|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding  by Artificial Intelligence  www.mpai.community |

|  |  |
| --- | --- |
| **N656** | 22/04/20 |
| **Source** | Requirements (NNW) |
| **Title** | MPAI-NNW Use cases & functional requirements WD 0.2 |
| **Target** | MPAI Community |

# Introduction

During the last decade, Neural Networks are deployed in an increasing variety of domains, but solution and especially deep neural network are costly. The process of AI training is costly not only in terms of resources (GPUs, CPUs, memory) but also time. According to ThinkML, the development of a custom AI solution ranges from $ 6, 000 to $ 300, 000, while renting a pre-built module would cost around $ 40, 000/year. Consequently, it becomes important to guaranty the traceability (owner) and integrity (user) of Neural Networks. Inherited from the multimedia realm, watermarking regroups a family of methodological and applicative tools allowing to **imperceptibly** and **persistently** insert some **metadata** (payload) into an original NN model and to subsequently detect/decode it from the model itself or from any of its inferences.

# Purpose of the standard

The purpose of the MPAI Neural Network Watermarking (NNW) standard is to enable watermarking technology providers to qualify their products. MPAI-NNW will provide the means to measure, for a given size of the watermarking payload, the ability of:

* The watermark inserter to inject a payload without deteriorating the performance of the Neural Network. This item requires for a given application domain:
  + A testing dataset to be used for the watermarked and unwatermarked NN.
  + An evaluation methodology to assess any change of the performance induced by the watermark.
* The watermark detector to recognize the presence of the inserted watermark when applied to a watermarked network that has been modified (*e.g.*, by transfer learning or pruning) or to any of the inferences of the modified model. This item requires for a given application domain:
  + A list of potential modification types expected to be applied on the watermarked NN as well as of their ranges (*e.g.*, random pruning at 25%).
  + Performance criteria for the watermark detector (*e.g.*, relative numbers of missed detection and false alarm).
* The watermark decoder to successfully retrieve the payload when applied to a watermarked network that has been modified (*e.g.*, by transfer learning or pruning) or to any of the inferences of the modified model. This item requires for a given application domain:
  + A list of potential modification types expected to be applied on the watermarked NN as well as of their ranges (*e.g.*, random pruning at 25%).
  + Performance criteria for the watermark decoder (*e.g.*, 100% or (100-α)% recovery).
* The watermark inserter to inject a payload at a low computational cost, *e.g.*, execution time on a given processing environment.
* The watermark detector/decoder to detect/decode a payload from a watermarked model or from any of its inferences, at a low computational cost, *e.g.*, execution time on a given processing environment.

# Users of watermarking technology for NN

Four types of actors are identified as playing a role in the use cases.

* *NN owner* – the developer of the NN, wishing to ensure that ownership of NN can be claimed.
* *NN watermarking provider* – the developer of the watermarking technology able to carry a payload in a neural network or in an inference.
* *NN customer* – the user who needs the NN owner’s NN to make a product or offer a service.
* *NN end-user* – the user who buys an NN-based product or subscribes to an NN-based service.

# Use cases

The use cases are structured into two categories: the first relates to the NN *per se* (*i.e.*, to the data representation of the model, as discussed in Section 3.1) while the second to the inference (*i.e.*, to the result produced by the network when fed with some input data, as discussed in Section 3.2).

The use cases are presented as sequence diagrams describing the positions and actions of the four main actors in the workflow.

## Use cases related to watermarking the Neural Network model

Two types of use case belong to this category:

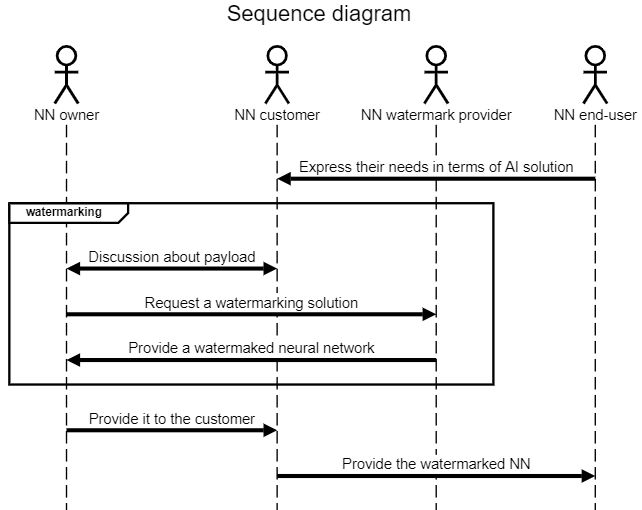
* *payload*, *i.e.*, data carried by the watermark is used to identify the actors or the model; this case is presented in Section 3.1.1.
* *loss of integrity*, *i.e.*, data carried by the watermark is used to identify modifications in the model; this case is presented in Section 3.1.2.

### Payload (channel of information)

Data is carried by the watermark is used to identify:

* the ownership of an NN.
* an NN (as if it were a DOI).

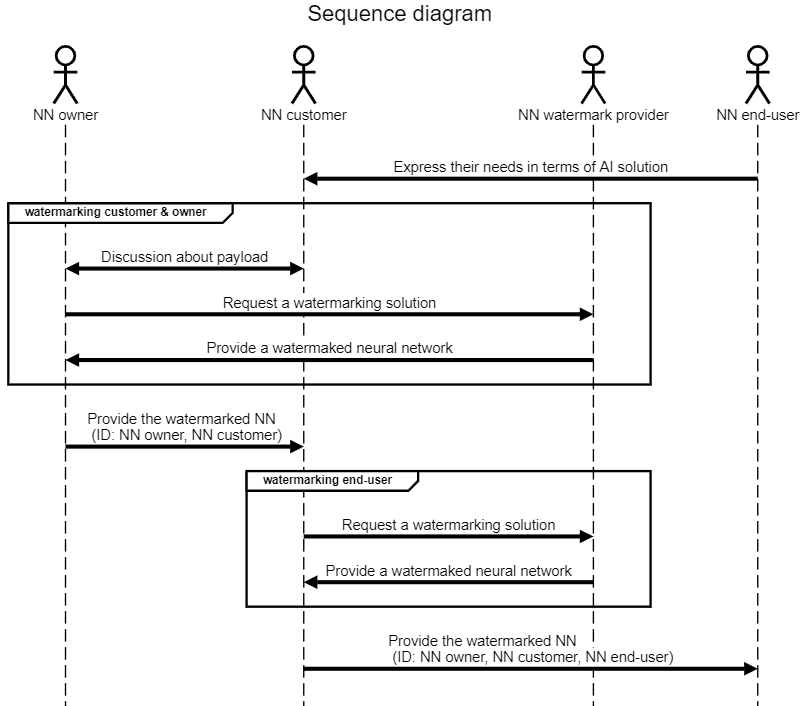
#### Identify the ownership of an NN



*Figure 1:* *Identify the ownership of an NN use case: NN owner and NN customer identifiers are inserted*

*Description of Figure 1 workflow:*

* *NN customer* gets needs from product/service from *NN end-users*.
* *NN customer* requests NN model from *NN owner* in order to be able to create the product/service requested by the end user.
* *NN customer* and *NN owner* share the need to protect NN intellectual property; *NN customer* does not want others to use the model to make similar products or offer similar services; *NN owner* wants to acquire others customers as *NN customer*; ideally, the *NN end-user* ID should also be added to the watermark (*cf*. the workflow in Figure 2).
* *NN end-user* acquires the product and/or access to the service with the embedded NN watermark.

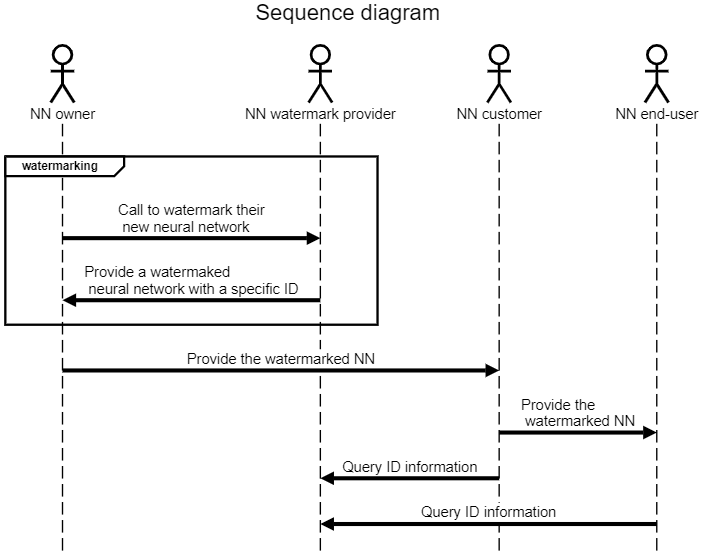


*Figure 2:* *Identify the ownership of an NN use case*: in addition to *NN owner* and *NN customer* identifiers, the *NN end-user* identifier is also inserted

*Description of Figure 2 workflow:*

* *NN customer* gets needs from product/service from end-users.
* *NN customer* requests NN model from *NN owner* in order to be able to create the product/service requested by the *NN end-user*.
* *NN customer* and *NN owner* share the need to protect NN intellectual property; *NN customer* does not want other to use the model to make similar products or offer similar services ; *NN owner* wants to acquire other customers as *NN customer*.
* *NN customer* needs to make sure that *NN end-users* do not share the AI solution, thus they insert an identifier for each *NN end-user*.
* *NN end-user* acquires the product and/or access to the service with the embedded NN watermark.

#### Identify an NN (e.g. DOI)

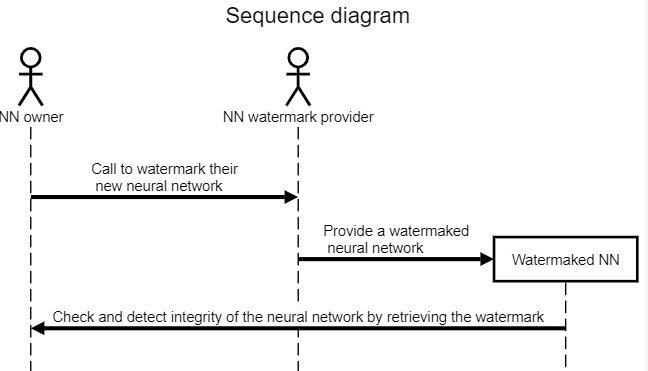


*Figure 3:* *Identify an NN use case*: *NN receives an ID (e.g. DOI)*

*Description of Figure 3 workflow:*

* *NN owner* wants its NN to receive a specific identifier.
* *NN watermark provider* gives a solution with a specific identifier for any new Neural Network and manages the ID usage through its lifecycle (e.g., validation to third parties, or ID record deletion when no longer used).

### Loss of integrity



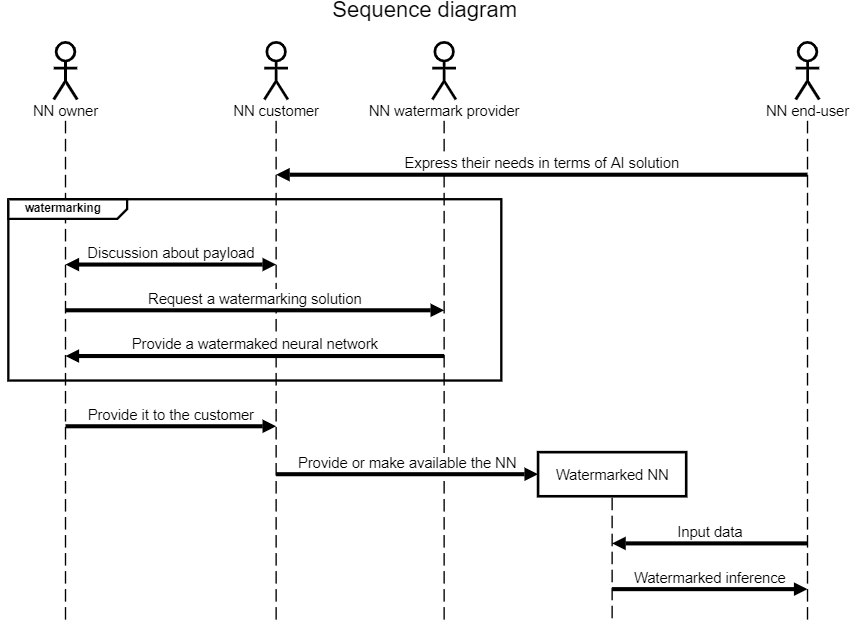
*Figure 4: Check the NN integrity use case*

*Description of Figure 4 workflow:*

* *NN owner* wants a watermark that permits them to check the integrity of the NN.
* *NN watermark provider* inserts an integrity validation watermark in the NN.
* *NN owner* can distribute the Watermaked NN to their customers.
* *NN owner* can check the integrity and detect modifications of their NN.

## Inference

The 4 use cases described in Section 4.1 are not restricted to watermarking the NN model and can be also applied to the watermarking of the NN inference. For instance, Figure 5 reflects the *Identify the ownership of an NN* use case for NN inference.



*Figure 5: Watermarked NN inference use case*

*Description of Figure 5 workflow:*

* *NN customer* gets needs from product/service from end-users.
* *NN customer* requests NN model from *NN owner* in order to be able to create the product/service requested by the end user.
* *NN customer* and *NN owner* share the need to protect NN intellectual property; *NN customer* does not want others to use the model to make similar products or offer similar services; *NN owner* wants to acquire other customers as *NN customer*.
* *NN end-user* can feed the Watermarked NN with input data and receive the inference which is watermarked. The contained ID related information can be the same as in Section 4.1 Use cases related to watermarking the Neural Network model, for instance.

# Summary of the use-cases

The use cases identified so far and presented in this document can be structured according to Figure 6.

Diagram

Description automatically generated

*Figure 6: Retrospective view on the use cases*

# Requirements

## Impact of the watermark on the performance

If the NN under test is an implementation of an NN standardized by an MPAI standard, then use the MPAI Conformance Testing process specified for that NN. If this is not case, the following process should be used:

1. If the NN has the input and output data format with specified semantics, use the following process:
   1. Define a test dataset of sufficient size.
   2. Feed the unwatermarked NN with the test dataset and measure the task-dependent quality of the produced inference.
   3. Feed the watermarked NN with the same test dataset and measure the task-dependent quality of the produced inference.
   4. Provide the quality statistical parameters measured in steps 1.b and 1.c.
2. If the input and output data format of the NN do not have specified semantics:
   1. Connect the NN to other NN until the input and output of the resulting configuration have input / output formats with specified semantics.
   2. Apply all steps in Case 1.

**To respondents:**

MPAI requests respondents to comment on the text described above and to propose general guidelines for testing NN for specific application domains.

Editor note: the text hereafter requires changes

## Detection capability

The ability of the watermark detector to recognize the presence of the inserted watermark when applied to a watermarked network that has been modified (e.g., by transfer learning or pruning). This item requires for a given application domain:

* + A list of potential modification types expected to be applied on the watermarked NN as well as of their ranges (*e.g.*, random pruning at 25%).
  + Performance criteria for the watermark detector (*e.g.*, probabilities of missed detection and false alarm).

## Decoding capability

The ability of the watermark decoder to successfully retrieve the payload when applied to a watermarked network that has been modified (e.g., by transfer learning or pruning). This item requires for a given application domain:

* + A list of potential modification types expected to be applied on the watermarked NN as well as of their ranges (*e.g.*, random pruning at 25%).
  + Performance criteria for the watermark decoder (*e.g.*, 100% or (100-α)% recovery).

## Processing cost

The processing cost of watermark injection, e.g., execution time on a given processing environment, computational complexity. This item requires to specify a set of processing configuration. (is it meaningful to specify a processing configuration? Will the resulting number an interest ?)

# Annex 1 – MPAI-NNW Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
| Attacks | Any transformation, malicious or not, applied after the mark injection. Attacks can be of various types:  *Removal attacks*: turn undetectable/unreadable the information conveyed by the watermark.  *Geometric attacks*: destroy the watermark synchronization, rather than to remove it.  *Cryptographic attacks*: detect and remove the mark without knowledge of the key, thanks to the knowledge of the embedded mark.  *Protocol attacks*: embed another watermark and/or create an ambiguous situation upon the mark detection/decoding. |
| Black-Box | Do not grant access to the network only to its inference. |
| Data payload | The amount of information injected through watermarking process. |
| Fine-tuning | Apply an additional training to the network on the same task. |
| Imperceptibility | Inference quality of the model on its original task should not be degraded significantly through the watermark injection. |
| Knowledge distillation | A surrogate model is trained by using the inference of a targeted network as label. |
| Ownership metadata | The data carried by the watermark representing the owner and the usage conditions. |
| Pruning | Remove a portion of a network's parameters while keeping its performances. |
| Quantization | Reducing the number of bits required for describing the model parameters. |
| Robustness | The ability of the watermark to withstand a prescribed class of attacks |
| Transfer-learning | Apply an additional training to the network on another (similar or not) task. |
| Watermark decoder | An algorithm able to decode an inserted watermark, when applied to a watermarked network. |
| Watermark detector | An algorithm able to detect an inserted watermark, when applied to a watermarked network. |
| Watermark forging | A particular case of *Protocol attacks*, where the attacker tries to forge the watermark to create an ambiguous situation where both parties can claim the ownership. |
| Watermark overwriting | A particular case of *Protocol attacks*, where the attacker embeds another watermark to create an ambiguous situation where both parties can claim the ownership. |
| White-Box | Grant access to the network and make it possible for the watermark to be embedded inside the parameters. |