DNN Watermarking: Four Challenges and a Funeral

Mauro Barni
barni@diism.unisi.it
University of Siena
Siena, Italy

Fernando Pérez-González
fperez@gts.uvigo.es
Atlanttic Research Center,
University of Vigo
Vigo, Spain

Benedetta Tondi
benedetta.tondi@unisi.it
University of Siena
Siena, Italy

ABSTRACT

The demand for methods to protect the Intellectual Property Rights (IPR) associated to Deep Neural Networks (DNNs) is rising. Watermarking has been recently proposed as a way to protect the IPR of DNNs and track their usages. Although a number of techniques for media watermarking have been proposed and developed over the past decades, their direct translation to DNN watermarking faces the problem of the embedding being carried out on functionals instead of signals. This originates differences not only in the way performance, robustness and unobtrusiveness are measured, but also on the embedding domain, since there is the possibility of hiding information in the model behavior. In this paper, we discuss these dissimilarities that lead to a DNN-specific taxonomy of watermarking techniques. Then, we present four challenges specific to DNN watermarking that, for their practical importance and theoretical interest, should occupy the agenda of researchers in the next years. Finally, we discuss some bad practices that negatively affected research in media watermarking and that should not be repeated in the case of DNNs.

CCS CONCEPTS

• **Security and privacy** → Information-theoretic techniques.

KEYWORDS

DNN watermarking, deep learning, DNN IPR protection, robust DNN watermarking, side informed DNN watermarking

ACM Reference Format:
Mauro Barni, Fernando Pérez-González, and Benedetta Tondi. 2021. DNN Watermarking: Four Challenges and a Funeral. In *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '21), June 22–25, 2021, Virtual Event, Belgium.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3437880.3460399

1 INTRODUCTION

Thanks to the outstanding performance they achieve, Deep Neural Networks (DNN) are increasingly deployed and commercialised in virtually all applications dealing with data and signals for which precise statistical models do not exist. Training a DNN model, however, is a difficult and computational intensive piece of work, requiring an extensive training procedure that may easily go on for weeks, even on powerful workstations equipped with several GPUs. A good deal of domain-specific know-how accompanied by a deep knowledge of the mechanisms underlying the training process is also necessary. For this reason, the demand for methods to protect the Intellectually Property Rights (IPR) associated to DNN is rising. Borrowing from similar efforts pertaining to media protection [22], watermarking has recently been proposed as a way to protect DNN IPRs and track the legitimate or illegitimate usage of DNN models.

Despite the large number of watermarking techniques developed in the past decades, addressing different media in a wide variety of application scenarios characterised by different requirements, the watermarking process always exploits some forms of redundancy present in the host document, thanks to which the document can be modified without impairing its informative or perceptual meaning. The same idea holds for DNN watermarking. The very large number of parameters (the network weights) that define DNN models, confers to the network a capability of processing the input data that often exceeds the difficulty of the task the network is trained for, hence leaving enough degrees of freedom in the choice of the model weights. The weights, then, can be modified or directly generated, in such a way to host the watermark.

In addition to the above basic principle, DNN and media watermarking share other common features. To start with, the requirements that any watermarking scheme must satisfy still follow the so-called watermarking trade-off triangle (see Figure 1), depicting the necessity of finding a good balance among three opposite requirements, namely: capacity, robustness (sometimes security) and unobtrusiveness. In DNN watermarking, unobtrusiveness refers to the capability of the watermarked network to accomplish the task it is thought for. Robustness is related to the possibility of correctly extracting the watermark from a modified version of the model (e.g. after fine tuning, or model pruning), while capacity (more correctly indicated as payload) measures the number of information bits conveyed by the watermark. Another concept applying to both DNN and Media watermarking regards the distinction between zero-bit and multi-bit watermarking. The former refers to a situation wherein watermark extraction (usually indicated as watermark detection) corresponds to deciding whether a given model contains a certain watermark or not, while in the multibit case, the watermark bits are extracted from the host model without knowing them in advance. Other characteristics applying to media watermarking as well as to DNN watermarking, include the distinction between robust and fragile watermarking, and the possible use of a key (usually referred to as the watermarking key) to prevent watermark extraction (as well as embedding and removal) by non-authorized users.

IH&MMSec '21, June 22–25, 2021, Virtual Event, Belgium

ACM ISBN 978-1-4503-8295-3/21/06…$15.00
https://doi.org/10.1145/3437880.3460399

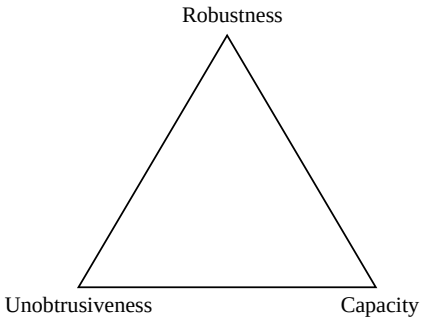

Figure 1: The watermarking trade-off triangle.

Notwithstanding the above similarities, embedding a watermark into a DNN and retrieving it from a marked model is quite a different piece of work with respect to media watermarking, as will be discussed in the following.

In the above framework, the goal of this paper is threefold:

- to discuss some characteristics of DNN watermarking that clearly distinguish it from media watermarking, leading to a DNN-specific taxonomy of DNN watermarking techniques (Sect. 2) ;
- to present 4 challenges specific to DNN watermarking that, for their practical importance and theoretical interest, should occupy the agenda of researchers for the next years (Sect. 3);
- to highlight some errors that negatively affected research in media watermarking, and that should not be repeated in the DNN case (Sect. 4).

We are confident that this paper will contribute to further raise the interest in DNN watermarking, helping researchers to focus on the most interesting open issues and opportunities associated to this field, and treasure on the wealth of theoretical and practical insights stemming from traditional media watermarking.

2 DNN-SPECIFIC WATERMARKING ISSUES

As we mentioned in the introduction, although the main features of DNNs and media watermarking are the same, there are some significant dissimilarities between media and DNN watermarking, stemming from the different nature of the application scenario. Before discussing them in Sect. 2.2, we briefly introduce some basic formalism of DNNs and watermarking models.

2.1 DNN models and watermarking

The strength of DNNs is their ability to (automatically) learn complex features and characteristic patterns directly from the input data. Let $x \in \mathbb{R}^n$ denote the input. The function learned by the DNN can be written as $\phi(x;\theta^L) \in \mathbb{R}^m$ where $\theta^L = [\theta_1, \theta_2, \cdots, \theta_L]$ is the vector of the network parameters, that is, the weights and biases associated to the neurons. The output dimension m depends on the task: in DNN classification, m is the number of classes; in DNN estimation or generative models, m denotes the output domain dimension (typically, $m = n$). The parameters are learned from training data and optimized in such a way to minimize some loss function between the prediction and the ground truth. The number of parameters L the network consists of is typically huge (this is especially the case with modern architectures). The idea behind DNN watermarking is then to exploit the redundancy of these parameters to embed additional information, without degrading the performance for the to-be-accomplished task. In the following, we denote the watermarked DNN function with $\phi_w(x) = \phi(x;\theta_w^L)$, where θ_w^L is the set of the parameters of the watermarked DNN model.

2.2 Dissimilarities between Media and DNN watermarking

A noticeable difference between DNN watermarking and media watermarking regards the *embedding procedure*. While in classical media watermarking the watermark is embedded into the document, called 'host' document, by minimizing the amount of introduced distortion, in the case of DNNs, the effect of a modification of the parameters (typically the weights) on the performance of the network is not easy to identify. Therefore, the injection of the watermark can not be performed through the direct modification of the weights of the model, but it has to be carried out during the learning phase, contextually to the learning procedure for the primary task, by properly designing the loss function. In this way, the desired behavior for the task and the watermark are learned simultaneously[1]. Therefore, the traditional concept of original non-watermarked asset does not apply to DNN watermarking. The solution found when the network is watermarked can be very different from the one obtained in the non-watermarked case, that is, the local minimum of the loss function θ_w^L which the network converges to can be very far away from θ^L.

Another important dissimilarity pertains to the concept of watermark *domain*. With classical image watermarking algorithms, embedding can be carried out either in the original domain or in a convenient transformed domain, e.g. the frequency domain, In contrast, in DNN watermarking, the embedder chooses the to-be watermarked layer(s) of the network: the watermark can be embedded directly into the weights by modifying the parameters θ_i of one or more layers (static watermarking), or associated to behavior of the network in correspondence to some specific inputs (dynamic watermarking). More precisely, *static watermarking* methods typically embed the watermark into the weights of the model [4, 25]; then, the watermark is recovered, as in classical watermarking, reading the weights and exploiting the knowledge of a secret key K, known to both the embedded and the decoder. The extraction function is then a function of θ_w^L and K. In *dynamic watermarking* instead, the secret key is the input data x_K, and the watermark is extracted by looking at the behavior of the network, e.g. at the status of the activation maps or at the final output. Notably, when the watermark is read from the network output, the watermark can be extracted in a black-box setting [29], without requiring access to the internal layers of model. In this case, the extraction function is a function of $\phi_w(x_K)$. Dynamic DNN watermarking has also immediate connections with DNN backdooring and backdoor attacks

[1]We stress that, even when the embedding is performed by fine-tuning the model already trained, the loss function should be designed in such a way to also take into account the primary task of the network.

[12]. Actually, when the backdoor is injected by the network designer himself, DNN backdooring can be regarded to as a particular kind of dynamic watermarking. In this case, the triggering event, that has the same role of the watermark key, is often a signal that is superimposed to the image. Obviously, the distinction between static and dynamic watermarking is a peculiarity of DNN watermarking and does not apply to media watermarking, where only static approaches are viable.

Another possible characterization of DNN watermarking schemes pertains to the way watermark extraction is carried out. In *white-box* schemes, the watermark is read from the internal parameters/status of the network (be them the weights, as in [19, 25] or the activation maps as in [23]), thus the decoder is assumed to have access to the model. In contrast, *black-box* watermarking schemes assume that the decoder can only access the final output of the network (e.g. a remote service API), and the watermark is recovered by querying the model with some specific inputs and looking at the output. Obviously, in this case, the maximum number of bit that can be extracted with a query depends on the dimensionality of the network output.

3 FOUR CHALLENGES FOR FUTURE RESEARCH

Due to the differences with traditional media watermarking, DNN watermarking techniques have to be developed facing the new challenges posed by the DL application scenario. We identified four main challenges that are discussed below.

3.1 Robust watermarking

The robustness requirement for the watermark regards the possibility of recovering the watermark from a perturbed version of the model. The model is perturbed for instance when it is fine-tuned on a different training set of images. In this case, in fact, even if the classification task remains the same and the input data are of the same type, the retraining process alters the parameters of the network and this may affect the watermark. More robustness against fine-tuning is expected when the watermark is embedded from scratch, together with the training on the main task of the model. Some works have shown that the watermark can be easily retained after fine-tuning if the network is trained for few epochs or iterations [16], but when the number of epochs increases, the watermark tends to disappear. Yet stronger perturbations occur when the model is used in a transfer-learning scenario, that is, when the model parameters are just the initial point for training on a different classification task, exploiting the fact that the knowledge acquired for one task can help to solve related ones.

Given the huge amount of resources required for training deep architectures, pre-trained models (usually made available by big providers) are often exploited to initialize other training processes, since this speeds up the learning process and permits to obtain better solutions (with respect to training from scratch). In many transfer-learning scenarios, a well behaving model is obtained by retraining only the fully-convolutional layers, since the features learned for the original problem in the convolutional part can also work for the task at hand. Obviously, in this case, a watermark embedded into the weights of the convolutional layers of the pre-trained model is not affected and then will also be present in the new model. However, when this is not the case (e.g. in an adversarial setting), transfer learning represents a serious threat, and researchers should strive to design watermarking algorithms that are robust to transfer-learning to the largest possible extent. A possibility in this direction, is to play with the sensitivity of the loss function to weight variations. Being the network trained to simultaneously learn the classification task and the watermark, part of the weights (those wherein the watermark is embedded) could be made more sensitive to loss variations, by properly defining the loss function. Therefore, when retraining is performed starting from the watermarked model, changes to the weights bearing the watermark will be penalized. Such an approach has recently been considered in [24] to design a watermarked model with improved robustness against fine-tuning on the same training set. However, it is not obvious how to extend the approach to make it effective in a more general fine-tuning scenario and in the more challenging case of transfer-learning.

The watermark should also be robust against model pruning [17], often performed in order to reduce the size of the trained models. Pruning can be performed randomly, or, more often, based on the parameters' contribution to the loss, by removing the least important parameters according to some criterion [18]. A watermarking approach that takes into account parameters pruning while performing watermark embedding during training, could help in this direction, e.g., by ensuring that the watermarked/embedded parameters are also relevant for the main task.

As for media watermarking, robustness can be improved by spreading the watermark over more parameters, i.e. reducing the payload (see again the trade-off triangle). However, the development of algorithms that can effectively control the trade-off between payload and robustness, in order to achieve improved robustness at the expense of a lower payload, is still a major challenge of DNN watermarking.

Another interesting aspect pertains to the robustness against query-based black-box attacks (often called surrogate model attacks). Such attacks aim at building a substitute model that mimics the original network by querying it and accessing its output labels. The local model is trained using adversarial data augmentation [21]: the target is first queried with test data and the classifier is trained based on the target label; then, to better approximate the boundary, white box adversaries targeting the substitute model are performed and evaluated on the target at every iteration. We might expect that, if the substitute model is a good approximation of the target one, and the classification boundary is learned with good approximation, the watermark can also be transferred to the substitute model. In this scenario, the transfer capability of the watermark could be linked to the number of queries made to build the surrogate model.

3.2 A theory for dynamic watermarking

As we already observed in Sect. 2.2, dynamic watermarking is a brand new opportunity offered by DNN watermarking that was not available in the multimedia case. With dynamic methods, the DNN is regarded as a functional rather than a static object and the

watermark is associated to the behaviour of the functional in correspondence to a set of properly selected inputs (watermark triggers), or even in correspondence to all the inputs. This perspective offers a wide range of new opportunities that are not available in the static case, all the more that, at least in the white-box case, the behaviour of the functional can also be defined at the intermediate levels of the network. The degrees of freedom the watermark designer has are virtually endless: i) choice of the level wherein the behaviour of the network is observed, ii) choice of the observed behaviour, which can range from a classical spread spectrum approach wherein the correlation of an activation map with a secret sequence is considered, to more sophisticated schemes observing the activation of specific neurons or the appearance of otherwise unexplainable errors, iii) choice of the input data used to reveal the presence of the watermark, iv) adoption of a specific training procedure to induce a specific behaviour, and so on. At first sight, the number of possibilities is so large that one could imagine the possibility of embedding an extremely high payload, or an extremely robust watermark, by simply increasing the number of triggers. Nevertheless, one should always remember that even if the watermark is *read/detected* by looking at the behaviour of the functional implemented by the DNN, its presence is ultimately dictated by the weights and biases of the network, and hence watermarking ultimately relies on the exceeding capabilities of the network with respect to the task it is thought to accomplish. In this sense, the presence of the watermark naturally conflicts with the capability of the network to handle difficult tasks while at the same time exhibiting good generalization capabilities. To clarify this point, let us consider the admittedly imperfect analogy with polynomial interpolation. Suppose with want to watermark a polynomial $p(x)$ of degree $n-1$, by ensuring that the polynomial assumes desired values $\{y_1 \dots y_k\}$ in $k < n$ points $\{x_1 \dots x_k\}$. Given that $n > k$, there are ∞^{n-k} polynomials for which $p(x_i) = y_i$. To define the watermarked polynomial $p_w(x)$, we may choose $n-k$ additional points $\{x_1^* \dots x_{n-k}^*\}$ and values $\{y_1^* \dots y_{n-k}^*\}$ and further impose that $p_w(x_j^*) = y_j^*$. The watermark will be associated to the behaviour of $p_w(x)$ on the particular inputs $\{x_1^* \dots x_{n-k}^*\}$, but it will ultimately depend on the coefficients of the polynomial, hence making it impossible to specify the behaviour of the polynomial on more that $n-k$ inputs. The robustness requirement can also be understood through the polynomial interpolation analogy. Let us assume that we are given a watermarked polynomial $p_w(x)$ and suppose that we want to slightly modify the values it assumes on the points $\{x_1 \dots x_k\}$, thus obtaining a new polynomial $p'_w(x)$. To what extent will the new polynomial satisfy the watermark conditions $p_w(x_j^*) = y_j^*$? If the new polynomial is computed from scratch, the watermark of course will disappear, but if the new polynomial is computed by slightly modifying $p_w(x)$, it is possible, in principle, that the values assumed by the polynomial on the triggering inputs $\{x_1^* \dots x_{n-k}^*\}$ do not change much. Figure 2 provides an illustration for the case $n = 5$, with $k = 3$. The values taken by $p'_w(x)$ in correspondence of the triggering inputs $\{x_1^*, x_2^*, x_3^*\}$ are similar to those taken by $p_w(x)$, while those taken by $p(x)$ (a general polynomial passing through the perturbed points, representing the training from scratch case) are very different, especially in x_1^* and x_3^*.

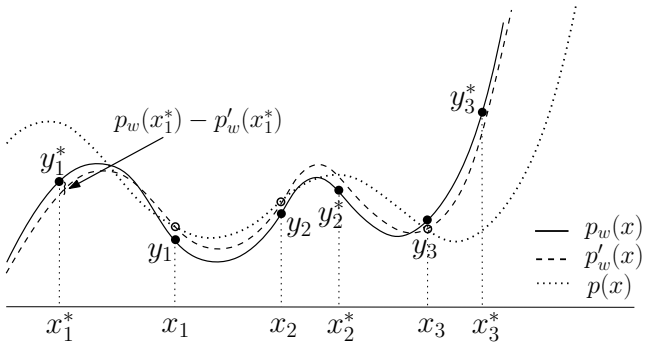

Figure 2: The polynomial interpolation analogy. $p_w(x)$ is the watermarked polynomial passing through $\{y_1, y_2, y_3\}$ and $\{y_1^*, y_2^*, y_3^*\}$ (solid circles), $p'_w(x)$ is the polynomial obtained by slightly modifying the values taken by $p'_w(x)$ in $\{x_1, x_2, x_3\}$ (empty circles), while $p(x)$ is a general polynomial passing through these three new points.

Going back to DNNs, it is clear that the possibility of dynamically watermarking a network depends on the exceeding dimensionality of the parameter space θ^L, however several questions need to be answered to clarify the potentialities of dynamic (vs static) watermarking: i) how many triggering inputs can we define without affecting the capability of the network to solve the problem it is designed for? ii) Assuming the expressive capability of the network is large enough, how should we design the training procedure to make sure that we find a suitable solution to the watermarking problem? iii) What is the impact of fine-tuning, retraining, pruning, on the behaviour of the network in correspondence to the watermark triggering inputs? iv) Is it preferable that the triggering inputs are chosen in the *vicinity* of the standard inputs or should they be alien to the task the network is asked to solve?

Given that the theory developed for media watermarking cannot provide a satisfactory answer to the above questions, the necessity of developing a sound and rigorous theory of dynamic watermarking is a pressing one, if we want that DNN watermarking establishes as a solid and well founded field.

3.3 DNN watermarking capacity

An important property of any multi-bit scheme algorithm is its payload, that is, the number of bits the watermark message consists of. Such a property is often, and misleadingly, associated to the concept of watermarking capacity. Given a multimedia object, or a DNN model, or even better, a class of models, the watermarking capacity, indicates the maximum payload that can be achieved by any watermarking algorithm assuming a certain level of robustness and unobtrusiveness (here we are not interested in giving a precise definition of robustness and unobtrusiveness, such an exact definition being part of the challenge we are discussing in this section). This problem has been deeply investigated in the case of media watermarking [1]. Despite its theoretical nature, the solution of the capacity problem (though limited to highly ideal scenarios) has literarily revolutionised the watermarking field, opening the way to the development of new classes of watermarking algorithms, greatly

outperforming the early solutions proposed previously [3, 14]. A similar question applies to DNN watermarking. How many bits can be reliably hidden within a DNN model consisting of a certain number of parameters and thought to solve a given task? Is there a difference, on this respect, between static and dynamic schemes? Some experiments regarding static watermarking [16, 25] show that watermark embedding acts as a regularisation term on the loss function used for training and may even result in a better generalisation capability of the network. We expect, though, that such an effect will be a limited one and that increasing the payload beyond a certain limit will impact negatively the network accuracy.

In media watermarking, achieving the watermarking capacity passes through quantization index modulation (QIM [3]), whereby the watermark is embedded by quantizing the coefficients hosting the watermark with one between two (or more) evenly spaced quantizers according to the watermarking bits. A scheme following such an approach has been proposed for DNN watermarking in [16], however it is not clear if QIM watermarking may also be used in DNN watermarking to increase the watermark payload and up to which extent. A closely related question regards the role of channel coding. While it is pretty obvious that channel coding may help to increase the robustness of DNN watermarking, the way channel coding should be incorporated within the embedding process during the training phase is not clear. Also unclear, it is the kind of channel codes that fits better the DNN scenario.

Following our experience with media watermarking, we deem that looking for theoretically sound answers to the above questions, and finding good theoretical models to eventually estimate the watermarking capacity of DNNs, will help developing practical solutions indicating the general guidelines that the design of a high capacity watermarking system should follow.

3.4 Joint DNN and media watermarking

Black-box models and white-box models are not necessarily mutually exclusive: for instance, the model can be treated as a black-box while querying it with a certain input (acting as a key) that is expected to trigger a certain response observable by the verifier; this would be sufficient to initiate further actions (e.g., a court order) to get white-box access to the weights in order to read a fingerprint embedded in them and trace the source that infringed the copyright.

Inputs that elicit a telltale output when querying the model might be detectable inside the black-box and then routed to a lower-performance non-copyrighted model also contained in the box. One appealing alternative would be to embed a watermark in *every* output of the DNN so that it is possible to carry out the copyright verification without having direct access to the black-box thereby raising less suspicion. Naturally, for this to happen, the output needs to reach a minimum entropy so that a distortion/performance constraint is met while being altered by the presence of the watermark. However, DNNs meeting this requirement are becoming more and more common, e.g. in multiclass ranking or networks that produce images. The latter, which offer ample possibilities for watermarking, are becoming more popular with the success of convolutional neural networks in applications like deblurring [20], denoising [30], superresolution [5], demosaicing [15], compression [13] or inpainting [27], to name a few.

We will focus here on the case of output images. Of course, one trivial way of watermarking the output would be to place a classical embedder after the DNN and totally uncoupled from it. The advantage of such an approach would be that all the vast existing knowledge about image watermarking would straightforwardly apply here. But this would not exploit the flexibility of the model in accomplishing additional tasks to its original purpose. Moreover, the uncoupling of the watermarking subsystem would make it prone to reverse engineering attacks aimed at removing or impairing it. It is clear then, that it would be desirable to require the DNN to produce an output containing a watermark. This watermark should not only be robust to conventional attacks such as compression, geometric distortions, cropping, filtering, etc, but also to attacks to the DNN itself, such as surrogate models (see Sect. 3.1). For the reasons given in the first paragraph of this section, it would also be advisable that the weights contain a directly decodable watermark (usually, a fingerprint) in case the black-box is to be opened-up. Then, with this approach there would be two types of watermark detectors/decoders: one directly applicable to the weights (white-box mode), another to the outputs (black-box mode). In a sense, models that watermark the output could be considered as *box-free*, as the detection/decoding could be done without accessing the box nor choosing the inputs. However, any proof of ownership will ultimately require querying the model (with no particular input) and examining the output in a verifiable setting.

Examples of networks for image processing that embed the watermark either in the weights [7] or in the outputs [28], [26] already exist, but to the best of our knowledge a solution to the joint problem is not available.

Also missing is a theory for watermarking images using DNNs. In principle, there are two possible approaches regarding the detector/decoder: the simplest one is to use a conventional decoder (e.g., spread-spectrum of side-informed) and train the network so that the output of the decoder is the desired one (e.g, through the cross-entropy of the embedded information), including a range of attacks in the loop. Another approach is to jointly train a DNN that performs the detection/decoding. Unfortunately, no theoretical support exists in either approach. For instance, a very interesting challenge would be to train the network for those cases in which theoretical limits and practical code constructions are available (e.g., Costa's writing on dirty paper scenario [8], where the host image and the channel are additive white Gaussian) and measure how close the codes produced by the DNN are to the optimal. In fact, one would expect the model to learn the principle of side-informed watermarking. This would in turn allow us to establish a beachhead to explore the uncharted field of watermarking with DNNs.

As a final note, we mention that embedding a watermark in every output of the network would also make it possible to verify the integrity of the images. For instance, it would be possible to check whether an image denoised with a proprietary deep network has not been tampered with at a later stage. For this to happen, a fragile or semi-fragile watermark must be embedded, which may be compatible with other watermarks targeted at copyright protection.

4 A FUNERAL

Or should we rather say funerals? For many years, both as reviewers and associate editors, we have observed a series of bad practices that, now that watermarking resurfaces on the occasion of the boom of DNNs, we should bury and organize their funeral. Interestingly, at the 2011 IEEE WIFS Ton Kalker, one of the pioneers of watermarking, gave a keynote titled "Watermarking: Quo Vadis?" which is an excellent starting point for deciding what to put in the coffin. We first discuss Kalker's contributions and adapt them to the case of DNNs with illustrative examples.

Confusing watermarking security with cryptographic security. In data-hiding, security should not be measured as the difficulty in reading the hidden information, because it is evident that if it is encrypted with a good cryptographic algorithm, it will not be possible to access the content without the cryptographic key. It is necessary to understand that most of the existing watermarking schemes also use a key for embedding; this helps giving them true security and makes them more host and (in the multi-bit case) message agnostic. But contrary to cryptography, in watermarking the threat model involves being able to erase or overwrite the watermark, especially if we are talking about IPR protection. In zero-bit watermarking the goal is to erase or alter the watermark in such a way that the detector returns a negative result. In multi-bit watermarking it is about eliminating or distorting the hidden message.

In multimedia watermarking researchers managed in the second half of the 2000s to propose different security metrics in various scenarios related to the amount of information that system outputs (in this case, the watermarked media) contain about the secret parameters derived from the embedding key (for example, the sequence used in spread-spectrum or the dithered lattice of side-informed methods) [2], [6]. In other words, how much can we learn from the secret parameters by looking at n pairs of inputs/outputs from the network? Although more formalization would be necessary for the DNN watermarking problem, the basic security measures are still valid, especially because in many black-box scenarios it is possible to obtain a large number of input/output pairs.

TEMIT. This is the acronym suggested by Kalker to denote "Transform-Embed-Inverse Transform." In the case of multimedia content watermarking, a huge number of papers have been published that offer no other novelty than working in a new transformed domain. As the number of invertible transformations is infinite, the number of possible papers with this methodology is also infinite. There are two problems with this approach: 1) there is nothing really innovative, because practically all TEMITs use well-known embedding algorithms, prominently spread-spectrum [9]; 2) most of the transformations are not sufficiently studied, so that the distortion measurement (which in multimedia contents must follow perceptual principles) is done in unknown terrain. Unless this funeral is successful, a similar explosion is foreseeable in the watermarking of DNNs with varying applications or topologies, even if the model on which they rely is conceptually similar. For example, the discovery that watermarking all layers of a network makes it more robust should happen only once.

Misunderstanding performance. The performance of a multi-bit watermarking algorithm should not be measured by hiding specific contents but pseudo-random binary sequences. Many papers have been published in which the information that is hidden is a logo of the authors' institution and the information retrieval gets help from the human eye because the logo continues to be seen despite the noise. The performance in the case of data-hiding should be measured with the Bit Error Rate, which is the ratio between the number of correctly recovered bits divided by the total of bits, and no human eye should intervene in the process. In the case of zero-bit watermarking, performance should be measured with a Receiver Operating Characteristic (ROC) and not with respect to a specific operating point (e.g., Equal Error Rate, EER), a synthesis parameter (e.g., Area Under Curve, AUC) or a proxy (e.g., the Normalized Cross Correlation, NCC). In the case of DNNs, especially if they are used to classify images (where there is the possibility of watermarking via backdoors), there is a risk that we will see logos again acting as the keys that open the backdoors. From a cryptographic point of view, this is as bad an idea as using a birthday as password. In any case, a good practice would be to not demonstrate effectiveness with a particular image, but use a random selection of images taken from an existing database as a backdoor triggers. A promising (and challenging) alternative would be to use as triggers invisible watermarks embedded in input images as they would be harder to detect inside a forged blackbox that routes suspicious inputs to a non-copyrighted lower-performance model.

In our opinion, Kalker did not emphasize enough another bad practice that due to its importance should have been buried first:

Paying too little attention to existing theory. Relatively shortly after the first multimedia watermarking methods were proposed, a very comprehensive theory was available that included fundamental limits on the amount of information that could be hidden as a function of the embedding distortion and of attacks. Although the theory was exact only in ideal scenarios, the formalization allowed researchers to discover how to efficiently implement the "writing on dirty paper theorem" proposed by MH Costa in the 80's [8]. The result, known as Quantization Index Modulation (QIM) [3], which employed the key concept of *distortion compensation*, not only opened the door to the use of lattices in watermarking, but also found a long-sought solution to the problem of channel coding with lattices for traditional Gaussian channels in communications [11]. Step by step, researchers were putting the pieces of the puzzle together, finding intermediate solutions between QIM and the traditional spread spectrum and finding the corresponding capacity formulas. The result is a beautiful and coherent theory on which it is possible to build pragmatic embedding codes that can reach rates very close to Costa's capacity [10]. Unfortunately, many authors have continued to publish papers with the old spread spectrum ideas as if none of this theory existed, combining them with TEMIT described above and showing a superiority that is not theoretically justifiable.

The exciting possibility of developing a theory for DNN watermarking now opens before us. Although some concepts and constructions are directly importable from multimedia watermarking, the scenario is different because it is no longer about watermarking signals but functionals, and because dynamics is essential to the

extent that training data is often part of the information to be protected. Furthermore, redundancy in the parameter space can lead to multiple equivalent minima, something that does not usually occur in multimedia watermarking. Considering how easy it is to take off-the-shelf DNN models and start fiddling with them, and given our own experience, we strongly advise practically-oriented researchers to follow and build on the latest theoretical developments.

5 CONCLUSIONS

Twenty five years after multimedia watermarking started gaining momentum, a new opportunity is emerging in the field of DNNs. From a practical point of view, there is a need for tools to protect IPR of deep models, and watermarking is an excellent candidate to be part of the answer, as it has already been proved useful in protecting software, digital layouts or IP cores. Some existing publications that propose watermarking for DNNs anticipate what might be a new golden age for watermarking research. As discussed in our paper, despite the similarities with media watermaking, DNN watermarking is quite different for two main reasons: the huge amount of degrees of freedom afforded by the weights of the model, and the fact that instead of embedding information in signals, here the hosts are functionals. The latter reason has already opened the venue for dynamic watermarking, that is, triggering model actions by presenting specific inputs that serve as secret keys. In turn, dynamic watermarking naturally leads to black-box detection/decoding in which it is possible to learn whether the model is watermarked by simply querying it without looking inside.

Leveraging on these differences, in this paper we have presented four challenges for future research. All of them, to a greater of lesser extent, call for new theoretical perspectives and tools, since if and how the existing theory for media watermarking can be translated to this new scenario is yet unclear. This need is clear for dynamic watermarking and in assessing the capacity of DNN watermarks; in turn, the latter requires being able to define the range of possible attacks and the interplay with practical methods to achieve robustness. On the other hand, the fact that watermarking is now applied to functionals, generates a richer space of interactions with their inputs and outputs; indeed, it is both possible to use watermarked inputs in order to trigger desired outputs and to produce watermarked outputs to carry over the copyright protection.

We hope that these challenges will foster new research in this promising new area. To avoid repeating past bad practices, we have added a section in which we bury them down and discuss where the focus should be. Let us bear in mind that "those who cannot remember the past are condemned to repeat it".

ACKNOWLEDGMENTS

Work of FPG is partially funded by Agencia Estatal de Investigación (Spain) and the European Regional Development Fund (ERDF) under project RODIN and by Xunta de Galicia and the European Regional Development Fund (ERDF) under project ED431G2019/08. Work of MB and BT is partially supported by the Italian Ministry of University and Research (MUR) under the PRIN 2017 2017Z595XS-001 program - PREMIER project.

REFERENCES

[1] M. Barni, F. Bartolini, A. De Rosa, and A. Piva. 1999. Capacity of the watermark channel: How many bits can be hidden within a digital image?. In *Security and Watermarking of Multimedia Contents. International Society for Optics and Photonics*.

[2] F. Cayre, C. Fontaine, and T. Furon. 2005. Watermarking security: theory and practice. *IEEE Transactions on Signal Processing* 53, 10 (2005), 3976–3987. https://doi.org/10.1109/TSP.2005.855418

[3] Brian Chen and Gregory W. Wornell. 2001. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Transactions on Information Theory* 47, 4 (May 2001), 1423–1443.

[4] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 105–113.

[5] J. Y. Cheong and I. K. Park. 2017. Deep CNN-Based Super-Resolution Using External and Internal Examples. *IEEE Signal Processing Letters* 24, 8 (2017), 1252–1256. https://doi.org/10.1109/LSP.2017.2721104

[6] P. Comesaña, L. Pérez-Freire, and F. Pérez-González. 2005. Fundamentals of Data Hiding Security and Their Application to Spread-Spectrum Analysis. In *Information Hiding. IB 2005, Lecture Notes in Computer Science*, Vol. 3727. Springer. https://doi.org/10.1007/11558859_12

[7] Betty Cortiñas-Lorenzo and Fernando Pérez-González. 2020. Adam and the Ants: On the Influence of the Optimization Algorithm on the Detectability of DNN Watermarks. *Entropy* 22, 12 (2020). https://doi.org/10.3390/e22121379

[8] Max H. M. Costa. 1983. Writing on Dirty Paper. *IEEE Transactions on Information Theory* 29, 3 (May 1983), 439–441.

[9] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. 1997. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing* 6, 12 (December 1997), 1673–1687.

[10] Uri Erez and Stephan ten Brink. 2005. A Close-to-Capacity Dirty Paper Coding Scheme. *IEEE Transactions on Information Theory* 51, 10 (October 2005), 3417–3432.

[11] U. Erez and R. Zamir. 2004. Achieving 1/2 log (1+SNR) on the AWGN channel with lattice encoding and decoding. *IEEE Transactions on Information Theory* 50, 10 (2004), 2293–2314. https://doi.org/10.1109/TIT.2004.834787

[12] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).

[13] F. Huszar, L. Theis, W. Shi, and A. Cunningham. 2017. Lossy Image Compression with Compressive Autoencoders. In *International Conference on Learning Representations (ICLR 2017)*. https://doi.org/110.17863/CAM.51995

[14] A. L. McKellips I. J. Cox, M. L. Miller. 1999. Watermarking as communications with side information. *Proc. IEEE* 87, 7 (1999), 1127–1141.

[15] F. Kokkinos and S. Lefkimmiatis. 2018. Deep Image Demosaicking Using a Cascade of Convolutional Residual Denoising Networks. In *Computer Vision – ECCV 2018. Lecture Notes in Computer Science*, Vol. 11218. https://doi.org/10.1007/978-3-030-01264-9_19

[16] Yue Li, Benedetta Tondi, and Mauro Barni. 2020. Spread-Transform Dither Modulation Watermarking of Deep Neural Network. arXiv:cs.CR/2012.14171

[17] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 273–294.

[18] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440* (2016).

[19] Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2018. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval* 7, 1 (2018), 3–16.

[20] S. Nah, T. H. Kim, and K. M. Lee. 2017. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 257–265. https://doi.org/10.1109/CVPR.2017.35

[21] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[22] C. I. Podilchuk and E. J. Delp. 2001. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine* 18, 4 (2001), 33–46.

[23] B Rouhani, H Darvish, and F Chen. 2019. Deepsigns: an end-to-end watermarking framework for protecting the ownership of deep neural networks. In *The 24th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems, Rhode Island, USA*.

[24] Enzo Tartaglione, Enzo Grangetto, Davide Cavagnino, and Marco Botta. 2021. Delving in the loss landscape to embed robust watermarks into neural networks. In *International Conference on Pattern Recognition (ICPR)*. 273–294.

[25] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017*

ACM on International Conference on Multimedia Retrieval. 269–277.

[26] H. Wu, G. Liu, Y. Yao, and X. Zhang. 2021. Watermarking Neural Networks with Watermarked Images. *IEEE Transactions on Circuits and Systems for Video Technology* (2021). https://doi.org/10.1109/TCSVT.2020.3030671

[27] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. 2018. Generative Image Inpainting with Contextual Attention. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 5505–5514. https://doi.org/10.1109/CVPR.2018.00577

[28] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. 2020. Model Watermarking for Image Processing Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 07 (Apr. 2020), 12805–12812. https://doi.org/10.1609/aaai.v34i07.6976

[29] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. 2018. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 159–172.

[30] K. Zhang, W. Zuo, and L. Zhang. 2018. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Transactions on Image Processing* 27, 9 (2018), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891