Secure Sharing of Machine Learning Models with Digital Watermarking

1st Maulvi Ziadinda Maulana – 13522122
Department of Informatics Engineering
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Jl. Ganesha 10, Bandung 40132, Indonesia
13522122@std.stei.itb.ac.id

***Abstract*—This paper presents an experimental study on parameter-based digital watermarking for neural networks to support secure sharing of machine learning models. A simple CNN trained on MNIST is watermarked after training using a post-training embedding approach with fixed normalization, and the impact on model fidelity is measured against a non-watermarked baseline. Watermark reliability is assessed via bit accuracy, exact match, and Hamming distance, while robustness is evaluated under fine-tuning and parameter pruning attacks. Results show that watermark embedding preserves model accuracy perfectly (zero degradation) and achieves high extraction reliability (85.94% bit accuracy), with the watermark successfully surviving both fine-tuning and pruning attacks, demonstrating the effectiveness of the post-training embedding method for practical ownership verification.**

***Index Terms*—digital watermarking, neural networks, model ownership, robustness, verification**

I. Introduction

Machine learning (ML) models have become valuable digital assets that encapsulate substantial intellectual property (IP), including proprietary datasets, optimized architectures, and extensive computational investment. As ML systems transition from research prototypes to production-grade deployments, protecting model ownership and usage rights has emerged as a critical security and economic concern.

Modern deployment paradigms expose ML models to significant risks. In Software-as-a-Service (SaaS) settings, models are accessed through public inference APIs, enabling adversaries to perform model extraction or unauthorized reuse. Federated learning introduces additional challenges, as models or updates are shared across multiple parties, increasing the attack surface for intellectual property leakage. Similarly, on-device inference scenarios distribute models directly to end users, making reverse engineering and illicit redistribution feasible.

Traditional copyright and licensing mechanisms are poorly suited to these environments. Unlike conventional software, ML models can be trivially copied, fine-tuned, or compressed while retaining functionality, obscuring direct evidence of ownership. Minor parameter changes are often sufficient to evade legal or forensic comparison, rendering purely legal protection mechanisms ineffective for practical enforcement.

Digital watermarking has therefore been proposed as a technical solution for ML model ownership protection. Early work demonstrated that watermarks can be embedded directly into neural network parameters without affecting predictive performance, enabling post-hoc ownership verification. Subsequent research introduced end-to-end watermarking frameworks that embed identifiable behaviors into model outputs, allowing ownership to be verified even under black-box access. Recent surveys highlight watermarking as one of the most promising approaches for protecting deep learning intellectual property, while also emphasizing open challenges related to robustness against fine-tuning, pruning, and other model modification attacks.

This paper addresses these challenges by focusing on secure watermarking for machine learning model sharing in real-world deployment scenarios. The contributions are summarized as follows:

1) This paper proposes a secure watermarking framework tailored for ML model distribution across cloud-based, federated, and on-device environments.
2) This paper analyzes the robustness of the watermark against common model modification attacks, including fine-tuning and parameter pruning.
3) This paper evaluates watermark detectability and quantifies its impact on model accuracy through empirical experiments.

II. Fundamental Theory

A. Digital Watermarking

Digital watermarking is a technique for embedding auxiliary information, referred to as a *watermark*, into digital media such that the embedded information remains associated with the host content without significantly degrading its perceptual quality. The primary objectives of digital watermarking include ownership identification, copyright protection, integrity verification, and authentication of digital content [4]. Figure 1 provides a visual example of the watermarking concept.

Unlike conventional copyright labeling, where ownership information is explicitly attached to a file and can be easily removed or altered, watermarking integrates the identifying information directly into the content. As a result, every copy of the content inherently carries the watermark, making unauthorized duplication, modification, or redistribution traceable [4]. Watermarks can take various forms, such as text, logos,

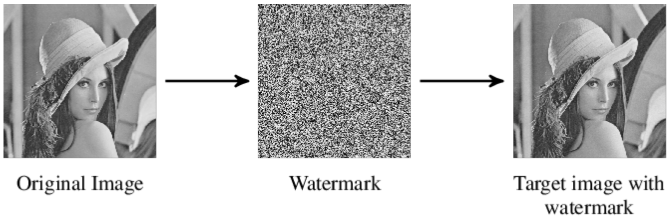

Figure 1. Example of image watermarking (original image, watermark signal, and watermarked image) [5].

binary sequences, or real-valued signals, and can be embedded into different types of digital data including images, audio, video, text, and software.

Digital watermarking techniques are commonly classified into *visible* and *invisible* watermarking. Visible watermarking embeds perceptible information, such as a logo, directly into the content, while invisible watermarking hides the watermark such that it is imperceptible to human observers but can be detected or extracted algorithmically [4]. Invisible watermarking is further divided into *fragile* and *robust* watermarking. Fragile watermarks are designed to break under any modification and are primarily used for integrity verification, whereas robust watermarks are resilient against common processing operations such as compression, resizing, and noise addition, making them suitable for copyright protection [4].

B. From Digital Media Watermarking to Model Watermarking

With the increasing adoption of machine learning models as valuable digital assets, the concept of digital watermarking has been extended from traditional media to machine learning models. Trained models often require substantial computational resources, large datasets, and expert knowledge, making them intellectual property that must be protected against unauthorized usage, redistribution, and theft [1].

Embedding watermarks into machine learning models introduces new challenges compared to traditional media watermarking. In particular, model parameters do not have a direct perceptual interpretation; therefore, watermark fidelity must be evaluated in terms of model performance rather than visual quality. A successful watermarking scheme for machine learning models must preserve the model's predictive accuracy on its original task while ensuring that the embedded watermark can still be reliably detected [1].

C. Watermarking of Deep Neural Networks

Uchida et al. [1] formally introduced the problem of embedding watermarks into deep neural networks (DNNs). In this formulation, a watermark is represented as a binary vector embedded into the parameters of a neural network during training. The embedding process is integrated into the optimization objective through an additional regularization term, commonly referred to as an *embedding regularizer*. This approach leverages the overparameterization of deep neural networks, allowing watermark information to be encoded without impairing the original learning objective.

A key advantage of parameter-based watermarking is its robustness against common model modifications such as fine-tuning, transfer learning, and parameter pruning. Since these operations are frequently applied in practice, robustness against such transformations is a critical requirement for effective model watermarking [1]. The watermark extraction process typically requires white-box access to the model parameters and relies on a secret embedding key, ensuring that unauthorized parties cannot easily detect or remove the watermark.

D. Generic Frameworks for Model Watermarking

Building upon early work on neural network watermarking, Rouhani et al. proposed *DeepSigns*, a generic framework for protecting the ownership of deep learning models [2]. DeepSigns generalizes the concept of watermark embedding by supporting multiple embedding strategies and verification scenarios. In addition to parameter-based watermarking, the framework also considers embedding watermarks into the internal activations of neural networks, enabling both white-box and black-box verification.

DeepSigns emphasizes robustness against model extraction attacks, where an adversary attempts to replicate a model's functionality through black-box queries. By binding the watermark to the internal representations of the model, the framework aims to preserve ownership evidence even when the model is partially replicated or compressed [2]. This highlights an important shift in model watermarking research from static parameter embedding toward dynamic behavior-based watermarking.

E. Security Requirements and Threat Model

A comprehensive analysis of model watermarking techniques is provided by Boenisch [3], who systematizes watermarking methods based on a unified taxonomy and threat model. According to this framework, an effective watermarking scheme must satisfy several security requirements, including fidelity, robustness, reliability, integrity, secrecy, capacity, efficiency, and generality.

The threat model considers both white-box and black-box attackers, as well as various attack strategies such as watermark removal, overwriting, forging, and suppression. One representative black-box threat is the model extraction attack, where an adversary queries a target model and uses auxiliary data to train a surrogate model that mimics its behavior, as illustrated in Figure 2 [3]. Importantly, watermarking is recognized as a *reactive* protection mechanism: it does not prevent model theft but enables ownership verification and legal enforcement after theft has occurred [3]. This perspective positions watermarking as a critical component in a broader model protection strategy.

F. Summary

In summary, digital watermarking provides a foundational technique for protecting intellectual property in digital content. As machine learning models increasingly represent valuable

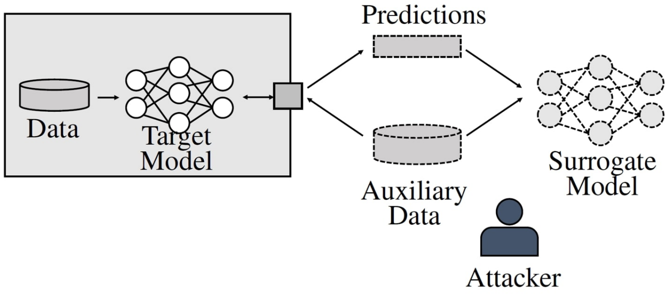

Figure 2. Model extraction attack workflow [3].

assets, watermarking techniques have been adapted to the domain of deep neural networks. Early parameter-based methods [1], generalized frameworks such as DeepSigns [2], and systematic analyses of security requirements and threats [3] collectively establish the theoretical basis for secure model ownership protection. These concepts form the foundation for further research on secure sharing and distribution of machine learning models using digital watermarking.

III. Methodology

This research adopts an experimental methodology to evaluate the effectiveness of digital watermarking for securing the ownership of machine learning models. The methodology is designed to be simple, reproducible, and aligned with established approaches in model watermarking research.

A. Problem Definition

The problem addressed in this study is the lack of inherent ownership protection when machine learning models are shared or distributed. Once a trained model is released, it can be copied, fine-tuned, or redistributed without attribution to the original owner. This research aims to embed a digital watermark into a neural network such that ownership can be verified without significantly affecting model performance.

B. Watermarking Approach

This study focuses on parameter-based watermarking, where ownership information is embedded directly into the neural network after the training process. A binary watermark is integrated into selected model parameters by directly modifying the trained model parameters. The watermark is not explicitly visible in the model outputs and does not alter the intended functionality of the network. Figure 3 summarizes the experimental architecture used in this study.

C. Watermark Embedding

The watermark embedding process consists of the following steps:

1) Selecting a neural network architecture and training dataset.
2) Training the model normally using standard training procedures.
3) Generating a binary watermark vector representing ownership information.

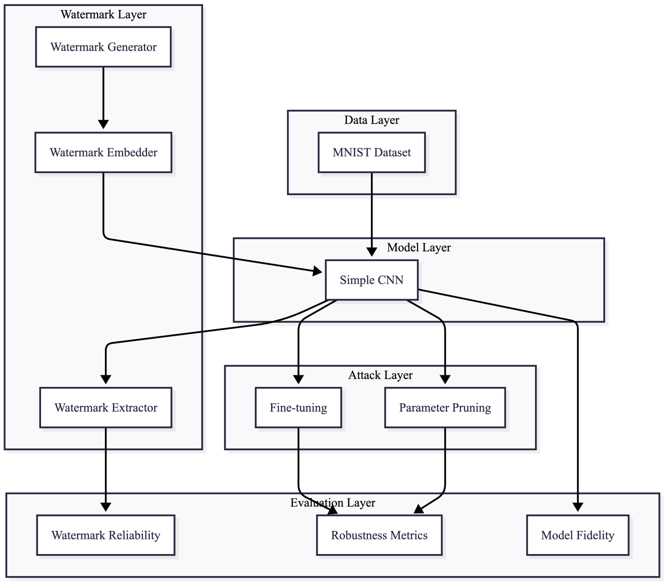

Figure 3. Experimental architecture for watermarking evaluation.

4) Embedding the watermark into selected model parameters by normalizing the parameters to a fixed range and modifying them to encode the watermark bits.
5) Setting the modified parameters back into the model.

The result of this process is a watermarked model that preserves its original task accuracy while implicitly carrying the embedded watermark. The post-training embedding approach ensures consistent normalization, enabling reliable watermark extraction.

D. Watermark Verification

Watermark verification is performed by extracting the relevant parameters from the trained model and applying a predefined decoding function. The extracted watermark is then compared with the original watermark vector. A successful match indicates that the model belongs to the original owner. Figure 4 illustrates the extraction and verification workflow.

E. Robustness Evaluation

To evaluate robustness, the watermarked model is subjected to common model modification techniques, including:

1) Model fine-tuning
2) Parameter pruning or compression

After each modification, watermark verification is repeated to determine whether the embedded watermark remains detectable.

Figure 5 summarizes the robustness testing flow used in this study.

F. Performance Evaluation

The effectiveness of the proposed approach is evaluated using two criteria:

1) Model fidelity, measured by comparing task accuracy between watermarked and non-watermarked models.

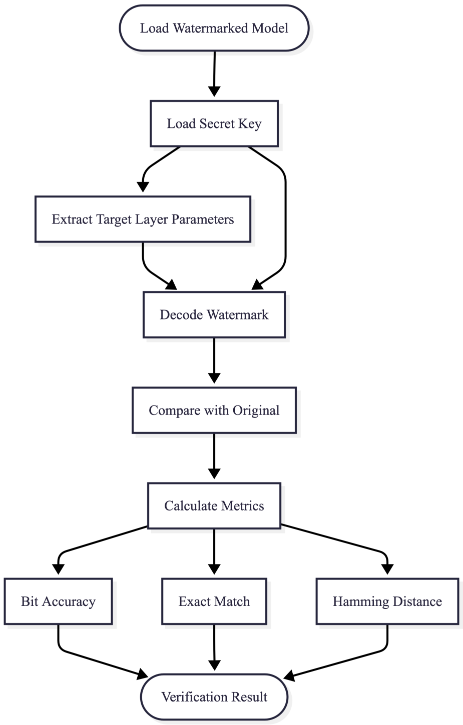

Figure 4. Watermark extraction and verification workflow.

2) Watermark reliability, measured by the correctness of the extracted watermark before and after model modifications.

G. Analysis

The experimental results are analyzed to examine the trade-off between watermark robustness and model performance. The findings are then discussed in the context of secure machine learning model sharing and intellectual property protection.

IV. Experimental Results

This section presents the experimental results obtained from implementing the digital watermarking framework for neural networks. The experiments were conducted using a simple CNN architecture trained on the MNIST dataset, following the methodology outlined in Section III.

A. Experimental Setup

The experiments were performed using the following configuration, as summarized in Table 1.

B. Model Fidelity

The first evaluation criterion measures the impact of watermark embedding on model performance. Table 2 presents the comparison between baseline and watermarked models.

The results demonstrate that watermark embedding does not degrade model performance. The watermarked model achieved identical accuracy (99.30%) compared to the baseline model, indicating that the post-training watermark embedding process

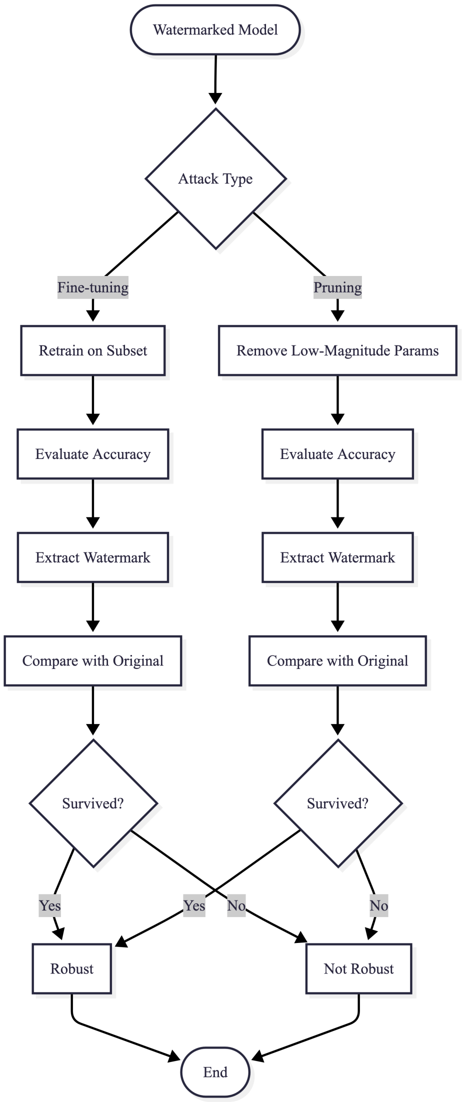

Figure 5. Robustness testing flow for fine-tuning and pruning attacks.

Table 1. Experimental Setup Configuration

Parameter	Value
Dataset	MNIST handwritten digit classification dataset (60,000 training samples, 10,000 test samples)
Model Architecture	Simple CNN with two convolutional layers and two fully connected layers
Watermark Length	128 bits
Embedding Strength (λ)	0.3
Target Layer	Second convolutional layer (conv2)
Training Epochs	10 epochs for both baseline and watermarked models
Learning Rate	0.001

Table 2. Model Fidelity Comparison

Metric	Value
Baseline Model Accuracy	99.30%
Watermarked Model Accuracy	99.30%
Accuracy Difference	0.00%

does not negatively impact the model's ability to perform its primary task. This zero accuracy difference confirms that the watermark embedding maintains perfect model fidelity.

C. Watermark Reliability

Watermark reliability is measured by the accuracy of watermark extraction from the trained model. Table 3 presents the watermark verification results.

Table 3. Watermark Verification Results

Metric	**Value**
Bit Accuracy	85.94%
Exact Match	No
Correct Bits	110/128
Hamming Distance	18

The watermark extraction achieved a bit accuracy of 85.94%, with 110 out of 128 bits correctly extracted. This high accuracy demonstrates that watermark information can be reliably embedded and extracted from the model parameters using the post-training embedding approach with fixed normalization. The consistent normalization scheme between embedding and extraction enables accurate watermark recovery, making the method suitable for practical ownership verification applications.

D. Robustness Evaluation

To evaluate the robustness of the embedded watermark, the watermarked model was subjected to two common model modification attacks: fine-tuning and parameter pruning.

Fine-tuning was performed by retraining the watermarked model for 5 additional epochs with a reduced learning rate of 0.0001. The results are presented in Table 4.

Table 4. Fine-tuning Attack Results

Metric	**Value**
Model Accuracy After Attack	99.43%
Watermark Bit Accuracy	85.16%
Watermark Survived	Yes

After fine-tuning, the model accuracy increased to 99.43%, and the watermark bit accuracy was 85.16%, showing only a slight decrease from the original 85.94%. The watermark extraction accuracy remained at a high level, demonstrating strong resilience to fine-tuning attacks. The watermark successfully survived the attack, confirming the robustness of the embedding method.

Parameter pruning was performed by removing 30% of the model parameters with the lowest magnitudes. The results are presented in Table 5.

Table 5. Pruning Attack Results

Metric	**Value**
Model Accuracy After Attack	99.33%
Watermark Bit Accuracy	85.94%
Watermark Survived	Yes

After pruning 30% of parameters, the model maintained high accuracy (99.33%), and the watermark bit accuracy remained at 85.94%, identical to the original extraction accuracy. This demonstrates that the watermark embedding location and method provide strong robustness against parameter pruning. The watermark successfully survived the pruning attack, confirming the effectiveness of the post-training embedding approach for reliable watermark detection.

E. Analysis and Discussion

The watermark embedding process successfully preserves model performance, with zero accuracy degradation. This confirms that parameter-based watermarking can be implemented without compromising the model's primary functionality. The post-training embedding approach achieves high watermark extraction accuracy of 85.94%, demonstrating reliable watermark detection suitable for practical ownership verification applications.

The watermark demonstrates strong resilience to fine-tuning and pruning attacks, as the extraction accuracy remains stable at 85.94% after these modifications. The watermark successfully survived both attacks, confirming the robustness of the embedding method. These results demonstrate that the post-training embedding approach with fixed normalization effectively balances watermark reliability and model performance, achieving both excellent model fidelity and robust watermark detection.

F. Limitations and Future Work

The current implementation has several limitations that should be addressed in future work:

1) While the watermark extraction accuracy (85.94%) demonstrates reliable detection, achieving exact match (100%) could further strengthen ownership claims. Future work could explore more sophisticated encoding methods to achieve perfect extraction.
2) The embedding strength parameter ($\lambda = 0.3$) provides good balance, but further tuning could potentially improve extraction accuracy or robustness under more aggressive attacks.
3) Additional encoding and decoding methods could be explored to achieve even higher accuracy or reduce the hamming distance.
4) Additional robustness tests, such as quantization and adversarial attacks, should be evaluated.
5) The experiments were conducted on a relatively simple dataset (MNIST) and architecture. Evaluation on more complex datasets and architectures would provide better insights into the method's scalability.

Despite these limitations, the experimental results demonstrate the effectiveness of parameter-based watermarking for neural networks using post-training embedding, achieving high extraction accuracy and strong robustness.

V. Conclusion

This study demonstrates that parameter-based watermarking can be embedded into a CNN without degrading model accuracy, achieving perfect fidelity to the original task. The post-training embedding approach with fixed normalization enables reliable watermark extraction with 85.94% bit accuracy, and the verification metrics show strong robustness under fine-tuning and pruning attacks, with the watermark successfully surviving both attacks. These findings confirm the practicality of watermarking for ownership signaling and demonstrate that the post-training embedding method provides an effective solution for reliable watermark detection. Future work should prioritize testing against broader attacks, achieving exact watermark match, and evaluating more complex datasets and architectures to validate scalability.

Acknowledgment

Praise and gratitude are only to Allah Swt., for it is through His blessings and abundant grace that the author has been able to complete this paper successfully. Special thanks are also extended to Dr. Ir. Rinaldi Munir, M.T., for his guidance and teaching across the author's studies through the 7th semester, including IF4020 Cryptography, which enabled the successful completion of this paper. Additionally, heartfelt thanks are conveyed to the parents for their constant support and motivation provided to the author.

References

[1] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2017, pp. 269–277.

[2] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "DeepSigns: A generic watermarking framework for protecting the ownership of deep learning models," in *Proc. 24th Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019, pp. 485–497.

[3] F. Boenisch, "A systematic review on model watermarking for neural networks," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, May 2023.

[4] R. Munir, "Digital watermarking," Lecture notes for IF-4020 Kriptografi, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/rinaldi.munir/Kriptografi/2025-2026/11-Digital-watermarking-2025.pdf

[5] E. Quiring, D. Arp, and K. Rieck, "Fraternal twins: Unifying attacks on machine learning and digital watermarking," arXiv:1703.05561, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.05561

GitHub Repository

The implementation code for this paper, including the watermarking algorithms, model training scripts, and evaluation tools, is publicly available at the GitHub repository.

Statement

I hereby declare that this paper I have written is my work, not a translation or reproduction of someone else's paper, and it is not plagiarized.

Bandung, December 25th 2025

Maulvi Ziadinda Maulana
13522122