IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 4, April, 2026, pp. 1273-1284
Journal homepage: http://iieta.org/journals/isi

Protecting Intellectual Property of Deep Learning Models: A Structured Literature Review with Taxonomy of Watermarking Approaches

Zinah S. Abduljabbar[1*], Muhanad Tahrir Younis[2]

[1] Department of Computer Science, Mustansiriyah University, Baghdad 10052, Iraq
[2] Department of Artificial Intelligence, Mustansiriyah University, Baghdad 10052, Iraq

Corresponding Author Email: zinahsadeq@uomustansiriyah.edu.iq

https://doi.org/10.18280/isi.310422

Received: 21 November 2025
Revised: 25 January 2026
Accepted: 17 April 2026
Available online: 30 April 2026

Keywords:
intellectual property, deep learning, watermarking, ownership verification, embedding phase, MLaaS, security

ABSTRACT

The increasing commercial value of deep learning (DL) models has created an urgent need to protect their intellectual property (IP). Unlike traditional software, DL models rely on massive training datasets, high-performance hardware, and specialized expertise, making them valuable assets. Among various IP protection strategies, digital watermarking has emerged as a promising solution. This structured literature review (SLR) presents a comprehensive five-attribute taxonomy of watermarking approaches, organized around the embedding phase, verification scenario, protection type, capacity, and protected content. By prioritizing the embedding phase as the primary organizing principle, this review identifies critical design considerations for proactive and reactive ownership protection. The review synthesizes findings from 2017 to 2025, analyzing pre-training, training, and post-training embedding strategies, and comparing white-box, black-box, gray-box, and no-box verification scenarios. Key observations include the dominance of passive watermarking, the limited adoption of multi-bit schemes in black-box settings, and emerging techniques for generative and image-processing networks. Comparative analyses of methods highlight trade-offs in robustness, fidelity, capacity, and computational efficiency. The SLR also identifies challenges, such as vulnerability to pruning, fine-tuning, and adaptive attacks, and proposes future directions including multi-bit encoding, gradient-free optimization for black-box verification, and expansion to non-classification models like GANs and large language models. Overall, this review provides a structured roadmap for researchers and practitioners seeking to secure DL model IP in cloud-based and MLaaS environments, emphasizing both legal and technical considerations.

1. INTRODUCTION

In recent years, machine learning (ML) techniques such as deep learning (DL) have played a vital role in many important applications in our lives, including but not limited to education, social media, and healthcare. Learning a model from scratch requires expensive hardware, considerable effort, specialized expertise, and a significant amount of time. The performance of these models depends on the quality and quantity of data (training data) provided to the model. Collecting and labeling data requires a significant amount of effort and time. Hence, training data is a valuable asset, and this makes us recognize that the ML model is a valuable IP while considering all ways to protect these models and ensure their use in compliance with applicable legislation and ethical rules related to privacy.

The idea of watermarking has been applied to multimedia content for decades to provide content authentication, tamper detection, and copyright protection [1, 2]. The basic watermarking techniques introduce covert information into the media and then use the information to verify ownership [3]. But these traditional approaches are not easily transferable to DNNs because they are fundamentally different from static media files and are multi-layered with millions of trainable parameters.

Several methods have emerged to protect model ownership verification, such as blockchain and hashing technologies, where the evidence is created when the parameters are given. In the previous technologies [4], a model plagiarizer may also generate evidence by using the same technologies, perhaps before the legitimate owner records the proof of ownership. The owner should create the evidence early during the learning process when the parameters of the model are yet to be defined [5].

The structured literature review (SLR) proposes a multiple-perspective taxonomy based on five attributes to classify and compare the most important existing works. The embedding phase is the primary direction in our taxonomy due to the lack of attention paid to it in previous works. However, the embedding phase is a foundation element and can determine many aspects of the watermarking scheme.

The remainder of SLR is organized as follows: Section 2 provides an overview of the preliminaries of deep neural network (DNN). Section 3 presents the requirements of DNN watermarking techniques. Section 4 presents the proposed taxonomy. Section 5 reviews the related works according to

the proposed taxonomy. Section 6 introduces the challenges and suggestions for the future outlook. Section 7 is the conclusion.

2. PRELIMINARIES OF DEEP NEURAL NETWORK

In this section, we briefly describe ML and DL and then explain the DNN deployment modes and associated attack types.

2.1 Machine learning and deep learning

ML and DL are subfields of artificial intelligence (AI) used to solve many problems [6]. They are about developing data-driven algorithms, which means that predictions, pattern recognition, and other complex tasks are performed by learning from massive amounts of data without explicit programming [7, 8]. Using data is the fundamental idea behind building ML models to generalize, adapt, and enable automated decision-making, which involves learning from experience and improving the performance of specific tasks by gaining more practice and accumulating knowledge [9]. A DNN is a type of ML model inspired by neural processes. It consists of an input layer for receiving raw data, an output layer for making decisions or predictions, with several hidden layers in between. DL is based on neural network topologies, known as a DNN [10]. Each layer consists of interconnected neurons with many parameters that process the input data and pass the information forward until the output layer produces the output. The fundamental distinction between ML and DL lies in the latter's ability to represent the world as a nested hierarchy of concepts, resulting from the larger number of layers learned at a higher level of abstraction compared to the ML model.

2.2 Deep neural network distribution mode

DL is a branch of ML that uses multi-layer artificial neural networks to learn from data. Artificial neural networks excel in many fields, including autonomous systems, computer vision, and natural language processing, to name a few. Many subtle considerations when building DNNs require powerful computing resources, high-quality data, and expertise in DNNs. Figure 1 illustrates the two model deployment modes. Machine Learning as a Service (MLaaS) is on the right of the figure, and on the left is model distribution. MLaaS is a cloud deployment model that users can access via application programming interfaces (APIs). At the same time, a complete copy of the model is distributed to the end user in distributed mode, meaning the user has full access to the trained model.

Figures 2 and 3 show that threats vary depending on the deployment type. Model extraction is a common attack in MLaaS mode, where the adversary has access only to the host model API (i.e., a black-box setup). The attacker sends multiple query samples to the API and combines the prediction results, leading to the training of a surrogate model that is very similar to the original model's behavior [11]. Therefore, effective protection techniques must consider factors such as inference perturbation to ensure resistance to model extraction. In model distribution mode, the adversary has full access (white-box setup) and can obtain a copy of the host model, performing a model transformation attack to generate a stolen model. This type of attack can be classified into model fine-tuning and model compression [12, 13].

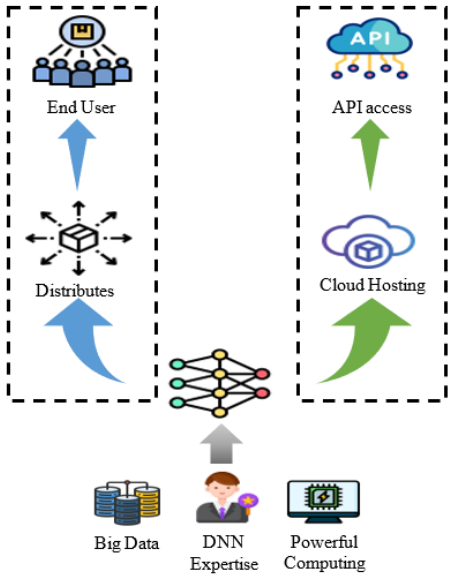

Figure 1. Two deployment modes for deep neural network (DNN) models: (left) model distribution where a full copy is given to the end user; (right) MLaaS, where users access the model via application programming interface (API) only

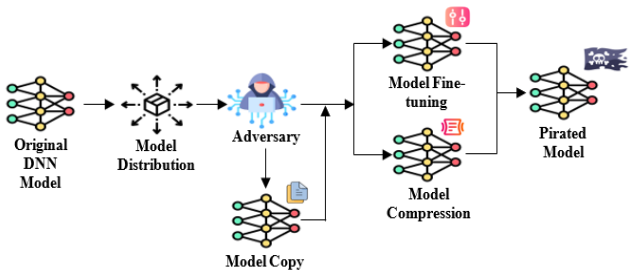

Figure 2. Threats in the model distribution mode (white-box access): Adversary can obtain a full model copy, then apply fine-tuning, compression, or direct copying

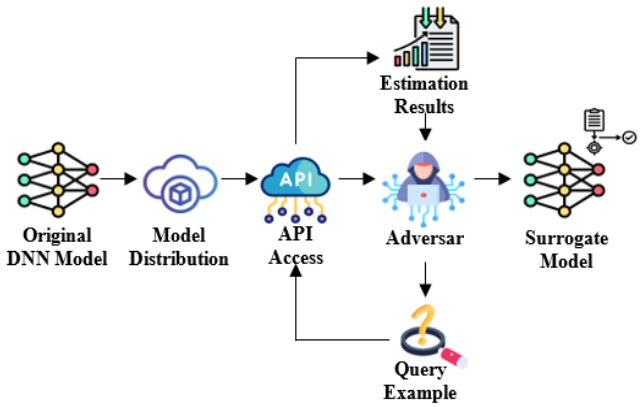

Figure 3. Threats in MLaaS mode (black-box access): Adversary queries the application programming interface (API) and uses responses to train a surrogate model

3. REQUIREMENTS OF DEEP NEURAL NETWORK WATERMARKING TECHNIQUES

Designing an effective digital watermark for protecting the

IP of DNNs is a challenging task. Besides the fundamental trade-off between robustness, fidelity, and capacity, several other functional requirements must be considered.

Robustness: means the ability to correctly extract the embedded watermark even when the host model encounters a modification scenario. The modification may involve optimization or malicious purposes, and in both cases, the robustness must ensure the watermark persists. Below are two types of modifications that watermarking should be robust against.

A. **Network pruning** refers to a common strategy that includes element-wise pruning (weight pruning) and structural pruning (neuron pruning). Weight pruning is achieved by removing values below a certain threshold value, while neuron pruning aims to remove some neurons that are considered less important. Reducing overall complexity is always the goal behind this strategy to improve the efficiency of the models and make them more suitable for deployment in low-power devices [14].

B. **Model adaptation** is a broad term that includes both fine-tuning and transfer learning, two related but different concepts in DL. Fine-tuning, the model weights are partially or completely updated to adapt to a new task using a new dataset, while transfer learning can be called the reuse of knowledge across tasks, i.e., exploiting the knowledge embedded in a previously trained model for a new, different but related task rather than training it from scratch [15].

Fidelity: is an important factor that guarantees the presence of a watermark does not significantly affect the model performance, so high fidelity means imperceptibility for the watermark. Since DL models are important in many aspects of real life, if the protection scheme leads to a large deviation, it may affect the performance of the model and thus affect usability and reliability, so it is important to consider this property when designing watermarking schemes [16].

Capacity: means embedding watermark information into the host; high capacity is one of the most important factors in model protection and ownership verification, but this property can conflict with accuracy and robustness. Depending on the verification scheme, when only the watermark information is verified, this indicates a single-bit system, whereas when detailed information is extracted, it is a multi-bit system. Researchers have made many efforts in this area, for example, in the study [17], the authors designed a dual-embedded watermark model for efficient location identification and achieved a good result from the perspective of capacity, fidelity and robustness.

In the study [18], the authors proposed a watermarking scheme depending on optimal embedding locations by identifying the important weights that affect the model functions and using them for embedding during the training phase.

Security: The watermark does not leave noticeable imprints in the target model, so an unauthorized individual cannot identify the watermarked model from the unwatermarked one [19].

Generality: The watermarking scheme should apply to various architectures, datasets, verification scenarios, and computing platforms [20].

Integrity: Watermark integrity guarantees that unmarked models are not falsely identified as watermarked, consequently preventing misuse or false claims of ownership [21].

Efficiency: The computational cost of watermark embedding and extraction operations should be minimal and fast [20].

Reliability: The watermarking scheme should ensure accurate and consistent detection as well as prevent ownership disputes or false claims [19].

4. A PROPOSED CLASSIFICATION FOR DEEP NEURAL NETWORK IP PROTECTION TECHNIQUES

For this SLR, a careful search strategy was used to identify relevant studies from 2017 to the present. In this SLR, we propose a methodological classification that captures all the attributes adopted in previous works, categorizing them into five attributes, each of which is further categorized, as shown in Figure 4. This classification reveals different perspectives, and the proposed taxonomy provides a framework to address the following guiding questions:

Q1: When is the embedding process first introduced in the model pipeline?

Q2: How can copyright ownership be verified?

Q3: When is the protective action triggered?

Q4: What is the amount of information the watermark can carry?

Q5: What is the primary function of the protected model?

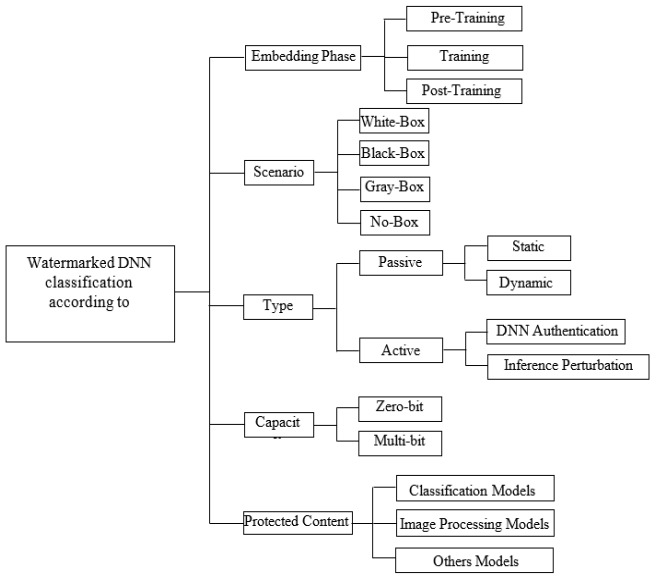

Figure 4. Proposed five-attribute taxonomy for deep neural network (DNN) IP protection: Embedding phase, verification scenario, type, capacity, and protected content

4.1 Embedding phase

To answer the important question "**When is the embedding process first introduced in the model pipeline?**", we classify the embedding process into three phases: pre-training, during training, and post-training, each with its benefits in terms of security, robustness, and efficiency, as well as the application domain for model implementation.

4.2 Scenario of verification

The scenario or access granted answers this question: **"How can copyright ownership be verified?".** Depending on

access granted, watermarking methods are primarily categorized into white-box and black-box, as seen in Figure 5, with two additional categories represented by gray-box and no-box.

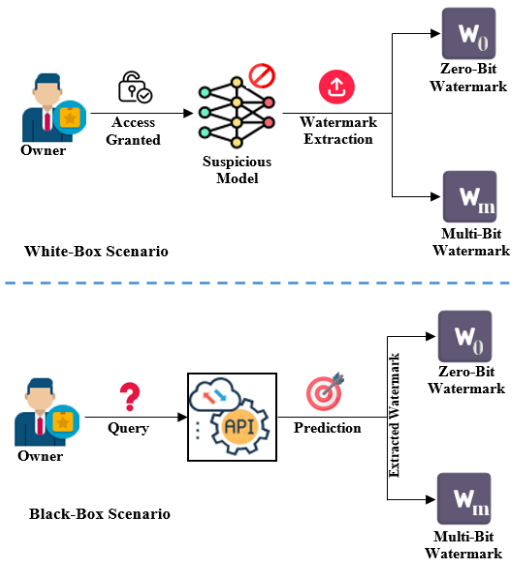

Figure 5. Two primary verification scenarios: (top) white-box requires access to internal parameters; (bottom) black-box uses application programming interface (API) queries and trigger sets

4.3 Type

DNN watermark type can be classified into active protection and passive protection, and determining the type is the appropriate answer to the following question: "**When does pattern protection start?**". Passive (reactive) protection establishes ownership after any violation is detected, in contrast to active (proactive) protection, which prevents unauthorized use.

4.4 Capacity

The amount of information embedded within the model using the watermarking method for verification and intellectual property protection purposes is called "capacity." Watermarking methods can be classified according to capacity into zero-bit and multi-bit schemes, depending on whether the verification system focuses only on the presence of the watermark or on a specific sequence of bits for verification.

4.5 Protected content

Targeted watermarking models can be classified according to their functional role. Most models in this direction are classification models, but others perform different tasks. This variation in functional role highlights the need to study the current distinction between watermarking strategies and output types.

As we all know, watermarking technology is used to protect the copyright of various types of multimedia content, such as audio, video, and images. This inspiration led authors to use this technology to protect ML models, which are considered valuable IP due to their high cost and time. Unlike multimedia files, DNNs are complex, multi-layered structures consisting of many layers of interconnected neurons with millions of parameters, so traditional watermarking methods cannot be applied to protect these models. Depending on the deployment method, whether through distributed architectures or APIs, as described earlier, the watermarking technology must address both types of deployment. In the sections below, a comprehensive review of current works in this field is discussed with a comparative table according to our classification.

5. RELATED WORKS ACCORDING TO THE PROPOSED TAXONOMY

5.1 Embedding phase

5.1.1 Pre-training

In our classification, we consider any alteration in the data training or the model architecture before the training phase to be a pre-training phase, although we know that watermarking only becomes effective after training, and this provides a complementary perspective to existing taxonomies. The authors [22] propose a key-based protection method that involves block-wise pixel shuffling of the input images in the preprocessing stage. As a result, the model becomes unusable with an incorrect key. In the study [23], the authors propose a method to protect the IP of DNNs based on the chaotic encryption method by exchanging the positions of weights within the fully connected layer, instead of encrypting the weight values, while maintaining the model's accuracy and returning incorrect predictions when accessed without valid decryption keys. In the study [24], the authors propose a method to protect DNNs against forgery attacks by generating a series of specifically labeled trigger samples using a one-way watermark hash function. Then, during training, the model learns to associate the set of triggers with specific outputs, making it hard for an attacker to forge these trigger samples. In the study [25], the authors propose a method to protect the IP of DNN by converting the training image into a block-wise encrypted version with a secret key to embed the watermark information. This method is resistant to piracy attacks and does not require a special watermark trigger set.

We can conclude that these four previously mentioned works use data encryption as the core of their security strategy, with each of these methods using a secret-driven transformation to bind the model's behavior to the private key, while the following three works represent the change in the model architecture. The authors [26] propose a passport-based method to verify DNN ownership by adding a passport layer after each convolution layer. This method allows the model to perform well when verifying the passport; otherwise, the model's performance deteriorates. This method is robust against network tampering and ambiguity attacks, but the passport layer can impose an overhead. In the study [27], the authors propose a structure-based watermarking method by leveraging DNN channel pruning to embed the watermark instead of changing the parameters. To this end, watermark bits are split into several segments used to control a pruning rate and then applied to the convolutional layers model. By analyzing the channel pruning rate, the watermark can be retrieved for verification purposes. In the study [28], the authors propose a watermarking method by training a neural network called HufuNet, which is divided into two parts. The left part is embedded in the target model using a secret key and serves as a watermark, while the right part is kept by the owner

and then combined with the extracted watermark to verify ownership. The final work in this section [29] is considered as a hash-based mechanism, and the authors present an approach for protecting the IP of image captioning models using two different embedding schemes by embedding the key in the hidden memory state of a recurrent neural network (RNN), Their embedding process ensures a paralyzed model with a forged key and resistance to removal and ambiguity attacks.

5.1.2 Training

Watermark embedding occurs during the model's initial training. To achieve this, labels, gradients, or specially designed loss functions must be modified, in which case the watermark becomes part of the model's internal behavior or parameters. The straightforward idea is to exploit the over-parameterized property in DNNs to embed the watermark. The first attempt to implement digital watermarking in a DNN model for protection purposes was made by Uchida et al. [20]. Their algorithm allows for embedding watermarks in the parameters and training the model with an additional regularization loss term. The authors [30] improve Uchida et al.'s method by using the host neural network and the independent neural network and trained them together. The second network is used to embed the watermark information in the host network; the watermarked network is released, while the independent network is kept secret. This work can't avoid the ambiguity attack. In the study [31], the authors propose a method based on choosing a random set of model parameters as a watermark and during training, the weights/parameters are frozen. In this work, the loss function is intended to maximize the network's sensitivity to changes in the watermarked weights, which leads to increased capacity and robustness, which is the advantage of this method. In the study [32], the authors introduce an adversarial training to embed a robust and covert watermarking approach for a white-box scenario. They construct an automated approach through two-party competition, where the generator performs the watermarking process using an adversarial learning network, and the discriminator is a watermark detector. Instead of direct weight modification, authors [33] use the transformed weight modification in fully-connected layers of fine-tuned models to embed the watermark information using discrete cosine transform (DCT) and a quantization index modulation (QIM) instead of a loss function, and their method ensures less weight value modification and less impact on model performance. In the study [19], the authors propose a watermarking framework named DeepSigns based on dynamic content (activations), which is both data and model-dependent and supports both white-box and black-box scenarios to protect the IP of DNNs. During training, DeepSigns embeds the watermark gradually using the probability density functions of the activation set. Access to model parameters is required in white-box verification, while in the second scenario, a set of inputs is used to trigger the watermark information remotely.

Depending on the properties of DNN, which is represented by the non-convex optimization problem and the over-parameterization, which leads to more than one solution, the authors [34] propose a method that is the first to exploit the gradient of the cross-entropy cost function to embed the owner's signature. This method is considered a black-box verification since it allows remote verification of the watermark using a prediction API. In the study [35], the authors present a method to verify model ownership by inserting external features based on style transfer instead of training image features as trigger input; their approach cannot be used in a real-world API because the set of triggers uses abstract patterns instead of valid images, which is a constraint. In the study [36], the authors propose watermarking framework for image transformation network where both the inputs and the outputs are images, the host neural network and the watermark extraction network are trained together using a combined loss function, the output from the watermarked model also includes a watermark, so in this method in addition to identifies the copyright of the original neural network it also verifies whether the image is generated by original neural network. In addition to the above techniques, the backdooring technique is commonly used for embedding a watermark into a behavioral model's response to a carefully designed trigger set. This method is hidden during normal use and can be activated later to verify ownership. The works below in this section are backdoor-based watermarking. In the study [37], the authors propose a method to implant and learn three types of watermark key generation at the training phase, which are then used as the watermark trigger set. The key generation algorithm involves training data with additional meaningful content, choosing irrelevant images from another dataset, and overlaying images with random noise. In the study [38], the authors propose a method to resist autoencoders that can remove critical samples. Their method uses unmodified training images as a key but gives them different labels than the original prediction. To overcome the modification processing, they use exponential weights in their work, where the predictions are based on large absolute values.

To overcome watermarking techniques that are represented by inserting outlier input-output pairs as a watermark into a model, the authors in the study [39] propose a technique called entangled watermarking embedding (EWE), which allows the model to learn the classification task with watermarks encoding together, and as a result, any attempt to remove the watermarks degrades the performance of the model. In the study [40], the authors propose a method to protect the IP of DNNs where trigger samples at the classification boundary are automatically generated using conditional generative adversarial networks (CGAN) and leverage chaotic automatic annotation for labeling in the training phase. during verification, the predictions on the trigger set are matched to the chaotic label. Their method is resistant to various attacks and maintains the model's performance.

5.1.3 Post-training

The "post-training" phrase refers to applying watermarking techniques after a model has already been trained, and this is done through weight modification, fine-tuning, or parameter encryption. This approach is compatible with pre-trained or externally sourced models, allowing for efficient and flexible integration without the need to retrain models from scratch. One mechanism used in post-training is backdoors, where this technique, as in all watermarking techniques, consists of two stages: embedding and verification. In the first stage, the owner embeds the backdoor and maintains the functionality of the model, while in the second stage, the validation samples can be used in the verification stage. The authors in the study [41] present an example of watermarking-based backdoors in post-training to protect the IP of DNNs where the embedding process is applied at the API level to predict by changing a small portion of the query responses which results in the watermark being embedded in any surrogate models trained through the API query, and the model owner can confirm the

presence of the watermark in any suspected surrogate method by querying it and comparing the prediction to backdoor tags.

In the study [42], the authors propose a robust DNN watermarking framework based on a bi-level optimization approach. The inner loop produces robust watermark exemplars while the outer loop maintains the model functionality by optimizing the model parameters using a masked adaptive approach. Their work focuses on modifying a small subset of parameters to minimize the watermarking overhead.

In other works, the authors also exploit fine-tuning as a post-training technique by slightly updating a pre-trained model with a small set of tailored inputs. Embedding is performed by changing the model's behavior and maintaining its performance without retraining it from scratch. To protect the IP of generative adversarial networks (GANs), the authors in the study [43] propose a supervised watermarking method to ensure that all images generated by GANs include an invisible watermark and can be extracted for verification purposes through pre-trained decoders. The embedding is done by fine-tuning a fully pre-trained GAN instead of retraining it from scratch; this results in reduced computation, the model retains its performance, and it is scalable to complex models. In the study [44], the authors introduce watermarking approach using adversarial examples as the watermark key and this lead an alteration in classifier decision boundary to specific shape when attaching a small perturbation to the input example and as a result an incorrect prediction is the output of the modified example. In the study [45], the authors propose a Pivotal Tuning watermarking method for a pre-trained image generator such as StyleGAN2 and StyleGAN3 as a post-processing step, which reduces the time compared to retraining from scratch. In the study [46], the authors utilize encryption as a technique to encode a few parameters with the most significant impact on model performance using adversarial perturbation. During encryption, a secret key is generated and only authorized users are allowed to decrypt and use the model. In the study [47], the authors propose a framework to protect the IP of the model and resist attacks from surrogate models using spatial invisible watermarks to embed watermarks into the resulting images using sub-networks for embedding and extraction. In the study [48], the authors propose an embedding method by encoding and spreading the trained model's weight information through a wide range of weights. Code division multiple access (CDMA) is used to embed the watermark information directly into the model weights.

5.2 Scenario

5.2.1 Whit-Box

This scenario allows the verifier to verify or extract watermarks by directly examining the model's internal parameters. Many works rely on directly embedding recognizable values into the model parameters, requiring these parameters to be examined to retrieve watermarks, as introduced in the studies [27, 31, 48]. Other works are key-based methods that rely on secret encryption or decryption keys to protect and verify ownership, as in the study [23]. Other types are extractor-based methods that involve an extractor trained with the host model to learn for extraction, as shown in the studies [30, 32].

Most works in this scenario are multi-bit watermarks and are verified by extracting a message containing bits, while in zero-bit watermarks [23], verification is achieved by a binary decision to confirm the presence of a specific watermark. Although this scenario forms a powerful foundation for ownership verification, it requires the verifier to access internal model parameters, making it difficult to implement in real life, such as in a cloud-based application where the model is accessed via APIs.

5.2.2 Black-Box

In this scenario, the verifier can only interact with the model via its prediction API and observe the model's responses to specific queries, which means that the verifier does not have access to the internal model's parameters or architecture. The idea behind this scenario is to embed watermarks by initializing specific trigger sets. Pairs of inputs and labels are uniquely designed to derive specific, variable outputs, and then these trigger sets are used for verification remotely.

Black-box techniques range from label modification as in the studies [24, 40, 41] to label transformation within training data [22, 25] and injecting adversarial or synthetic inputs as in the studies [37, 42, 44].

5.2.3 Gray-Box

A hybrid approach to watermarking and a dual-level integration between white-box embedding and black-box validation, it combines characteristics from two scenarios, making use of the white box access to embed a watermark directly in the model's internal structure, and relying on black-box access to verify ownership by querying the model externally and observing the model's behavior. Unlike the traditional black-box scenario, which relies on creating external trigger sets for validation, the gray-box encodes the watermark into the model's behavior, and validation is achieved through patterns in output predictions or statistical analysis. The best example of this scenario is the work in the study [28], where the authors use two neural network slices and rely on model reconstruction and accuracy comparison as a new verification methodology that differs from the methodology used in the black-box scenario. Many works functionally perform a hybrid approach, but not all are classified as gray-box since they often use traditional backdoors and trigger sets. The work in the study [34] is an example of work that can be considered a gray-box functionally.

5.2.4 No-Box

This is an emerging scenario in IP protection. This model allows verification of ownership without direct interaction with the model itself and does not require internal access or API queries. This model can be seen in the studies [45, 46], where the verification in these works is no-Box.

5.3 Type

The widespread use of DNNs in many vital sectors of our daily lives has raised concerns about their IP protection. A well-trained model is considered a significant investment of resources, including the massive data used in training, computational power, and expert knowledge. Therefore, safeguarding these models against illegal copying and redistribution has become a vital challenge at present. IP protection of DNNs can be categorized into active and passive, and each type is subdivided as shown in Figure 4.

5.3.1 Active DNN IP protection

Active IP protection for DNNs refers to a proactive mechanism designed to prevent and directly influence the model's behavior under unauthorized use. This type of protection can be divided into DNN authentication and inference perturbation. The former focuses on authorized use of the model, ensuring legitimate users have a valid authentication key, while preventing unauthorized users, significantly degrading model performance, as in the studies [22, 23, 26, 29].

Unlike DNN authentication, inference perturbation introduces controlled noise or distortions into API outputs during inference, to degrade the stolen model's performance and defend deployed models from extraction attacks. This type is suitable in an MLaaS setting, as in the study [41].

5.3.2 Passive DNN IP protection

Passive IP protection for DNNs refers to a reactive mechanism focusing on ownership verification after IP violation is observed, rather than actively blocking unauthorized usage during deployment. It embeds hidden proofs into the model's parameter or functional behavior, and depending on the embedding location, this type of protection can be divided into static and dynamic. In static passive protection, the ownership information is embedded into fixed elements such as weight or internal parameters, as in the studies [20, 27, 28, 30, 33, 40, 44, 48]. In contrast, in dynamic passive protection, the embedding is performed into the model's behavior as in the studies [19, 24, 25, 34, 37].

Table 1. Attribute combination matrix: Verification scenario vs. Active/Passive

	Active	Passive
White-Box	[23, 26, 29]	[20, 27, 30, 33, 48]
Black-Box	[22]	[24, 25, 37, 39, 44]
Gray-Box	—	[28, 34]
No-Box	[46]	[45]

The passive type is suitable for supporting robust legal evidence in cases of IP violation. Active and passive protection are independent of verification scenarios; any combination is possible, demonstrating that these attributes are orthogonal dimensions, as shown in Table 1.

5.4 Capacity

Depending on the accessibility that allows extracting embedded information from the model's parameters in a white-box setting that supports high watermark capacity. In contrast, black-box verification is often a zero-bit watermarking scheme. As shown in Table 2, many of the works that are based on black-box verification are zero-bit watermarking, and this is due to limited access in this scenario to only the API, and mostly depend on trigger-based or behavior-based watermarking. However, some progressive methods have proven that it is possible to extract multiple bits of a watermark in a black-box scenario, such as in [24, 36, 37, 47].

5.5 Protected content

The objective model for most watermarking methods is often a classification model, as shown in Table 2; despite that, many works support various targets such as image processing models, image captioning models, and image generation models.

The authors of the study [29] introduce a method for protecting an image captioning model by embedding a secret key in the hidden memory state of the RNN. a task-agnostic frameworks in two works [36, 47] present progress in IP protection for generative and transformation models that perform image processing tasks by embedding a watermark directly into the resulting image. For protecting image synthesis models, the authors [43, 45] introduce strategies for ownership verification via image-based watermark extraction.

Table 2. Comparison of existing IP protection methods for deep neural network (DNN) models

Paper No.	Embedding Phase	Mechanism	Verification Method	Type	Behavior	Capacity	Protected Content	Attack Resistance	Key Limitation
[26]	Pre-Training	Passport-Based layer	White-Box & Black-box	Active	DNN authentication	Multi-bit	Classification model	Resists ambiguity & removal (fine-tune/prune)	Extra inference cost (~10%) & passport distribution needed
[22]	Pre-Training	Data encryption using pixel shuffling with a key	Black-Box	Active	DNN authentication	Zero-bit	Classification model	Resists fine-tuning attacks with small dataset	Low key space; vulnerable to brute-force or key estimation attacks
[23]	Pre-Training	Encryption-based method	White-Box	Active	DNN authentication	Zero-bit	Classification model, segmentation, NLP	Resists brute-force & side-channel attacks	Secret key leakage risk; requires on-chip decryption
[24]	Pre-Training	Trigger-set based on one-way hash chaining	Black-Box	Passive	Dynamic	Multi-bit	Classification model	Resists forging (ambiguity) attacks; robust to fine-tuning	Large trigger set size needed (e.g., 38 for 10 classes); hash chain management
[25]	Pre-Training	Data Encryption/ block-wise image transformation	Black-Box	Passive	Dynamic	Zero-bit	Classification model	Resists fine-tuning, pruning (up to 60%), and piracy attacks	Large block size (M≥8) degrades watermark detection; key security assumed
[27]	Pre-Training	Structure-based channel pruning with quantization index modulation	White-Box	Passive	Static	Multi-bit	Classification model	Robust to pruning/fine-tuning	Non-blind & vulnerable to distillation
[28]	Pre-Training	Parameter embedding of	Gray-Box	Passive	Static	Multi-bit	Classification model /NLP	Robust to pruning/fine-	Requires restore (cosine similarity/SVD);

		convolution kernels from a split watermark model						tuning/structure attacks	complex key management
[29]	Pre-Training	Embedding a secret key into a hidden memory state	White-Box	Active	DNN authentication	Multi-bit 512	Image captioning model	Robust to pruning/fine-tuning/ambiguity attacks	Vulnerable if attacker knows full training details (key & signature fine-tuning)
[20]	Training Phase	Embedding a binary bit vector into weights with a custom regularization function	White-Box	Passive	Static	Multi-bit (256, 512, 1024, 2048 bits)	Classification model	Robust to pruning (up to 65%) & fine-tuning	Vulnerable to watermark overwriting & network morphism
[37]	Training Phase	Backdoor-based watermarking	Black-Box	Passive	Dynamic	Multi-bit	Classification model	Robust to pruning/fine-tuning/model inversion	May cause false positives; watermark pattern can be detected by defense mechanisms (e.g., MagNet)
[19]	Training Phase	Loss regularization using activation distributions	White-Box & Black-Box	Passive	Dynamic	Multi-bit up to 128 bits	Classification model	Robust to pruning/fine-tuning/overwriting	Requires GMM assumption; hyperparameter sensitive
[38]	Training Phase	Label change with exponential weighting	Black-Box	Passive	Static	Multi-bit	Classification model	Robust to pruning/retraining & query modification (autoencoder)	Requires hyperparameter tuning (T); key samples detectable via non-watermarked model comparison
[30]	Training Phase	Joint training with auxiliary loss function	White-Box	Passive	Static	Multi-bit 256 to 4096 bits	Classification model	Robust to pruning/fine-tuning	Vulnerable to watermark overwriting/ambiguity attacks via an independent network
[31]	Training Phase	Parameter freezing with loss landscape optimization	White-Box	Passive	Static	Multi-bit 3306 bytes	Classification model	Robust to fine-tuning and compression (quantization)	Requires replica generation during training
[36]	Training Phase	Output-based watermarking via joint training with watermark decoder	Black-Box	Passive	Dynamic	Multi-bit up to 256×256 color images	Image processing model	Robust to cropping and noise (if trained with augmentation)	Key sensitivity not perfect
[32]	Training Phase	Joint training with dual neural networks	White-Box	Passive	Static	Multi-bit binary strings or 128×128×3 images	Classification model	Robust to overwriting, fine-tuning (REFIT), and pruning (up to 99%)	No theoretical boundary for ownership claim
[33]	Training Phase	Quantization-based frequency domain embedding	White-Box	Passive	Static	Multi-bit 128 bit	Classification model	Robust against pruning attacks	Needs evaluation on multiple fine-tuning models to confirm if the initial modification alone is sufficient
[34]	Training Phase	Gradient Regularization	Gray-Box	Passive	Dynamic	Multi-bit 16, 32, 64 bits	Classification model	Robust to pruning, fine-tuning, quantization, adversarial fine-tuning, input noise, score rounding, score perturbation	High verification cost
[35]	Training Phase	Style transfer-based feature injection	White-Box & Black-Box	Passive	Dynamic	Multi-bit	Classification model	Resistant to fine-tuning, saliency-based detection, STRIP, and trigger synthesis	An increasing transformation rate may decrease the victim model accuracy
[39]	Training Phase	Backdoor	Black-Box	Passive	Dynamic	Multi-bit	Classification model	Robust to pruning, fine-pruning, Neural Cleanse, transfer learning, piracy attacks	Scaling to complex tasks without accuracy loss remains an open problem
[40]	Training Phase	Backdoor with chaotic trigger labeling	Black-Box	Passive	Static	Zero-bit	Classification model	Robust to fine-tuning, compression (pruning up to 60%), and overwriting attacks	Evaluated only on MNIST
[44]	Post training	Adversarial examples near the decision boundaries	Black-Box	Passive	Static	Zero-bit	Classification model	Robust to pruning and SVD compression; resistant to overwriting via adversarial fine-tuning	Evaluated only on MNIST; extension to other tasks (e.g., regression, segmentation)

[47]	Post training	Output watermarking via the perturbation module	Black-Box	Passive	Dynamic	Multi-bit	Image processing model	Robust to surrogate models with different architectures and loss functions (L1, L2, perceptual, adversarial)	Limited to tested combinations only; generalizability not fully verified.
[41]	Post training	Output Perturbation	Black-Box	Active	Inference Perturbation	Zero-bit	Classification model	Robust to pruning, training/inference noise, double extraction, fine-tuning (with data access limitations)	An adversary with unlimited natural data access can remove the watermark while preserving utility
[42]	Post training	Bi-Level optimization with selective parameter	Black-Box	Passive	Static	Multi-bit	Classification model	Robust to fine-tuning, pruning (up to 50%), and overwriting attacks	More keys reduce the authentication and accuracy rate
[46]	Post training	Encryption-based method	No-Box	Active	DNN Authentication	Zero-bit	Classification model	Robust to fine-tuning, pruning, and adaptive attack (attacker knows all parameters)	Fine-tuning with large data/large LR can restore accuracy
[43]	Post training	Embedding a binary watermark into images	Black-Box	Passive	Dynamic	Multi-bit	Image generation model	Robust to JPEG compression, noise, blur, and color transformations	Bit accuracy drops below 75% under very heavy perturbations
[45]	Post training	Embedding a binary watermark in the images	No-Box	Passive	Dynamic	Multi-bit	Image generation model	Robust to black-box attacks (crop, blur, JPEG, noise, quantize, super-resolution)	Not robust against white-box adaptive attacker (Reverse Pivotal Tuning) with access to ≥200 non-watermarked images
[48]	Post training	CDMA spread-spectrum	White-Box	Passive	Static	Multi-bit	Classification model	Robust to fine-tuning (RTAL/FTAL), REFIT, parameter pruning (up to 99%), shuffling, and overwriting	Vulnerable to model compression (physical neuron removal) and model distillation; requires un-shuffling procedure if parameters are shuffled

6 CHALLENGES AND SUGGESTIONS FOR THE FUTURE OUTLOOK

Although there is a lot of work in IP protection for DNN models, they are still in the infancy stages and require significant effort across various fields. In this section, we review and discuss the most important observations and address many research gaps in this field. An important starting point is to emphasize the importance of protecting models deployed in the MLaaS model. This distribution type is a promising trend, especially in cloud, commercial, and enterprise environments.

1. A lack of proactive protection (active watermarking technique), and most works are passive watermarking, which establishes ownership after the violation occurs. It's important to emphasize methods that support proactive protection, which disrupt unauthorized usage, as this is especially important when the distribution mode is MLaaS.
2. Most current works in the black-box setting are zero-bit watermarks, which rely on the trigger-based mechanism, but the black-box setting with multiple-bit watermarks can enable the requirements of real-world applications and provide expressiveness and strength to the information retrieved on ownership. There are three concrete ways in which future research could proceed: (1) to embed ownership bits in predicted probability distributions of query output using multiple API calls where different patterns of output probability correspond to different patterns of ownership information; (2) to build sets of triggers that have predictable input-output pairs, with ownership bits being embedded in each such subset; (3) to use "natural evolution strategies" (NES) or other gradient-free optimization techniques to design input patterns that yield multiple bits using measurable patterns of output prediction confidence. These methods would transform black-box watermark verification into a powerful, expressive, and legally secure ownership proof system.
3. Many watermarking techniques are based on label-based supervision, or output layers of classifiers, which are not found in generative or regression models. In the future, output perturbation of generated samples or latent-space watermarking of GANs and LLMs should be explored.
4. There is still very little support for no-box verification, as most watermarking schemes require parameter access or API queries. New approaches embed watermarks directly into model output without requiring model interaction. This direction should be further explored for social media content authenticity.

7. CONCLUSION

With the increasing innovation of DNNs in AI-driven industries, protecting DNNs is still a crucial matter, especially when MLaaS is deployed. Hence, IP protection of DNN models becomes an important issue. This SLR's proposed taxonomy has been created on the basis of a five-attribute taxonomy: the embedding phase, verification scenario, protection type, capacity, and protected content. The taxonomy is structured mainly at the embedding phase, as this is the main concept of the taxonomy.

From Table 2, many of the existing methods are white-box watermarking methods and passive, and classification models. There are still several crucial limitations to consider, such as the absence of proactive protection mechanisms and insufficient multi-bit support in a black-box setting, as well as limited generative models and image processing networks. Based on this taxonomic analysis, we list three concrete research directions: (1) new active protection strategies that can prevent the use of the content without authorization, a task different from verification of ownership; (2) in black-box situations, using multi-bit watermarking with probability distribution encoding, ensemble trigger subsets, and gradient-free optimization; (3) extending watermarking to non-classification models, such as generative adversarial networks and large language models.

ACKNOWLEDGMENT

The authors would like to thank Mustansiriyah University (WWW.uomustansiriyah.edu.iq) in Baghdad, Iraq, for its support in the present work.

REFERENCES

[1] Hartung, F., Kutter, M. (1999). Multimedia watermarking techniques. Proceedings of the IEEE, 87(7): 1079-1107. https://doi.org/10.1109/5.771066

[2] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T. (2007). Digital Watermarking and Steganography, 2nd edn Burlington. MA: Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-372585-1.X5001-3

[3] Barni, M., Bartolini, F., Piva, A. (2001). Improved wavelet-based watermarking through pixel-wise masking. IEEE Transactions on Image Processing, 10(5): 783-791. https://doi.org/10.1109/83.918570

[4] Jeon, H.J., Youn, H.C., Ko, S.M., Kim, T.H. (2021). Blockchain and AI meet in the metaverse. In Advances in the Convergence of Blockchain and Artificial Intelligence. https://doi.org/10.5772/intechopen.99114

[5] Fan, L., Chan, C.S., Yang, Q. (Eds.). (2023). Digital Watermarking for Machine Learning Model Techniques, Protocols and Applications. Intellectual Property. https://doi.org/10.1007/978-981-19-7554-7

[6] Younis, M.T., Hussien, N.M., Mohialden, Y.M., Raisian, K., Singh, P., Joshi, K. (2023). Enhancement of ChatGPT using API wrappers techniques. Al-Mustansiriyah Journal of Science, 34(2): 82-86. https://doi.org/10.23851/mjs.v34i2.1350

[7] Mousa, S.H., Shati, N.M., Sakthivadivel, N. (2024). DeepRing: Convolution neural network based on blockchain technology. Al-Mustansiriyah Journal of Science, 35(2): 61-69. https://doi.org/10.23851/mjs.v35i2.1476

[8] Taye, M.M. (2023). Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers, 12(5): 91. https://doi.org/10.3390/computers12050091

[9] Sarker, I.H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3): 160. https://doi.org/10.1007/s42979-021-00592-x

[10] Al-Tai, M.H., Nema, B.M., Al-Sherbaz, A. (2023). Deep learning for fake news detection: Literature review. Al-Mustansiriyah Journal of Science, 34(2): 70-81. https://doi.org/10.23851/mjs.v34i2.1292

[11] Sanyal, S., Addepalli, S., Babu, R.V. (2022). Towards data-free model stealing in a hard label setting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, pp. 15284-15293. https://doi.org/10.1109/CVPR52688.2022.01485

[12] Lukas, N., Jiang, E., Li, X., Kerschbaum, F. (2022). Sok: How robust is image classification deep neural network watermarking? In 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 787-804. https://doi.org/10.1109/SP46214.2022.9833693

[13] Choudhary, T., Mishra, V., Goswami, A., Sarangapani, J. (2020). A comprehensive survey on model compression and acceleration. Artificial Intelligence Review, 53(7): 5113-5155. https://doi.org/10.1007/s10462-020-09816-7

[14] Blalock, D., Gonzalez Ortiz, J.J., Frankle, J., Guttag, J. (2020). What is the state of neural network pruning? Proceedings of Machine Learning and Systems, 2: 129-146. https://doi.org/10.48550/arXiv.2003.03033

[15] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1): 43-76. https://doi.org/10.1109/JPROC.2020.3004555

[16] Chen, H., Rouhani, B.D., Fu, C., Zhao, J., Koushanfar, F. (2019). Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In Proceedings of the 2019 ACM International Conference on Multimedia Retrieval, Ottawa ON, Canada, pp. 105-113. https://doi.org/10.1145/3323873.3325042

[17] Li, P., Zhang, X., Xiao, J., Wang, J. (2024). IDEAW: Robust neural audio modeling with efficient self-supervised pre-training. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, pp. 4500-4511. https://doi.org/10.18653/v1/2024.emnlp-main.258

[18] Zheng, M., Ren, J. (2024). A high-load DNN watermarking scheme based on optimal embedding position. In 2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Wenzhou, China, pp. 382-387. https://doi.org/10.1109/ICBASE63199.2024.10762045

[19] Darvish Rouhani, B., Chen, H., Koushanfar, F. (2019). DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, pp. 485-497. https://doi.org/10.1145/3297858.3304051

[20] Chida, Y., Nagai, Y., Sakazawa, S., Satoh, S.I. (2017). Embedding watermarks into deep neural networks. In Proceedings of the 2017 ACM International Conference on Multimedia Retrieval, Bucharest, Romania, pp. 269-277. https://doi.org/10.1145/3078971.3078974

[21] Chen, H., Rouhani, B.D., Fan, X., Kilinc, O.C., Koushanfar, F. (2018). Performance comparison of contemporary DNN watermarking techniques. arXiv preprint arXiv:1811.03713. https://doi.org/10.48550/arXiv.1811.03713

[22] Pyone, A., Maung, M., Kiya, H. (2020). Training DNN model with secret key for model protection. In 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), Kobe, Japan, pp. 818-821. https://doi.org/10.1109/GCCE50665.2020.9291813

[23] Lin, N., Chen, X., Lu, H., Li, X. (2020). Chaotic weights: A novel approach to protect intellectual property of deep neural networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(7): 1327-1339. https://doi.org/10.1109/TCAD.2020.3018403

[24] Zhu, R., Zhang, X., Shi, M., Tang, Z. (2020). Secure neural network watermarking protocol against forging attack. EURASIP Journal on Image and Video Processing, 2020(1): 37. https://doi.org/10.1186/s13640-020-00527-1

[25] Maung Maung, A.P., Kiya, H. (2021). Piracy-resistant DNN watermarking by block-wise image transformation with secret key. In Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, Virtual Event, Belgium, pp. 159-164. https://doi.org/10.1145/3437880.3460398

[26] Fan, L., Ng, K.W., Chan, C.S. (2019). Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. Advances in neural information processing systems. arXiv preprint arXiv: 1909.07830v3. https://doi.org/10.48550/arXiv.1909.07830

[27] Zhao, X., Yao, Y., Wu, H., Zhang, X. (2021). Structural watermarking to deep neural networks via network channel pruning. In 2021 IEEE International Workshop on Information Forensics and Security (WIFS), Montpellier, France, pp. 1-6. https://doi.org/10.48550/arXiv.2107.08688

[28] Lv, P., Li, P., Zhang, S., Chen, K., Liang, R., Zhao, Y., Li, Y. (2021). HufuNet: Embedding the left piece as watermark and keeping the right piece for ownership verification in deep neural networks. arXiv preprint arXiv:2103.13628. https://doi.org/10.48550/arXiv.2103.13628

[29] Lim, J.H., Chan, C.S., Ng, K.W., Fan, L., Yang, Q. (2022). Protect, show, attend and tell: Empowering image captioning models with ownership protection. Pattern Recognition, 122: 108285. https://doi.org/10.1016/j.patcog.2021.108285

[30] Wang, J., Wu, H., Zhang, X., Yao, Y. (2020). Watermarking in deep neural networks via error back-propagation. Electronic Imaging, 32: 1-9. https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-022

[31] Tartaglione, E., Grangetto, M., Cavagnino, D., Botta, M. (2021). Delving in the loss landscape to embed robust watermarks into neural networks. In 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, pp. 1243-1250. https://doi.org/10.1109/ICPR48806.2021.9413062

[32] Wang, T., Kerschbaum, F. (2021). RIGA: Covert and robust white-box watermarking of deep neural networks. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, pp. 993-1004. https://doi.org/10.1145/3442381.3450000

[33] Kuribayashi, M., Tanaka, T., Suzuki, S., Yasui, T., Funabiki, N. (2021). White-box watermarking scheme for fully-connected layers in fine-tuning model. In Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, Brussels, Belgium, pp. 165-170. https://doi.org/10.1145/3437880.3460402

[34] Aramoon, O., Chen, P.Y., Qu, G. (2021). Don't forget to sign the gradients! Proceedings of Machine Learning and Systems (MLSys), 3: 194-207. https://doi.org/10.48550/arXiv.2103.03701

[35] Li, Y., Zhu, L., Jia, X., Bai, Y., Jiang, Y., Xia, S.T., Ren, K. (2025). MOVE: Effective and harmless ownership verification via embedded external features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 47(6): 4734-4751. https://doi.org/10.1109/TPAMI.2025.3546223

[36] Wu, H., Liu, G., Yao, Y., Zhang, X. (2021). Watermarking neural networks with watermarked images. IEEE Transactions on Circuits and Systems for Video Technology, 31(7): 2591-2601. https://doi.org/10.1109/TCSVT.2020.3030671

[37] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I. (2018). Protecting intellectual property of deep neural networks with watermarking. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS '18), Incheon, South Korea, pp. 159-172. https://doi.org/10.1145/3196494.3196550

[38] Namba, R., Sakuma, J. (2019). Robust watermarking of neural network with exponential weighting. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, Auckland, New Zealand, pp. 228-240. https://doi.org/10.1145/3321705.3329808

[39] Jia, H., Choquette-Choo, C.A., Chandrasekaran, V., Papernot, N. (2021). Entangled watermarks as a defense against model extraction. In 30th USENIX security symposium (USENIX Security 21), pp. 1937-1954. https://doi.org/10.48550/arXiv.2002.12200

[40] Huang, Z.J., Zhang, Y.Q., Jia, Y.R. (2021). A novel watermarking mechanism for deep learning models based on chaotic boundaries. In 2021 15th International Symposium on Medical Information and Communication Technology (ISMICT), pp. 104-109. https://doi.org/10.1109/ISMICT51748.2021.9434906

[41] Szyller, S., Atli, B.G., Marchal, S., Asokan, N. (2021). DAWN: Dynamic adversarial watermarking of neural networks. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, China, pp. 4417-4425. https://doi.org/10.1145/3474085.3475591

[42] Yang, P., Lao, Y., Li, P. (2021). Robust watermarking for deep neural networks via bi-level optimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, Canada, pp. 14841-14850. https://doi.org/10.1109/ICCV48922.2021.01457

[43] Fei, J., Xia, Z., Tondi, B., Barni, M. (2022). Supervised GAN watermarking for intellectual property protection. In 2022 IEEE International Workshop on Information Forensics and Security (WIFS), Shanghai, China, pp. 1-6. https://doi.org/10.48550/arXiv.2209.03466

[44] Le Merrer, E., Perez, P., Trédan, G. (2020). Adversarial frontier stitching for remote neural network watermarking. Neural Computing and Applications, 32(13): 9233-9244. https://doi.org/10.1007/s00521-019-04434-z

[45] Lukas, N., Kerschbaum, F. (2023). PTW: Pivotal tuning watermarking for pre-trained image generators. In 32nd

USENIX Security Symposium (USENIX Security 23), Anaheim, USA, pp. 2241-2258. https://doi.org/10.48550/arXiv.2304.07361

[46] Xue, M., Wu, Z., Wang, J., Zhang, Y., Liu, W. (2023). AdvParams: An active DNN intellectual property protection technique via adversarial perturbation based parameter encryption. IEEE Transactions on Emerging Topics in Computing, 11(3): 664-678. https://doi.org/10.1109/TETC.2022.3231012

[47] Zhang, J., Chen, D., Liao, J., Fang, H., Zhang, W., Zhou, W., Yu, N. (2020). Model watermarking for image processing networks. Proceedings of the AAAI Conference on Artificial Intelligence, 34(7): 12805-12812. https://doi.org/10.1609/aaai.v34i07.6976

[48] Pagnotta, G., Hitaj, D., Hitaj, B., Perez-Cruz, F., Mancini, L.V. (2024). Tattooed: A robust deep neural network watermarking scheme based on spread-spectrum channel coding. In 2024 Annual Computer Security Applications Conference (ACSAC), Honolulu, HI, USA, pp. 1245-1258. https://doi.org/10.1109/ACSAC63791.2024.00099

GLOSSARY

White-Box	Verification scenario with full access to model parameters
Black-Box	Verification scenario with API-only access (no internal parameters)
Gray-Box	Hybrid: white-box embedding + black-box verification
No-Box	verification is done without any direct interaction with the model itself
Active protection	Proactive mechanisms that prevent unauthorized use (e.g., performance degradation)
Passive protection	Reactive mechanisms that verify ownership after violation
DNN authentication	Active protection requiring valid key for normal model operation
Inference perturbation	Active protection adding noise to API outputs to deter extraction