model task, wherein the one or similarities comprise similarities based on: one or more features of the pre-trained ML model; training data requirements for the pre-trained ML model; a ML model algorithm of the pre-trained ML model; a prior usage of the pre-trained ML model; a prior rating of the pre-trained ML model; and a rating of a source of the pre-trained ML model.

[0177] The at least one processor may be configured to cause the first network entity to rate the one or more available pre-trained ML models for applicability to an identifier of: an analytics; an ML model task; or an ML model profile.

[0178] The second information may comprise the rating parameter.

[0179] The second information may comprise the one or more available pre-trained ML models.

[0180] The first network entity may comprise an artificial intelligence machine learning 'AIML' enablement server. The second network entity may comprise an application data analytics enabler server 'ADAES'; an edge application server, or a vertical application layer 'VAL' server. The third network entity may comprise a pre-trained model repository 'PMR' or a pre-trained model provider 'PMP', wherein optionally: the PMR is an application layer analytics data repository function 'AADRF', a common API framework core function 'CCF', an analytics data repository function 'ADRF', or an edge repository; and/or the PMP is a VAL server, an AIML enablement server or a model training and management entity 'MTME'.

[0181] The ML model task may be an analytics task.

[0182] There is provided a second 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 second network entity to: send, to a first network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; and receive, from the first network entity, a second information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0183] The first network entity may comprise an artificial intelligence machine learning 'AIML' enablement server. The second network entity may comprise an application data analytics enabler server 'ADAES'; or a vertical application layer 'VAL' server.

[0184] The one or more criteria may comprise context information for identifying the at least one pre-trained ML model, wherein the context information comprises at least one of: one or more required features of a pre-trained ML model; one or more training data requirements; a required confidence level for analytics; a required type of transfer learning; one or more user equipment 'UE' associated with the ML model task; a service or service type associated with the ML model task; an area and/or time of interest associated with the ML model task; an environment associated with the ML model task; a required type of ML algorithm; one or more permissions for a pre-trained ML model; one or more restrictions

for a pre-trained ML model; a key performance indicator ‘KPI’ for delivery of a pre-trained ML model.

[0185] The one or more criteria may comprise at least one of: an identifier for analytics; an identifier for the ML model task; an identifier for an application service; an identifier for a service profile; an ML model profile; or an ML model task profile.

[0186] The second information may comprise a rating parameter.

[0187] The second information may comprise the one or more available pre-trained ML models.

[0188] The at least one processor may be configured to cause the second network entity to fetch the one or more available pre-trained ML models (for instance from a PMR or PMP).

[0189] The at least one processor may be configured to cause the second network entity to determine a need for using transfer learning for a particular ML task, a particular job identifier, or a particular analytics identifier. In response to determining such a need, the second network entity may send the first request.

[0190] There is provided a third 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 third network entity to: receive, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at least one pre-trained ML model; determine, based on the one or more criteria, a first information of one or more available pre-trained ML models; and send, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models.

[0191] The first network entity may comprise an artificial intelligence machine learning ‘AIML’ enablement server. The third network entity may comprise a pre-trained model repository ‘PMR’ or a pre-trained model provider ‘PMP’, wherein optionally: the PMR is an application layer analytics data repository function ‘AADRF’, a common API

framework core function ‘CCF’, or an edge repository; and/or the PMP is a VAL server, an AIML enablement server or a model training and management entity ‘MTME’.

[0192] The one or more criteria may comprise context information for identifying the at least one pre-trained ML model, wherein the context information comprises at least one of: one or more required features of a pre-trained ML model; one or more training data requirements; a required confidence level for analytics; a required type of transfer learning; one or more user equipment ‘UE’ associated with the ML model task; a service or service type associated with the ML model task; an area and/or time of interest associated with the ML model task; an environment associated with the ML model task; a required type of ML algorithm; one or more permissions for a pre-trained ML model; one or more restrictions for a pre-trained ML model; a key performance indicator ‘KPI’ for delivery of a pre-trained ML model.

[0193] The one or more criteria may comprise at least one of: an identifier for analytics; an identifier for the ML model task; an identifier for an application service; an identifier for a service profile; an ML model profile; or an ML model task profile.

[0194] There is provided a method performed by a first network entity, comprising: receiving, from a second network entity, a first request to identify at least one pre-trained machine learning ‘ML’ model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; discovering, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models; obtaining, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and providing, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0195] The second information may be based at least in part on the first information.

[0196] The first network entity may comprise an artificial intelligence machine learning ‘AIML’ enablement server. The second network entity may comprise an application data analytics enabler server ‘ADAES’; or a vertical application layer ‘VAL’ server. The third

network entity may comprise a pre-trained model repository ‘PMR’ or a pre-trained model provider ‘PMP’, wherein optionally: the PMR is an application layer analytics data repository function ‘AADRF’, a common API framework core function ‘CCF’, or an edge repository; the PMP is a VAL server, an AIML enablement server or a model training and management entity ‘MTME’.

[0197] The one or more criteria may comprise context information for identifying the at least one pre-trained ML model. The context information may comprise at least one of: one or more required features of a pre-trained ML model; one or more training data requirements; a required confidence level for analytics; a required type of transfer learning; one or more user equipment ‘UE’ associated with the ML model task; a service or service type to associated with the ML model task; an area and/or time of interest associated with the ML model task; an environment associated with the ML model task; a required type of ML algorithm; one or more permissions for a pre-trained ML model; one or more restrictions for a pre-trained ML model; a key performance indicator ‘KPI’ for delivery of a pre-trained ML model.

[0198] The one or more criteria may comprise at least one of: an identifier for analytics; an identifier for the ML model task; an identifier for an application service; an identifier for a service profile; an ML model profile; or an ML model task profile.

[0199] The discovering, based on the one or more criteria, the third network entity, may comprise: mapping the ML model profile to a list of identifiers of ML model tasks; and discovering the third network entity based on the mapping.

[0200] The discovering, based on the one or more criteria, the third network entity, may comprise mapping the ML model task profile to a list of identifiers of pre-trained ML models; and discovering the third network entity based on the mapping.

[0201] The obtaining, from the third network entity, the first information, may comprise: sending, to the third network entity, a second request for available pre-trained ML models, wherein the second request comprises the one or more criteria for identifying the at least one pre-trained ML model; and receiving, from the third network entity, a second response comprising the first information.

[0202] The method may comprise evaluating the one or more available pre-trained ML models as pre-trained ML models for the ML model task.

[0203] The evaluating the one or more available pre-trained ML models may comprise rating the one or more available pre-trained ML models, thereby generating a rating parameter.

[0204] The rating may comprise rating the one or more available pre-trained ML models based on one or more similarities to the ML model task, wherein the one or similarities comprise similarities based on: one or more features of the pre-trained ML model; training data requirements for the pre-trained ML model; ML model algorithm of the pre-trained ML model; a prior usage of the pre-trained ML model; a prior rating of the pre-trained ML model; and a rating of a source of the pre-trained ML model.

[0205] The rating may comprise rating the one or more available pre-trained ML models for applicability to an identifier of: an analytics; an ML model task; or an ML model profile.

[0206] The second information may comprise the rating parameter.

[0207] The second information may comprise the one or more available pre-trained ML models.

[0208] The ML model task may be an analytics task.

[0209] There is further provided a method performed by a second network entity, comprising: sending, to a first network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; and receiving, from the first network entity, a second information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

[0210] There is further provided a method performed by a third network entity, comprising: receiving, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at

least one pre-trained ML model; determining, based on the one or more criteria, a first information of one or more available pre-trained ML models; and sending, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models.

[0211] Transfer Learning is used to support an entity to ease the training process (i.e., if conventional ML training is not possible or if the data and time to train is not sufficient). For TL enablement in 3GPP systems focusing more on the application layer, the following problems arise: how to identify whether a pre-trained model can be used for another task (i.e., a new ML job) and how to ensure that such solution will have benefits; and how to configure the TL-enabled entities for supporting the distribution of the pre-trained model.

[0212] The disclosure herein describes the identification of candidate pre-trained ML models and their selection based on criteria. The disclosure herein also describes the configuration of involved entities for the distribution (i.e., the source of pre-trained ML models, the destination, the coordinator/registry). The disclosure herein also includes the authorization and charging aspects. Such solutions use a middleware (TLSSP) to provide the identification and selection of the pre-trained ML model for the requested ML model task.

[0213] Current solutions do not consider transfer learning techniques (i.e., the use of a pre-trained ML model from one task to another). Hence the identification and selection of the entities and pre-trained ML models is not currently supported in 3GPP systems.

[0214] There is provided, a method (at, for instance, a TL enabler entity) for supporting the discovery of transfer learning capabilities for one or more ML model tasks, the method comprises: receiving an application requirement from a first application entity to identify at least one pre-trained ML model for an ML model task, wherein the application requirement comprises a set of criteria for identifying the at least one pre-trained ML model; discovering a second application entity to provide information on the availability of pre-trained ML models based on the set of criteria of the application requirement; obtaining from the second application entity information on one or more pre-trained ML models based on the set of criteria of the application requirement; selecting at least one pre-trained ML model from the set of obtained pre-trained ML models (based on, for instance, an

evaluation of the obtained pre-trained ML models); and sending information on the at least one selected pre-trained ML model to the application.

[0215] The ML model task may be an analytics task identified by an analytics identity.

[0216] The application requirement may comprise a requirement as described with respect to the examples herein.

[0217] The discovering may comprise: sending a request to a second application entity for receiving information; and receiving a response from the second application entity.

[0218] The second application entity may be a ML model repository.

[0219] The second application entity may be a pre-trained ML model producer entity (i.e., a VAL entity, an ADAES, or a MTME).

[0220] The method may further comprise rating the obtained pre-trained models with respect to the set of criteria.

[0221] The set of criteria may comprise one or more criteria as described with respect to the examples herein.

[0222] The application requirement may comprise an ML model profile or ML model task profile.

[0223] The method may further comprise mapping the ML model profile or ML model task profile to a set of ML model task identities.

[0224] The method may further comprise mapping the ML model profile or ML model task profile to one or more pre-trained ML model identities.

[0225] 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.

[0226] 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.

[0227] 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”.

[0228] The following abbreviations are used herein: TL, Transfer Learning; MDA, Management Domain Analytics; AIML , Artificial Intelligence / Machine Learning; ADAES , Application Data Analytics Enabler Server; 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; and ChaF, Charing Function.

CLAIMS

1. A first 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 first network entity to:

receive, from a second network entity, a first request to identify at least one pre-trained machine learning 'ML' model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model;

discover, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models;

obtain, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and

provide, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

2. The first network entity of claim 1, wherein the one or more criteria comprise context information for identifying the at least one pre-trained ML model, wherein the context information comprises at least one of:

one or more required features of a pre-trained ML model;

one or more training data requirements;

a required confidence level for analytics;

a required type of transfer learning;

one or more user equipment 'UE' associated with the ML model task;

a service or service type associated with the ML model task;

an area and/or time of interest associated with the ML model task;

an environment associated with the ML model task;

a required type of ML algorithm;

one or more permissions for a pre-trained ML model;

one or more restrictions for a pre-trained ML model;

a key performance indicator ‘KPI’ for delivery of a pre-trained ML model.

3. The first network entity of any preceding claim, wherein the one or more criteria comprise at least one of:

an identifier for analytics;

an identifier for the ML model task;

an identifier for an application service;

an identifier for a service profile;

an ML model profile; or

an ML model task profile.

4. The first network entity of claim 3, wherein the one or more criteria comprise the ML model profile, wherein the at least one processor is configured to cause the first network entity to discover, based on the one or more criteria, the third network entity by causing the first network entity to:

map the ML model profile to a list of identifiers of ML model tasks; and

discover the third network entity based on the mapping.

5. The first network entity of claim 3, wherein the one or more criteria comprise the ML model task profile, wherein the at least one processor is configured to cause the first network entity to discover, based on the one or more criteria, the third network entity, by causing the first network entity to:

map the ML model task profile to a list of identifiers of pre-trained ML models; and

discover the third network entity based on the mapping.

6. The first network entity of any preceding claim, wherein the at least one processor is configured to cause the first network entity to obtain, from the third network entity, the first information, by causing the first network entity to:

send, to the third network entity, a second request for available pre-trained ML models, wherein the second request comprises the one or more criteria for identifying the at least one pre-trained ML model; and

receive, from the third network entity, a second response comprising the first information.

7. The first network entity of any preceding claim, wherein the at least one processor is configured to cause the first network entity to evaluate the one or more available pre-trained ML models as pre-trained ML models for the ML model task.

8. The first network entity of claim 7, wherein the at least one processor is configured to cause the first network entity to evaluate the one or more available pre-trained ML models by causing the first network entity to:

rate the one or more available pre-trained ML models, thereby generating a rating parameter.

9. The first network entity of claim 8, wherein the at least one processor is configured to cause the first network entity to rate the one or more available pre-trained ML models based on one or more similarities to the ML model task, wherein the one or similarities comprise similarities based on:

one or more features of the pre-trained ML model;

training data requirements for the pre-trained ML model;

a ML model algorithm of the pre-trained ML model;

a prior usage of the pre-trained ML model;

a prior rating of the pre-trained ML model; and

a rating of a source of the pre-trained ML model.

10. The first network entity of any one of claims 8-9, wherein the at least one processor is configured to cause the first network entity to rate the one or more available pre-trained ML models for applicability to an identifier of:

an analytics;

an ML model task; or

an ML model profile.

11. The first network entity of any one of claims 8-10, wherein the second information comprises the rating parameter.

12. The first network entity of any preceding claim, wherein the second information comprises the one or more available pre-trained ML models.

13. The first network entity of any preceding claim, wherein:

the first network entity comprises an artificial intelligence machine learning ‘AIML’ enablement server;

the second network entity comprises an application data analytics enabler server ‘ADAES’; an edge application server, or a vertical application layer ‘VAL’ server; and/or

the third network entity comprises a pre-trained model repository ‘PMR’ or a pre-trained model provider ‘PMP’, wherein optionally:

the PMR is an application layer analytics data repository function ‘AADRF’, a common API framework core function ‘CCF’, an analytics data repository function ‘ADRF’, or an edge repository;

the PMP is a VAL server, an AIML enablement server or a model training and management entity ‘MTME’.

14. The first network entity of any preceding claim, wherein the ML model task is an analytics task.

15. A second 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 second network entity to:

send, to a first network entity, a first request to identify at least one pre-trained machine learning ‘ML’ model for transfer learning, the transfer learning for a ML model

task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model; and

receive, from the first network entity, a second information indicating one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

16. The second network entity of claim 15, wherein:

the first network entity comprises an artificial intelligence machine learning 'AIML' enablement server; and/or

the second network entity comprises an application data analytics enabler server 'ADAES'; or a vertical application layer 'VAL' server.

17. A third 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 third network entity to:

receive, from a first network entity, a second request for available pre-trained ML models, wherein the second request comprises one or more criteria for identifying at least one pre-trained ML model;

determine, based on the one or more criteria, a first information of one or more available pre-trained ML models; and

send, to the first network entity, a second response comprising the first information of the one or more available pre-trained ML models.

18. The third network entity of claim 17, wherein:

the first network entity comprises an artificial intelligence machine learning 'AIML' enablement server; and/or

the third network entity comprises a pre-trained model repository 'PMR' or a pre-trained model provider 'PMP', wherein optionally:

the PMR is an application layer analytics data repository function 'AADRF', a common API framework core function 'CCF', or an edge repository;

the PMP is a VAL server, an AIML enablement server or a model training and management entity ‘MTME’.

19. A method performed by a first network entity, comprising:

receiving, from a second network entity, a first request to identify at least one pre-trained machine learning ‘ML’ model for transfer learning, the transfer learning for a ML model task, wherein the first request comprises one or more criteria for identifying the at least one pre-trained ML model;

discovering, based on the one or more criteria, a third network entity for indicating the availability of pre-trained ML models;

obtaining, from the third network entity, a first information of one or more available pre-trained ML models based on the one or more criteria; and

providing, to the second network entity, a second information indicating the one or more available pre-trained ML models as pre-trained ML models for the transfer learning for the ML model task.

20. The method of claim 19, wherein:

the first network entity comprises an artificial intelligence machine learning ‘AIML’ enablement server;

the second network entity comprises an application data analytics enabler server ‘ADAES’; or a vertical application layer ‘VAL’ server; and/or

the third network entity comprises a pre-trained model repository ‘PMR’ or a pre-trained model provider ‘PMP’, wherein optionally:

the PMR is an application layer analytics data repository function ‘AADRF’, a common API framework core function ‘CCF’, or an edge repository;

the PMP is a VAL server, an AIML enablement server or a model training and management entity ‘MTME’.