3GPP TSG-RAN3 #112-e R3-212688

Online, 17th – 27th May 2021

Agenda Item: 18.2

Source: Ericsson (moderator)

Title: SoD Data Collection Principles and Framework

Document for: Approval

# Introduction

**CB: # 45\_DataColl\_PrincDef**

**- Chair: suggest to structure discussion around 3 areas, to become 3 TPs, splitting work among companies:**

**1) updates/corrections to current text (if any / if agreeable) (merging any agreeable parts from e.g. 1632, 1681, 1682, 2027,1615) (NEC?);**

**2) high level principles and definitions TP (merging any agreeable parts from e.g. 1632, 1754, 2300, 2373) (Intel?);**

**3) functional framework TP (merging any agreeable parts from e.g. 1632, 2178, 2189, 2299, 2314, 2372, 2503, 2522) (HW?/E///?)**

(E/// - moderator)

# For the Chairman’s Notes

Propose the following:

R3-21xxxa, R3-21xxxc merged

R3-21xxxc rev [in xxxg] – agreed

R3-21xxxd rev [in xxxh] – agreed

R3-21xxxe rev [in xxxi] – agreed

R3-21xxxf rev [in xxxj] – endorsed

Propose to capture the following:

**Agreement text…**

**Agreement text…**

**WA: carefully crafted text…**

Issue 1: no consensus

**Issue 2: issue is acknowledged; need to further check the impact on xxx. May be possible to address with a pure st2 change. To be continued…**

# Discussion

The papers in AI18.2 discuss a variety of proposals covering different aspect of the study on Enhancement for Data Collection for NR and EN-DC SI.

The discussion will be structured as follows:

* In a first round of offline discussions, it is proposed to focus on proposals with good support in the submitted papers or proposals on aspects not yet tackled in the discussions but concerning existing TR content
* Depending on convergence, in a second round of discussions, new proposals (e.g. adding new functions) and TP proposals can be discussed and if possible agreed

## First Round of Discussions

### High Level Principles

In [1] additions have been proposed for the High Level Principles section in TR37.817. These principles have been readapted below, in light of the inputs provided in other papers in AI18.2.

List of proposed principles:

1. The process of training is up to implementation. The study should focus on exchange of information between the training function and other functions
2. The inference and training functions should be able to request, if needed, specific information to be used to execute or train the AI/ML algorithm and to avoid reception of unnecessary information.
3. The inference function should signal the outputs of the model only to nodes that have explicitly requested them (e.g. via subscription), or nodes that are subject to actions based on the ML output.
4. If the inference function provides output predictions, a corresponding indication of accuracy should be indicated to the nodes that request/subscribe to this information.

Principle 1 focusses on the process of training. Namely, the actual process of training should be left to implementation, while the standard should specify how training data are exchanged between the training function and other functions.

Principle 2 can be resumed to support for the data subscription techniques already used by e.g. the X2/Xn Resource Status Reporting procedures. This principle seems to be acknowledged in other papers like [2] and [3].

Principle 3 is a direct consequence of Principle 2.

Principle 4 is also supported in [2] and it concerns complementing predictions from an inference function with a prediction accuracy.

**Companies are invited to provide their view on whether principles 1 to 4 are agreeable for inclusion in Section 4.1**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Principles 1 to 4 can be agreed.  In particular principle 1 is an assumption following the same logic for which AI inference models are left to implementation, i.e. AI training models are also left to implementation.  Principles 2 and 3 are needed to avoid uncontrollable flows of data over common interfaces.  Principle 4 is needed to ensure that nodes subscribing to prediction outputs can make proper use of such prediction by knowing the prediction accuracy. |
| Deutsche Telekom | Principles 2 – 4 are fine with us.   * Question:   + Does the service request/subscription model mentioned in those principles require an SBA-based approach?   + If yes, we would support it.   The meaning of Principle 1 is somehow unclear. We agree that definition of the training process itself is implementation specific and does not require a fixed standardized approach, but it sets some requirements on delivery of input data and on final exchange of created models that has to be described. It is ok for us to focus in the study on exchange of information between the training function and other functions, but the training is a basic process of the whole framework and has to be covered in the description. Furthermore, there is a need to differentiate between offline and online training, as the latter one may have strong impact if performed in the RAN (e.g. via reinforcement learning). Also, initial offline training, even if performed in a 5GS domain outside of RAN (e.g. OAM), may have impact as it may rely on data collected by RAN. Requirements to both offline and online training can be covered via selected use cases. |
| Futurewei | Principals 1-3: agree.  Principal 4: disagree. Inference or prediction accuracy is not known at the inference function at the time of inference. Model performance can be calculated after receiving feedback(s) from the “Subject of action” and we recommend keeping the model performance calculation in the same functional block as “Model training” after it receives the feedback(s). |
| CATT | Agree with 1-3.  For 4, we think delivering the accuracy of prediction should be an optional feature. The accuracy is not always necessary for taking further action, especially for the case of load prediction. |
| Nokia | Principle 1: In our view the process of Training (as also the process of Inference) are up to implementation. However, data collection for the purpose of Training (and Inference) should be in the scope of the SI.  Principles 2, 3: Can be agreed.  Principle 4: It is unclear how prediction accuracy can be evaluated at inference since at that stage accuracy of the predictions is not known. This accuracy can be evaluated only after real measurements are received. Or is the meaning here accuracy with respect to how an ML model was trained? |
| NEC | 1. Agree. Process of training may be different for different ML models. The standard shall specify interfaces, procedures, messages, and information elements to support various training methods, preferably without limiting training data sources, ML models and algorithms.  2. Agree. However, this is protocol specific and may be too specific for the study stage. As mentioned, this problem is not specific to AI/ML and standard solutions (e.g., subscription) already exist.  3. Agree.  4. This may be beneficial in some cases. May be not needed in other cases. Looks like specific solution. Could be optional parameter based on subscription (see 2).  Is it always possible to generate predication accuracy?  Is it always needed? For example, most currently estimated values are reported without estimation accuracy value. |
| Samsung | Principle 1 is fine for us.  For principles 2 to 4, the detailed interface impact should be studied case by case. So we prefer to discuss these issues in future use case study. |
| Intel | Principle 1: It is also unclear to us what is “the process of training”. We agreed in R3-110e meeting that the detailed AI/ML algorithms/models for use cases are out of RAN3 scope. This is because, for each use case, the adopted ML algorithm is an implementation issue. From our understanding, the process of training may be impacted by three categories of ML models, i.e. supervised/unsupervised/reinforcement learning. For example, supervised/unsupervised learning may not require immediate feedback from action node. On the other hand, reinforcement learning requires an immediate performance and reward feedback reporting back to the online training node, so that a reinforcement learning model can be continuously optimized. In this case, the process of training may be impacted by whether a reward feedback will be provided to the training node.  Principle 2: we agree the intention of principle 2, while it is unclear in the proposal what should be requested, per data and exchanged information or per use case?  Principle 3: we agree with principle 3.  Principle 4: we agree that a “confidence level” or “accuracy” showing the accuracy of output from inference function should be supported. However, the output of the inference function can be predictions or actions directly.  Moreover, for each output from inference model, we also think a corresponding “validity time” should be introduced, indicating whether the output is still valid or not. If the time expires, the received function node should not use such information as it may be out-of-date.  Hence, we propose the principle to be updated as below:  **Principle 4: If the inference function provides output, a corresponding indication of accuracy and validity time should be indicated to the nodes that request/subscribe to this information.** |
| ZTE | For principle 1, in previous meeting, it is agreed that the detailed AI/ML algorithms/models for use cases are out of RAN3 scope. Moreover, we should not only focus on the exchange information between the training function and other functions, but also data collection between different nodes.  For principle 2 and 3, we think this is the interface enhancement for AI/ML. These two are fine for us, because it can be included in the AI Function Management as proposed in R3-212034.  For principle 4: this should be discussed case by case. Not all use case needs this feature. |
| vivo | P1-4 are agreeable.  As to P4, the accuracy or uncertainty can be derived by the training data volume, which can be informed to the actor, the subsequent action may rely on implementation. |
| CMCC | For P1, we already have the agreement of “the detailed AI/ML algorithms/models for use cases are out of RAN3 scope”. The proposal seems not needed, considering people may have different interpretation of the process of training, e.g., whether it includes the process of data collection and information exchange among nodes  Fine with P2 and P3  P4 may be case by case |
| AT&T | P1: We understand the training method itself may be up to implementation, but we are not sure if it can be fully transparent to the involved nodes. Depending on the algorithm and performance capabilities of the nodes, different data collection and information exchange may be required. As a result, there may be a need for configuration or identification of different training types or capabilities at the training function and other functions.  P2/P3: Ok with these as long as this does not preclude routing of information through intermediate nodes (e.g. via containers)  P4: This needs further study to understand how to exactly define an indication of accuracy for specific types of outputs and different use cases. |
| Lenovo, Motorola Mobility | Similar view as DT, principle 2-4 are fine to us, principle 1 needs further clarifying. |
| LGE | Agree on 2, 3, 4 |
| Qualcomm | P1: It is not clear on “process of training”. We should define the signaling for training host to receive the training request, collect required data and report the training result.  P2, P3: agree.  P4: OK, but the “indication of accuracy” should be optional IE to allow flexible implementation. |

### Functional Framework

Note: many papers provide proposals on the definitions to be added in section 3.1 of TR37817. The discussion will be structured to first converge on agreements concerning the elements forming the functional framework and then (in the second round of discussions) to converge on the exact definitions of such elements.

### “Actor + Subject of Action” or “Action”

A number of papers propose to merge the boxes “Actor” and “Subject of Action” into a single box. Such box is named in a number of papers “Action”, see [2] to [7]. Other papers name this single box differently, see [1] and [8], however the name should not influence the functional interpretation of the box, which is that of receiving outputs from an inference function and decide/execute actions depending on such inputs.

**Companies are invited to provide their view on whether the current “Actor” and “Subject of Action” in Figure 4.2-1 of TR37.817 can be merged into one function called “Action”.**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Agree. The “Action” hosts an ML assisted function and/or a function that receives the output of an AI/ML algorithm. Based on the ML output, the Actor may trigger actions directed to other entities or to itself |
| Deutsche Telekom | Ok to merge “Actor” and “Subject of actions” into a common block called “Action” as this is sufficient from a pure functional framework description.  Nevertheless, the description of the term “Action” is currently missing in TR 37.817 and should be added (see e.g. the proposal in R3-211632 [4]). |
| Futurewei | Agree with clarification.  If “Actor” and “Subjection of Action” are merged into one functional block, we recommend clarifying that the final action(s) to be executed may depend on further processing of the ML output or inference results. |
| CATT | No strong opinion. |
| Nokia | Agree. Keeping them separate could reflect different deployment options in the sense that actor and subject of action could be located in different entities. However, we support keeping the framework as simple as possible and thus we think that for the purpose of this SI a single box is sufficient. |
| NEC | Disagree.  The objective of this SI includes to study standardization impacts on the nodes, functions, and network interfaces in the current NG-RAN architecture “to convey the input/output data among network nodes or AI functions.”  If Actor is one NG-RAN node and Subject of action is another NG-RAN node, then Action is transmitted via Xn interface. If Actor is gNB-CU and Subject of action is gNB-DU, then Action is transmitted via F1 interface. In these examples Xn or F1 interfaces are impacted. If Actor and Subject of Action are in one box, such impact is missing in the Functional Framework. |
| Samsung | The detailed standard impact need to be studied case by case. The framework is just to provide a guidance or reference for further use case study.  No strong view. Separate blocks or merged block are both OK. |
| Intel | Agree to merge “Actor” and “Subject of Action” into one box. Considering “Action” is a movement, we suggest the name of this box should be “Actor”.  Besides, as mentioned by many companies, the “Actor” can allocate in one node or multiple nodes. We think it would be good to remain the sub-boxes of “Subject of Actions” within the “Actor” box. If “Subject of Action” is removed from the figure, we may need to add this description in the definition of “Actor” at least. |
| ZTE | Agree to merge “Actor” and “Subject of Action” into one box.  One box is enough to reflect the function of action. |
| vivo | Agree.  The framework based on the function description is clearer than that based on physical entity. |
| CMCC | No strong preference actually. For the sake of progress, we could go majority view that merging them in one box from functional point of view and not stuck on this point. If needed, we could add some note to clarify that “resulting action may be spread across several logical RAN nodes” |
| AT&T | Agree with NEC and Intel, there are cases where the “subject of action” still should be identified, even if captured within a single “Actor” box since they may not be the same logical RAN node. |
| Lenovo, Motorola Mobility | Agree, we don’t see the need to sperate them. |
| LGE | Agree |
| Qualcomm | Agree, if we change framework figure from entity based to function based (e.g. from “training **host**” to “training”). |

### Model performance feedback and Performance Feedback

A number of papers propose to remove the arow “Model performance feedback” going from Model Inference Host to Model Training Host, see [2], [3], [5], [8], [9]. As mentioned in [5] “Model Performance can be evaluated only after an action is taken”, hence most companies believe that the model inference function is not in a position to provide feedback concerning the model.

In some papers it is mentioned that the Model Performance Feedback is a set of performance metrics concerning how the model is “running”, e.g. consumed memory, processing power, response time (see [13]). However, these metrics are very much model specific and for that not an absolute indication of how well a model is running.

Following the same logic, most companies see the need of maintaining the Performance Feedback arrow from “Action” to “Data Sources”, given that such information is based on the actual KPIs that should be optimized by means of AI.

**Companies are invited to provide their view on whether the “Model performance feedback” from Model Inference Host to Model Training Host should be removed**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Agree.  The model inference function does not necessarily have an understanding of how good the model outputs are because actions and performance measurements following the action may be taken by different functions. |
| Deutsche Telekom | Disagree.  We see that link still as useful, as it provides model quality information from ML Inference function to ML Training function (note that this is a relation between functions, not between hosts as in current figure in TR 37.817). This is not related to possible RAN KPI feedback provided by Action function. The feedback link between ML Inference and ML Training function is especially useful to create a closed optimization loop in case of online training where both functions are hosted in RAN nodes. |
| Futurewei | Agree to remove “Model performance feedback” arrow between “Model inference” and “Model training”.  The model inference functional block does not have the knowledge of model performance at the inference time. The model performance can be calculated using the inference result together with the feedback(s) received from the “Subject of action” after executing the action.  We recommend letting the “Model training” functional block to calculate the model performance, thus the inference result should be sent back from “Model inference” to “Model training” for it to calculate model performance. |
| CATT | We think that it should be removed for at least the ML cases other than prediction.  For the case with prediction, the “Performance feedback” arrow seemingly does not cover providing of prediction result itself (**prediction result is not a consequence of action!**), but we think the data source or the model training entity need to get aware of the prediction result so that it can figure out when the accuracy of the model deteriorates, and then trigger model retraining.  Nevertheless the end point of this arrow can be toward the data source itself, and if the “Performance feedback” arrow itself is removed, this arrow should be removed as well. |
| Nokia | Agree to remove it. Model Performance can only be evaluated when an Action is taken. Since in general Inference and Action can be located in separate entities, Model Performance may not be available at Model Inference. |
| NEC | Maybe better not to remove.  Performance feedback is feedback after some actions are taken. Such feedback is generated by some parts of RAN. Examples are throughput, HO success, RLF, etc.  Model performance feedback is feedback that could be generated directly by ML model. Examples are processing and memory load and response time of Model Inference, some performance metrics, e.g., prediction accuracy etc.  Both types of feedback could be useful to study. |
| Samsung | It is better to keep “model performance feedback”.  “Model performance feedback” is transferring the model performance to provide the information for model training. This feedback is from the view of inference. When the model is no longer suitable for inference (e.g. prediction accuracy is too low), this feedback is to provide information to training to trigger retraining process. |
| Intel | Agree.  For online training/reinforcement training, the closed loop for model optimization/iteration requires feedback from the environment after certain action(s) is taken. Hence, the feedback which can be used for model optimization is generated by “Actor” node. The feedback from model inference to model training is not needed. |
| ZTE | “Model performance feedback” is essential.  If the output of the Model inference is the prediction (load prediction/UE trajectory prediction), there is no case where model performance is evaluated only after an action is taken”. When **the accuracy of prediction** is not good/too low, model inference itself should inform model training to retrain the ML model. |
| vivo | Better to keep it.  Some results of inference, e.g. load prediction, can be acquired by the model inference in real time. In this situation, the arrow “Model performance feedback” between “Model inference” and “Model training” can be used to feedback the performance. Further, model inference may feedback whether the model complexity is too high or not. |
| CMCC | There are basically two ways as shown in our paper, model inference generates the model performance feedback by comparing the inference results with the real data, or provide the inference results to other function, e.g., data sources, data sources will generate the model performance feedback to help the retraining. So it seems beneficial for at least prediction scenario. |
| AT&T | We prefer to keep it at this stage. For certain scenarios, it can be clarified if this feedback is always needed/available. |
| Lenovo, Motorola Mobility | Better let “data collection & preparation” provide the performance feedback.  In our view, taking prediction task as an example, the model performance feedback can be deduced by comparing the prediction result and measurement result, while “data collection&preparation” is the function collecting all kinds of data.  Besides, when the model performance feedback is provided, sometimes it will automatically trigger the model retraining. Thus, the retraining dataset could be provided together with the model performance feedback.  Considering above, we believe it’s more proper to let the “data collection & preparation” be responsible of providing model performance feedback together with the dataset for retraining. |
| LGE | We prefer to keep it from the time being |
| Qualcomm | Agree to remove it.  The inference host does not know the quality of its output by itself. So, it cannot provide good performance feedback to training.  As to the feedback on the resource consumption (memory, CPU/GPU/NPU …), it can be covered by the OAM of the inference platform. |

**Companies are invited to provide their view on whether the “Performance feedback” from “Action” to “Data Sources” should be kept**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Disagree.  The “Performance feedback” from “Action” to “Data Sources” is conceptually correct. However, there is no need to explicitly mark this arrow if Data Sources are defined as the function that also provides performance feedback. By removing this arrow the diagram would be simpler and there will not be the possible misunderstanding that information reaching “data sources” can only be made of what is explicitly shown as flowing into Data Sources. |
| Deutsche Telekom | Agree.  This feedback loop is required to describe the full optimization process. Please note that this is in line with the change of Data Sources to Data Collection & Preparation for a functional description. |
| Futurewei | Agree with modification (from “Performance feedback” to “Feedback”).  “Feedback” after executing the action is needed for 1) evaluating the model performance, 2) calculating rewards and others for RL setting. Our suggestion is to keep the information exchange arrow between “Subject of action” and “Data sources” to be clear.  However, there is no need to include “performance” in the wording of the feedback as the feedback may also include other attributes, e.g., new state that the “Subject of action” transitions to after the action.  We propose to change the current “Performance feedback” to “Feedback (state and performance)” for clarity reason or just “Feedback” with added text to clarify the feedback may include state, performance, and others. |
| CATT | No strong opinion. Slightly prefer to keep as it is. |
| Nokia | Agree with some modifications. Performance feedback is a natural way with which measurements, KPIs and other performance counters are being collected from the network. So, it is always present. In addition, Model Performance feedback is also available from Action towards Data Collection since this is where model performance can be evaluated. In our view, both feedbacks are needed from Action to Data Collection and having a feedback arrow is an intuitive way to express the feedback loop based on which, more data can be retrieved. We suggest though to change the naming from Model Performance feedback to “Feedback” to illustrate that this can be also performance feedback (or other feedback) used as an input to Data Collection to retrieve more measurements. |
| NEC | Better to keep. |
| Samsung | It is better to keep.  The performance feedback is to collect the actual network performance after applying AI-aided solution to describe the full optimization process. |
| Intel | Agree.  This feedback is essential to ML model continuous optimization and iteration, to both online training and offline training. |
| ZTE | Agree to keep.  Performance feedback indicates the network performance including KPIs, network measurements, etc, which can be collected to “Data Collection & Preparation” block for online training. |
| vivo | Agree to keep the “feedback”, whether the “performance” should be kept or not depends on the feedback content. |
| CMCC | Agree to keep, this shows the whole process |
| AT&T | Prefer to keep. |
| Lenovo, Motorola Mobility | Agree.  Since the performance feedback can be used later for retraining etc. We consider it beneficial to keep the performance feedback in the diagram. |
| LGE | Prefer to keep it |
| Qualcomm | Agree to keep.  This feedback is essential for model performance monitoring and model update. |

It is also discussed in a number of papers whether “Performance Feedback” should be explicitly shown from “Action” to “Model Training”.

**Companies are invited to provide their view on whether the “Performance feedback” from “Action” to “Model Training” should be introduced**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Disagree.  As explained above, “Performance feedback” will be part of data hosted by the Data Sources. If the Model Training function needs performance feedback concerning the model, the Model Training will request such data from Data Sources. An explicit arrow may also wrongly imply that such feedback is always provided to the Model Training, while it should only be provided if requested. |
| Deutsche Telekom | Disagree.  The ML Training function gets its input information especially via the Data Collection & Preparation function. Dependent on the need of the training process such data is prepared in a suitable way as e.g. not all info (e.g. KPIs) provided by Action function may be required for the training. |
| Futurewei | Disagree.  Our suggestion is to direct the feedback to the “Data sources” without introducing additional data exchange between the “Subject of action” and the “Model training” functions. |
| CATT | Disagree. |
| Nokia | No strong view. We think that direct Feedback from Action to Model Training can be useful in certain use cases but for the sake of progress we are fine to leave it out. |
| NEC | Agree to introduce.  Same NG-RAN node can function as Data source and Subject of action. Also, if there are several different AI models deployed in the NG-RAN, same NG-RAN node can be Actor for one model and Subject of action for another model. It can even be host for Model training and Model inference for other AI models depending on use cases and deployment scenarios.  In order not to limit the study of impacts on existing nodes and interfaces, it is proposed that both direct feedback and indirect (via data sources) is considered in the study.  Also, as mentioned in some papers some part of performance feedback is basically automatically coming from Data Sources, e.g., RLF report. As mentioned in other papers for reinforcement learning some specific performance feedback is generated by RAN that should go directly to Model Training. |
| Samsung | Prefer the framework without feedback from action to model training.  There are multiple factors to affect the network performance, such as channel condition, generated policy, equipment issues etc. If the network performance is poor, the analysis is required to be done to find out the reason firstly. When the problem is coming from AI model based on analysis, the model needs to trigger the retraining procedure, otherwise, the model-retraining should not be triggered. So the retraining procedure cannot be triggered directly by the network performance.  For the reinforcement learning or online training, the performance data can be one of the training inputs. For the current framework, the performance data can be obtained by training from data sources/collection as training data, which is more reasonable for the role of performance data for reinforcement learning or online training. |
| Intel | Disagree.  Agree with E/// that “ML Training host” can request data from “Data Collection & Preparation” Node. |
| ZTE | Agree to introduced.  Model training may be triggered to retrain the ML model when the model performance feedback is not good. Similarly, we cannot preclude the case where the output of the model inference is AI-generated decision. After the decision is adopted/configured in the Action, if the network performance feedback from Action is not good, Model training should also be triggered to retrain the ML model or update the ML model.  Some companies mentioned that “Model Performance can be evaluated only after an action is taken”. Hence, model training needs to obtain the performance feedback from action. |
| vivo | Agree to introduce.  The direct “Performance feedback” from “Action” to “Model Training” is beneficial for model enhancement. |
| CMCC | For the sake of progress, we are fine to leave it out |
| AT&T | Disagree |
| Lenovo, Motorola Mobility | Disagree,  Same view as DT |
| LGE | No strong view on this point |
| Qualcomm | Disagree.  The training host can get the performance data from “data source”. It is not necessary to have direct feedback to training host. |

### “Data Sources” and “Data Collection and Preparation”

A number of papers argue that the “Data Sources” block should be renamed “Data Collection and Preparation”, see [2], [4], [6], [8], [10]. As described in [4],

“*The block “Data sources” has to be renamed to “Data collection” to describe the related functionality, but as the function goes beyond plain collection of data – it has also to classify the data and prepare output data according to the need of following processing blocks (see also Fig. 2) – the name should be extended to “Data collection & preparation”.*”

However, a number of paper sustain that the process of “data preparation” is model specific and an assumption that model preparation can happen in functions outside the model inference would imply that there is knowledge about the model implementation outside the model inference function, see [1], [3], [5], [9]. As mentioned in [3]:

“*the data preparation always highly depends on the model design, and there is no common processing method. For example, the stopping criteria for selecting the best subset of features in Wrapper methods (e.g., forward selection, backward elimination, Bi-directional elimination) are usually pre-defined such as when the performance of the model decreases or a specific number of features has been achieved. Another example is the Embedded methods (e.g., regularization, tree-based methods), where the feature selection is blended as part of the learning algorithm thus having its own built-in feature selection methods.*”

**Companies are invited to provide their view on whether the “Data sources” block should be replaced by “Data collection and Preparation”**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Disagree.  Data preparation is highly dependent on the model design and it cannot be assumed that a function outside the Model Inference would be able to prepare data as required by a specific model implementation. Data preparation can be assumed as part of the training and inference functions. We propose to keep “Data Sources” with the following definition:   * Data source(s) is one or more entities that provide input data that is needed for model training and inference. Examples of input data may include measurements from UEs or different network entities, Performance feedback, ML feedback/output.   + Training Data: information needed for training the model.   + Inference Data: new information needed as an input for the inference host to provide a corresponding output. |
| Deutsche Telekom | Agree.  The agreement was to describe the framework from a functional perspective. Therefore, the term Data Sources does not fit. We need to introduce a function that collects data (from different sources) and prepares it in a suitable way as input for ML Training and ML Inference functions. Please note that there may be several instances of Data Collection & Preparation spread across the RAN nodes (or even implemented in other domains like 5GC or OAM). |
| Futurewei | Disagree to include data preparation step inside the data collection block as data preparation is AI/ML algorithm dependent and should be considered as implementation details.  We agree to change “Data sources” to “Data collection”. |
| CATT | Slightly prefer to keep as it is. |
| Nokia | We support to change Data Sources to Data Collection (Which would be in line with Model Training, Model Inference, and Action boxes rather than referring to “Hosts”). However, we do not support including Data Preparation in Data Collection. |
| NEC | Disagree.  We agree to rename “Data sources” to “Data collection.”  Data collection and Data preparation assume very different impacts on NG-RAN nodes and network functions.  Data collection is ML model independent while Data preparation is ML model dependent. |
| Samsung | The definition of “Data collection” has been captured in the TR 37.817, but what “preparation” would do is not clear and there is no common understanding for it. It is better to have a definition/common understanding of “data preparation” first. |
| Intel | Agree.  It is possible that the data can be requested by multiple ML training nodes for the same purpose. In this case, it would be nice to do some common data preparation in the “Data Collection” function, such as labeling/filtering, especially when “Data Collection” function is located in a centralized node. “ML Training” and “ML Inference” function node can either use the received data directly, or it can further process the data to fit ML algorithms in those function nodes. |
| ZTE | Agree to replace the “Data sources” by “Data collection & preparation”.  If we regard data preparation as the part of ML function, we are also fine with “Data Collection”. |
| vivo | Agree to modify the “Data sources” to “Data collection and Preparation” form function perspective.  Meanwhile, to enable the data collection with the ability to delete some unnecessary data can reduce the message load between data collection and model training. |
| CMCC | We would like to remind that this is the first release in RAN to discuss AI related topics which are indeed not expertise of the group. So in our view, we should start with simple framework. In this regard, we may not show data preparation in the figure although we know in implementation we do have some kind of preparation function. The data preparation may not even have any spec impact. |
| AT&T | We are not sure about the implications of data preparation outside of a ML-model specific context. At this stage we prefer to just change “data sources” to “data collection” |
| Lenovo, Motorola Mobility | Agree.  What we are drawing is a function diagram, and different functionalities could be located in the same node or not.  Comparing to assuming every Function (e.g. ML training, ML inference) has a sub-function of data collection and preparation, we prefer to draw the “data collection and preparation” as an individual function. We can of course discuss whether they belong to the same node case by case. |
| LGE | Revise it into “Data collection” is acceptable |
| Qualcomm | Agree, if we change the framework figure from entity/host based to function based. |

Also, a number of companies proposed to include functions such as “Data Preparation”, which are model implementation specific, inside the “Model Training” function, so to ensure that the information from “Data Sources” to “Model Training” remain model implementation independent (see for example [5] and [13].

**Companies are invited to provide their view on whether the “Data Preparation” function should be included in the “Model Training” function**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Agree.  This would allow for full standardization of the information from Data Sources to Model Training, without the need of exposing model implementation specific information |
| Deutsche Telekom | Disagree.  Both the ML Training and the ML Inference function may use a further internal preparation step if required. There is no restriction on that. But the Data Collection & Preparation function should allow a preparation process (classification/(pre-)selection) of data that should be finally delivered to the two functions mentioned before. The target is to reduce the amount of data to be delivered. |
| Futurewei | Agree, as data representation is AI/ML algorithm dependent and should be considered as implementation details. |
| CATT | Not needed.  We agree with the understanding that data preparation should be a step within the model training function, but we don’t think it needs to be shown in the figure. It is too natural. |
| Nokia | Agree.  Data Preparation is specific to an ML Model for which data is prepared. By including Data Preparation in the data collection we would possibly expose some details about the used ML model over the interfaces. We believe that a better place for Data Preparation is inside the Model Training and Model Inference, so that the details of preparation are kept implementation specific. For the same reason, we propose that Data is sent from Data Collection to Model Training and Model Inference (instead of the existing terminology “Training Data” and “Inference Data”). Training Data and Inference Data should be produced inside Model Training and Model Inference respectively and are the data prepared according to a specific ML model. |
| NEC | Agree.  This way Data collection block will be kept ML model independent and ML model dependent Data preparation will be inside ML model dependent Model training block. |
| Samsung | If “data preparation” means pre-processing collected data to the format of model-desired input, prefer to merge this function to model training as it highly depends on model design. But we think there is no need to show in the framework figure. |
| Intel | Disagree.  As commented in the previous question, considering the common information and usage requested by different “ML Training” and “ML Inference” nodes, “Data Collection & Preparation” function can take some common preparation, such as labeling, filtering, etc. |
| ZTE | If data preparation is regarded as the part of AI/ML implementation, so it’s fine for us to not to show data preparation in the framework figure. |
| vivo | Agree.  Enabling the data collection with the ability to delete some unnecessary data can reduce the message load between data collection and model training. |
| CMCC | We would like to remind that this is the first release in RAN to discuss AI related topics which are indeed not expertise of the group. So in our view, we should start with simple framework. In this regard, we may not show data preparation in the figure although we know in implementation we do have some kind of preparation function. The data preparation may not even have any spec impact. |
| AT&T | Data preparation/classification is typically considered part of the ML training process itself. If companies wish to distribute the process across multiple nodes (e.g. on different sides of an interface), there may be benefits, however it could be considered in a similar manner as the “Actor,” and “subject of action” merged blocks and still allow distribution over multiple logical RAN nodes while keeping the overall diagram more simplified. |
| Lenovo, Motorola Mobility | Disagree  What we are drawing is a function diagram, and different functionalities could be located in the same node or not.  Comparing to assuming every Function (e.g. ML training, ML inference) has a sub-function of data collection and preparation, we prefer to draw the “data collection and preparation” as an individual function. We can of course discuss whether they belong to the same node case by case. |
| LGE | Share the view with AT&T |
| Qualcomm | Disagree  Data preparing during training and inference are implementation details. The data preparation in “data collection & preparation” can generate some new kinds of data and may label, filter the data, as a common service. |

### Online/Offline training

Papers address also the need for changing the current Model Inference Host into a function called “ML Training Online/Offline”. There is no clear majority here and a substantial group of companies (see [1], [3], [9], [11], prefer to maintain the name as is, i.e. Model Inference Host (subject to removal of “host” see next section).

In [1] an alternative proposal is made, namely to name the function “Model Training Host” into “Non-co-Located Training Host”, and to name the function “Model Inference Host” into “Model Inference and co-located training Host” (the term “host” could be removed depending on the outcome of the following section). The latter proposal could at least capture co-location and non co-location of inference and training.

**Companies are invited to provide their view on whether the Model Inference Host should be renamed into “ML Training Online/Offline” and alternatively, whether the “Model Training Host” and “Model Inference Host” can be renamed “Non-co-Located Training Host” and “Model Inference and co-located training Host” respectively**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Renaming Model Inference Host into “ML Training Online/Offline” creates confusion due to the meaning of “online training”, which refers to a specific training technique. It is our understanding that RAN3 did not intend to identify specific training techniques in a functional framework.  The second option (“Non-co-Located Training Host” and “Model Inference and co-located training Host”) could at least convey whether inference and training happen in the same node/function, which might have implications in terms of standardization. |
| Deutsche Telekom | As we want to describe the framework from a functional perspective, the term “host” doesn’t make sense, as it is related to a deployment aspect. Therefore, we prefer the use of ML Training and ML Inference function. Deployment aspects addressing if function instances are co-located in same RAN node or non-co-located can be addressed in use case examples.  The addition of “Offline/Online” to ML Training block in the figure can be further discussed. Nevertheless, both online and offline training have to be covered in the framework and via a suitable description in the text. |
| Futurewei | First option: disagree.  Online or offline training is part of learning setting decision, which should be determined at implementation time for a given use case.  Second option: agree if some modification is made.  There are 2 possible settings:   1. local node performs both training and inference: in this case, the entire ML function is a black box, i.e., all information exchange between the model training and model inference is considered as implementation-dependent and is outside 3GPP interest. 2. The other node(s) performs initial model training and local node performs model inference only, or model fine-tuning + model inference: in the latter case, the local node receives the partially trained model from the model training function and makes local modification to the model before performing inference.   It is our suggestion not to make any modification for setting 1) as it is implementation dependent. However, there is impact on the information exchange / interface for setting 2), i.e., some of the interim data may need to be accessible at the local node, thus our proposal is to change current “Model Inference” functional block to “Model inference and/or model fine-tuning”.  We also suggest leaving the “non-co-located” and “co-located” aspect to implementation decision. |
| CATT | We disagree with both.  In our understanding, including two separate boxes does not imply that they reside in different RAN entities.  The “Model inference” module and the “Actor/Action” module may be within one RAN entity, e.g. the gNB-CU-CP, but no one says they should be combined for some case, does it? |
| Nokia | Do not agree. In our view calling Model Training Host as “Model Training (offline/online)” and Model Inference Host as “Model Inference” is better since those names are simpler and aligned with existing ML terminology. ML Training is already captured in the TR as “An online or offline process to train an ML model by learning features and patterns that best present data and get the trained ML model for inference”. Clarifying that Model Training can refer to both online and offline can be done a) by changing the name of the box in the figure to Model Training (offline/online) or b) by keeping the figure name simple as “Model Training” and defining in the TR description that Model Training can refer to both online or offline training. In our view, online training does not refer to a specific technique but rather to training that uses real-time measurements as opposed to offline training where non real-time data is used. |
| NEC | Instead described two options, we would propose to consider the following renaming:  “Model Training Host” to “Model Training”  “Model Inference Host” to “Model Inference” |
| Samsung | Prefer“Model Training Host” to “Model Training”.  This framework is from the functional view. And prefer to not involve deployment in this stage. So it is better keep the block name as “Model Training”.  The detailed training process is out of scope. Online training may have the impact on interface for data collection. For the specified use case, if requiring online training, the feasibility and necessity need to be shown, and the relevant interface impact of online training is studied based on use case.  The definition of “ML training” is already been captured in the TR 37.817[3] as   * ML Training: An online or offline process to train an ML model by learning features and patterns that best present data and get the trained ML model for inference   So offline/online is included in the stage of training. Thus, to make the framework figure brief and clear, “online/offline” is unnecessary to stress repeatedly and “model training” is enough to describe the functionality. |
| Intel | We prefer to keep current naming “ML Training” and “ML Inference”.  1. We support both online and offline training should be in the scope. However, the definition of “ML Training” in the current TR captures both “online” and “offline” process. We wonder it is ok to keep such details in the definition, rather than showing in the box?  2. Whether ML inference and ML training are co-located or non-co-located is use case and implementation specific. |
| ZTE | Disagree with both options.  The framework is from function view, so the modification should be as follows:  -Change “Model training host” to “Model training (offline/online training)”  -Change “Model inference host” to “ Model inference”  According to the definition of the ML training in the current TR37.817:   * *ML Training: An online or offline process to train an ML model by learning features and patterns that best present data and get the trained ML model for inference.*   Based on the definition, ML training is included online/offline training. For the sake of progress, since the definition has involved online and offline, we are also fine to use model training to name the block. |
| vivo | Change “Model Training Host” to “Model Training”  Change “Model Inference Host” to “Model Inference”  The framework based on the function description is clearer than that based on physical entity.  The “online/offline” can be captured in the description. |
| CMCC | We prefer to keep it simple, i.e., current naming “ML Training” and “ML Inference”. If needed, we could add suitable description in the text to cover both online and offline training |
| Lenovo, Motorola Mobility | We don’t see the problem of adding it. Of course RAN3 needs to clarify the exact definition or use better wording. |
| LGE | Disagree the change  Prefer to keep it simple, removing “Host” is fine |
| Qualcomm | Disagree  “Model Training” and “Model Inference” are good enough. The other details could be covered by text description and need not to be in the framework figure. |

### Removing “host” from inference and training functions

A number of papers suggested to remove the word “Host” from “Model Inference Host” and “Model Training Host”. This would maintain the terminology aligned with the functional nature of the framework captured in TR37.817.

**Companies are invited to provide their view on whether the “Model Inference Host” and “Model Training Host” should be changed to “Model Inference” and “Model Training”**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Agree |
| Deutsche Telekom | Agree to remove “host” as this is already deployment related.  Our proposal is use the terms ML Training and ML Inference for the functions. |
| Futurewei | Agree |
| CATT | Agree. |
| Nokia | Agree. But if we consider a framework where the different boxes are “Model Inference”, “Model Training” and “Action” then naturally also Data Sources should be changed to “Data Collection”. |
| NEC | Agree |
| Samsung | OK to remove “host”. |
| Intel | Ok. |
| ZTE | OK |
| vivo | Agree |
| CMCC | Agree |
| AT&T | Agree |
| Lenovo, Motorola Mobility | Agree |
| LGE | Agree |
| Qualcomm | Agree |

### Other proposals related with existing functional framework

#### Model Deployment/Update

In [1] the following agreement taken by RAN3 is recalled:

*- The detailed AI/ML algorithms and models for use cases are out of RAN3 scope.*

It is argued that the agreement above implies that it is not possible to standardize the transfer of an AI/ML algorithm from one node to another because such transfer would imply that the nodes involved in the transfer can decode and interpret the model. Namely, the model would be decodable and interpretable in a standardized way, which implies that the model is not implementation specific anymore.

As a consequence [1] proposes the following:

**Either remove the “Model deployment/update” arrow from the functional framework or mark the Model Deployment/Update as “*limited to single vendor environment*”**

**Companies are invited to comment on the proposal above**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Agree to either remove the arrow or to mark the “Model deployment/update” as “limited to single vendor environment”.  As already discussed in other occasions, the transfer of undefined octet strings would leave the content of an IE unknown and implementation specific, but it would also imply that such IE is not decodable by any nodes that is not from the same vendor as the sender node. Therefore, the signalling becomes proprietary and for that it is out of 3GPP scope. |
| Deutsche Telekom | We don’t agree with both proposals as they are listed here.  The model deployment and update processes are important parts of the whole framework and cannot be omitted. But please note that this part may be described in cooperation with SA5 as it covers a management functionality.  The proposed restriction “limited to single vendor environment” can give a wrong impression. It is right that same model implementation should be used for training and inference, but it may run in the “environment” of different vendors if input/output needs are properly described. The model itself can be proprietary (provided e.g. by RAN vendor, operator, or 3rd party), but the applicability of AI/ML models in a multi-vendor RAN environment is a basic requirement from operators’ perspective.  Proposal for rephrasing of the restriction:  “limited to single model provider”. |
| Futurewei | Agree to add a note to indicate that “typically” this is for single vendor deployment scenario, however, this should not be a requirement.  We do not recommend removing the arrow as the functional framework is to illustrate the logical functions involved in RAN intelligence. If the arrow is removed, then the entire AI/ML function is considered as black box, thus there is no good reason to separate the “Model training” and “Model inference” blocks. |
| CATT | Not needed.  Yes they are algorithm specific, but we believe this arrow helps the reader to understand how the entire system works. |
| Nokia | Disagree. We have same view as DT. |
| NEC | No need to remove arrow. |
| Samsung | Prefer to keep current “Model deployment/update” as it is. When the input/output and functionality are proper, the model can be shared to multi-vendor nodes to reduce the training complexity. |
| Intel | Agree with DT. |
| ZTE | Agree with Futurewei. If we removed the model deployment/update, it break the logical functions in RAN intelligence.  Besides, it can not be precluded that if model training and model inference are deployed in different nodes, the ML model should be deployed from one node to other nodes. |
| vivo | Not needed. |
| CMCC | Not needed to remove, since this is a functional framework, the arrows does not mean a standard interface |
| AT&T | Same view as DT. This is a practical consideration, not a strict requirement. |
| Lenovo, Motorola Mobility | “Model deployment/update” makes total sense to us, which is the fundamental interaction between ML training and ML inference. Should be kept. |
| LGE | Prefer to keep it |
| Qualcomm | Agree with DT |

#### Introducing AI Model type

In [12] a proposal is made to introduce the 3 main categories of ML problems, namely supervised learning, unsupervised learning and reinforcement learning. Other papers also touch upon similar proposals, such as [14] and [15], where also Hybrid Learning, centralized learning, federated learning and distributed learning are proposed for inclusion in the functional framework. The motivation appears to be that of defining different types of inference inputs, inference outputs and training information, depending on the type of ML technique.

However, as discussed above, a framework in which training and inference functions can subscribe to reception of input data and where nodes can subscribe to reception of inference outputs, would not need to distinguish between types of data in dependence of the ML model type. Namely, a training function based on reinforcement learning will request to the Data Source function the reporting of reinforcement inputs (e.g. KPIs), and for that there is no need to define that the training function relies on reinforcement learning.

**Companies are invited to provide their view on whether there is a need to introduce a framework that distinguish between different types of AI algorithms, e.g. supervised learning, unsupervised learning, reinforcement learning, hybrid learning, centralized learning, federated learning and distributed learning**

|  |  |
| --- | --- |
| Company | Comment |
| Ericsson | Disagree.  As per RAN3 agreements, the AI Model is implementation specific and the framework developed by RAN3 should be agnostic to the type of model deployed. |
| Deutsche Telekom | The framework itself should be described in a generalized way that allows the implementation of all the different learning methods.  A differentiation can be made in the description of the use cases where dedicated learning methods may be applied with corresponding interrelations between functions hosted in certain RAN nodes (or other 5GS domains, if needed). |
| Futurewei | The framework should be general across a variety of learning problems vs. only work for certain types of ML problems.  Note that the description in the beginning of this subsection is the opposite of our proposal. We introduced various learning problems in R3-211615 to illustrate that they may have different standards impacts and we should make the description text for the functional blocks and information exchanges general to support a variety of learning problems. It is clearly specified in our proposal that “In order not to limit the RAN intelligence framework only to specific learning problem(s),,, “ certain wording modifications to the current framework are needed.  Note that supervised, unsupervised and reinforcement learning are considered as AI/ML problem types, and they should not be confused with AI/ML “algorithms”. |
| CATT | Slightly prefer to keep as it is. |
| Nokia | We have agreed that the exact algorithms are not in the scope of the SI. However, when it comes to a distinction between the different categories of ML learning (supervised learning, unsupervised learning and reinforcement learning) we think it is important due to the inherent difference between reinforcement learning from other types. Supervised and unsupervised learning allow training and inference to take place in different network entities, as opposed to reinforcement learning where training and inference are more coupled. Therefore, we think it is important to define the different types of learning since they affect the different ML deployment options for the different use cases. |
| NEC | Functional frame work should allow for different ML models and deployment scenarios. |
| Samsung | Due to high relevance of ML model and data set/function, the standard impact should be studied case by case. So, the role of framework is just to provide the reference or the guide for the detailed use case study.  So prefer not to distinguish AI algorithm for framework. |
| Intel | First, we would like to point out that it is not suitable to include centralized/federated/distributed learning as part of AI model type. From model type point of view, we use supervised/unsupervised/reinforcement learning, however centralized/federated/distributed learning are concept for different ML frameworks. This is irrelevant to what model is used, instead, different ML models can be used under the ML framework. Different ML frameworks would impact the coordination between different network nodes, which may be worth to deep dive from standard impact in interface and data collection.  Second, the framework itself can be kept in a generalized way as commented by DT. Our proposal is to capture such definitions of ML models and ML framework in the TR, so that we can consider their impact to existing interfaces and data collection case by case. |
| ZTE | This is corresponding to the pure AI/ML algorithms. No needed to introduce a framework that distinguish between different types of AI algorithms. The impacts of different AI algorithms can be studied case by case in the solution stage, so one general framework is enough. |
| vivo | We are fine with one framework for all different types of AI algorithms. FFS how the framework can fit in with different types of AI algorithms. |
| CMCC | First, the framework itself should be described in a generalized way that allows the implementation of all the different learning methods.  Since we are in studying phase, AI Model type can be put in the TR for information, if companies think they are useful |
| AT&T | Agree with Intel and Nokia, we need to consider the feasibility for different ML frameworks may have different impacts on the required coordination and interfaces. |
| Lenovo, Motorola Mobility | Disagree  Same view as Ericsson and DT. |
| LGE | Agree with DT and CMCC. It should be described in a generalized way. |
| Qualcomm | Disagree.  The framework should be general to cover all these learning types. |

# Conclusion, Recommendations [if needed]

If needed

# References

1. R3-212314, Framework for RAN intelligence (Ericsson)
2. R3-212178, Open issues of framework for AI (Lenovo, Motorola Mobility)
3. R3-212522, Further discussion on the general frame work (Huawei)
4. R3-211632, High-level principles and definitions for the AI/ML-based functional framework for RAN intelligence (Deutsche Telekom AG)
5. R3-212372, (TP for TR 37.817): AI/ML Framework Discussion (Nokia, Nokia Shanghai Bell)
6. R3-212027, High-level framework and definition for AI RAN (ZTE Corporation)
7. R3-212189, Discussion on framework of AI (CATT)
8. R3-212299, Functional Framework of AI/ML enabled NG-RAN Network (Intel)
9. R3-211681, Discussion on open issues in section 4.2 Functional Framework (NEC)
10. R3-212503, Further discussion on high-level framework for AI enabled RAN intelligence (CMCC)
11. R3-211968, Discussion on Functional Framework (Samsung)
12. R3-212373, (TP for TR 37.817): ML-related data support (Nokia, Nokia Shanghai Bell)
13. R3-212636, Response to R3-211632, R3-211681, R3-211682, R3-211754, R3-211968, R3-212027, R3-212178, R3-212189, R3-212299, R3-212300, R3-212314, R3-212372, R3-212503, R3-212522 (NEC)
14. R3-212300, High Level Principles and Definitions of AI/ML enabled NG-RAN (Intel Corporation)
15. R3-211615, Functional Framework for RAN Intelligence to support different learning problems (Futurewei)