Vol. 14 (2026): Proceedings of the 2nd International Conference on Artificial Intelligence, Modern Engineering and Environmental Sustainability (IC-AIMEES 2026)
E-ISSN: 3105-0522 P-ISSN：3105-0514
DOI: https://doi.org/10.70267/ ic-aimees.20260243251

Research Progress and Prospects of Fragile Watermarking for Model Integrity Protection

Zixin Zhou*

School of Computer Science, Wuhan University, Wuhan, 430072, China

**Corresponding author: Zixin Zhou.*

Abstract

Traditional deep learning models represented by convolutional neural networks face severe security threats in open environments, including model tampering and backdoor injection. As an active defense technique, fragile model watermarking aims to produce a sensitive response to any unauthorized modification of a model, thereby providing a "digital seal" for model integrity protection. This paper systematically reviews fragile watermarking techniques for model integrity protection and categorizes three mainstream paradigms: black-box verification based on sensitive samples, white-box/gray-box authentication based on parameter hashing and reversible embedding, and self-embedding and recovery mechanisms. Through comparative analysis, these three paradigms exhibit distinct advantages. Black-box watermarking offers convenient deployment and is well suited for model-as-a-service scenarios centered on classification tasks; parameter-level watermarking provides fine-grained authentication with cryptographic strength; self-embedding mechanisms extend the protection boundary from models to input content, offering a proactive solution for countering deepfakes. Finally, this paper discusses technical challenges and future development trends, providing references for building a trustworthy AI ecosystem.

Keywords

fragile watermarking, model integrity, artificial intelligence security, black-box verification

1. Introduction

Traditional deep learning models represented by convolutional neural networks face severe security threats in open environments, including model tampering and backdoor injection. To establish a trustworthy model supply chain, fragile model watermarking has emerged as a key active defense technique. Unlike robust watermarking used for ownership verification, fragile watermarking is designed to be highly sensitive to any unauthorized modification of a model, thereby providing a "digital seal" for model integrity protection.

This paper aims to systematically review the technical principles of fragile watermarking. The first section introduces commonly used datasets and evaluation metrics for fragile watermarking techniques. The second section presents black-box fragile watermarking based on sensitive samples. The third section discusses fragile watermarking based on parameter hashing. The fourth section introduces fragile watermarking based on self-embedding and recovery mechanisms. The fifth section comparatively analyzes the performance of these three

categories of fragile model watermarking techniques. The sixth section concludes the paper and provides future prospects.

2. Datasets and Evaluation Metrics

To evaluate the effectiveness of the aforementioned techniques, existing studies have widely adopted multiple public datasets for experimental validation, as summarized in Table 1, which provide a reproducible benchmarking foundation for research in this field. To systematically quantify the performance of fragile watermarking methods, existing studies have established evaluation frameworks from multiple dimensions, including watermark carriers, verification methods, tampering detection granularity, and the impact on model performance, together with corresponding quantitative metrics. Watermark carriers include decision boundaries and parameter spaces. According to the verifier's access privileges to the model, verification methods can be categorized into black-box verification, white-box verification, and gray-box verification. Tampering detection granularity refers to the smallest unit at which a watermarking mechanism can localize model tampering, which can be classified into behavioral-level, layer/block-level, and parameter-level granularity. The impact on model performance is quantified by accuracy degradation, and an ideal fragile watermark should incur as little accuracy loss as possible (typically <1%). For generative models, metrics such as FID or PSNR are used to assess the impact of watermarking on generation quality. For fragile watermarking schemes equipped with self-recovery capability, recovery success rate is adopted as an evaluation metric. Together, these evaluation metrics constitute a multidimensional standard for measuring the effectiveness and practicality of fragile model watermarking techniques, providing a quantitative basis for the subsequent comparative analysis of the three categories of techniques in this review.

Table 1: Common Training Datasets for Fragile Watermarking

Dataset	Training Set / Images	Test Set / Images	Total / Images	Task Domain
MNIST [1-5]	50,000	10,000	60,000	Handwritten digit recognition
Fashion-MNIST [1,6]	60,000	10,000	70,000	Clothing image classification
CIFAR-10 / CIFAR-100 [1-8]	50,000	10,000	60,000	General object classification
SVHN [1-4]	73,257	26,032	99,289	Street View house number recognition
ImageNet [6,8,9]	~1,200,000	50,000	~1,250,000	Large-scale visual recognition
Tiny ImageNet [7,9,10]	100,000	10,000	110,000	Image classification (200 classes)
GTSRB [1,2,4]	39,209	12,630	51,839	Traffic sign recognition
IIIT5K-Words [11]	2,000	3,000	5,000	Scene text recognition / OCR
VGGFace2 [6]	No fixed split	No fixed split	3.31M	Face recognition
CelebA [6]	162,770	19,962	202,599	Facial attributes / generation
MS-COCO [6,10]	118,287	40,670	163,957	Image transformation / generation
DIV2K [10]	800	100	900	Image super-resolution
LSUN-Bedroom [6]	3,033,042	300	3,033,342	Scene image generation
Purchase-100 [3,5,8]	180,000	20,000	200,000	User behavior prediction
Books Corpus [9]	11,038/books	-	11,038/books	Text corpus

3. Black-Box Fragile Watermarking Based on Sensitive Samples

Black-box fragile watermarking achieves remote integrity verification without accessing internal model parameters by generating sensitive samples bound to the model decision boundary and comparing whether the outputs of a suspicious model remain consistent with the original records. Existing studies cover several representative technical routes, including domain-adaptive trigger generation, latent-space global search, lossless authentication based on external generators, boundary fluctuation paired detection, and semi-fragile watermarking.

The fragile watermarking scheme for OCR tasks proposed in Ref. [11] represents a pioneering effort in deeply adapting black-box fragile watermarking to vertical domains. Its core components include trigger set generation, watermark embedding, and integrity verification. As shown in Figure 1, in Stage I, original text images are used as inputs, and high-quality samples are selected through entropy screening, $H(x) = -\sum p(c \mid x) \log f_0\, p(c \mid x) \geqslant \tau$, while perturbations are injected into font morphology, character spacing, and stroke

thickness to obtain $x' = x + \delta$, thereby generating a trigger set T that is visually and semantically coherent while located near the decision boundary. The trigger samples are visually indistinguishable from the original samples and remain stably positioned close to the model decision boundary. As shown in Figure 2, the trigger set is jointly encoded with a preset label γ_{target} , followed by two-stage fine-tuning. In the first stage, mixed training is performed with $L_1 = L_{original} + \alpha$. In the second stage, reinforced training is conducted with $L_2 = L_{trigger} + \beta R_{var}$, which drives trigger samples to stably cluster around the decision boundary. As illustrated in Figure 3, in Stage III, the verifier submits trigger samples and compares whether the model outputs are consistent with the preset labels. Consistency indicates model integrity, whereas inconsistency implies tampering. This work is the first to extend the black-box fragile watermarking framework to OCR tasks.

Figure 1: Trigger Set Generation in OCR Stage I

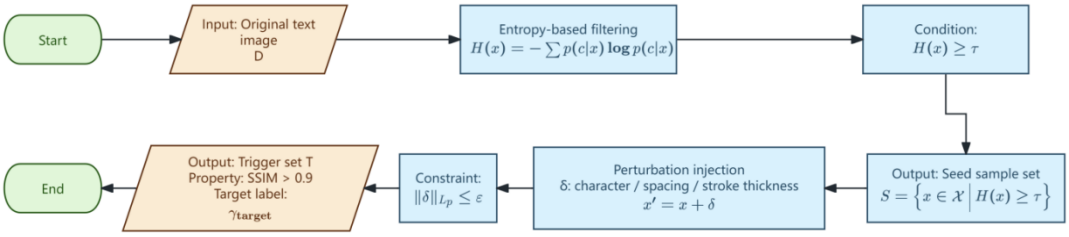


Figure 2: Watermark Embedding in OCR Stage II

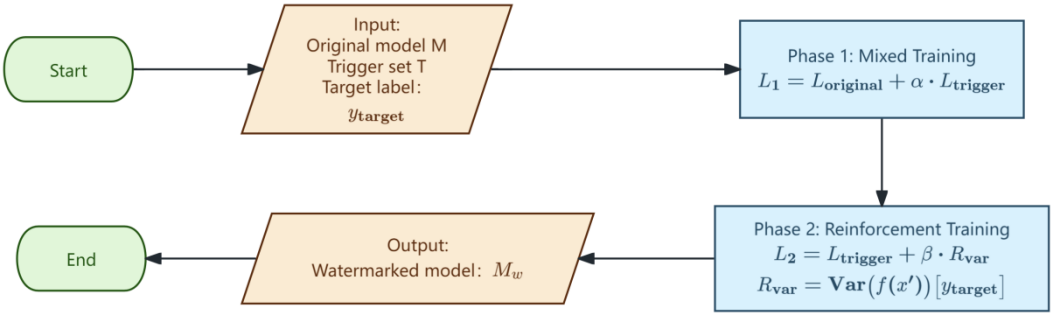
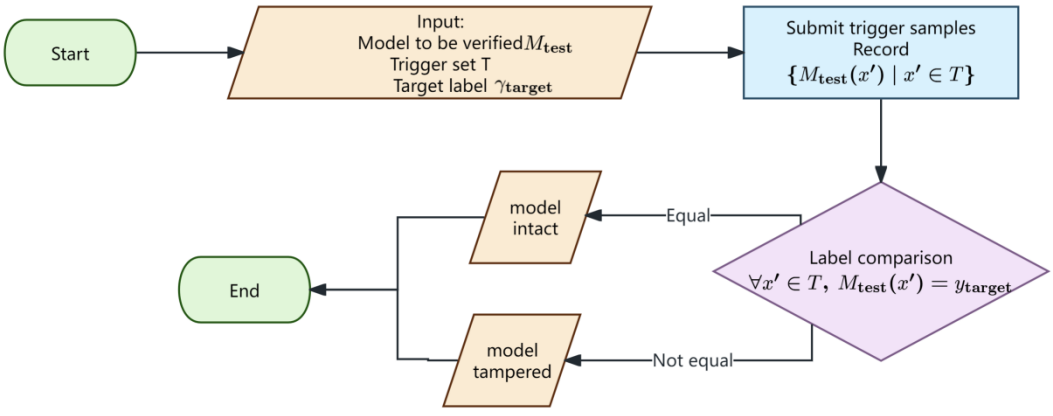


Figure 3: Integrity Verification in OCR Stage III



As summarized in Table 2, Ref. [1] employs Bayesian optimization and variational autoencoders (VAE) to search for sensitive samples in latent space, achieving detection coverage above 92%, although with relatively low efficiency. Ref. [2] proposes a semi-fragile watermarking method that leverages adversarial samples to distinguish malicious operations from benign modifications, achieving a detection rate above 90% and a false positive rate below 5%, demonstrating stronger practicality. With the development of generative AI, fragile watermarking has been extended to generative models. Ref. [6] proposes a watermarking framework for GANs

based on key-trigger samples and two-stage fine-tuning, enabling black-box authentication of model tampering with detection rates approaching 100%. Ref. [10] designs a "trigger input–expected output" mapping scheme for image transformation networks, further expanding the application boundary of fragile watermarking.

Table 2: Summary of Black-Box Fragile Watermarking Techniques

Method	Core Innovation	Detection Sensitivity	Impact on Model Fidelity	Main Advantages	Potential Limitations and Challenges
[1]	Bayesian optimization + VAE global search	High (coverage >92%)	None (sample generation only)	High-quality samples, approximates global optimum	Sequential optimization leads to low generation efficiency
[2]	Semi-fragile watermarking	Adjustable (>90% detection rate)	Low	Distinguishes malicious and benign operations, strong practicality	Defining and generalizing the boundary between benign and malicious operations remains challenging
[6]	Key-triggering + two-stage overfitting fine-tuning	Extremely high (close to 100%)	Extremely low (FID and related metrics remain nearly unchanged)	First fragile watermarking scheme for GANs; black-box authentication; highly sensitive to fine-tuning and pruning	Stealthiness of the trigger mechanism; large-scale generalization remains to be verified
[10]	Trigger input–expected output mapping	To be verified (effective in white-box settings)	None (input modification only)	First watermarking scheme for image transformation networks; extends to generative models	Only validated in white-box scenarios; black-box effectiveness remains to be verified

4. Fragile Watermarking Based on Parameter Hashing

Parameter-level fragile watermarking embeds hash digests of model parameters into the weights themselves, enabling tampering localization and reversible recovery. Existing studies cover several representative technical routes, including parameter block chain authentication, covert embedding in frequency-domain coefficients, importance-aware self-healing recovery, and lossless restoration via reversible information hiding.

Among fragile watermarking techniques for model integrity protection, the decision-iterative behavioral hashing method proposed in Ref. [9] is a representative approach due to its distinctive black-box verification capability. Its complete workflow accurately corresponds to the schematic of a "key-driven query process," consisting of watermark generation and integrity verification. As shown in Figure 4, the model owner uses a secret key K as a random seed to generate and optimize a query sequence X_K that is highly sensitive to parameter perturbations. X_K is fed into the original model M, and the resulting Top-1 prediction label sequence C_K is recorded, from which a hash value H_K is computed. The pair (K, H_K) is then securely stored. As shown in Figure 5, during the verification stage, the verifier reconstructs X_K using the same key K and inputs it into the model under verification M', obtaining a prediction label sequence $C_K{}'$ and computing a corresponding hash value $H_K{}'$. By comparing H_K with $H_K{}'$, model integrity can be determined. This method relies exclusively on Top-1 output labels throughout the process and therefore belongs to decision-level black-box access. It requires no modification of model parameters and incurs zero accuracy loss. Experiments show that the generated samples achieve a PSNR of 55 dB on TinyImageNet and remain sensitive to parameter modifications at the scale of 1×10^{-4}. Potential challenges lie in securely maintaining the key-driven query set and strengthening robustness against adversarial targeted fine-tuning.

Figure 4: Watermark Generation in Stage I of Decision-Level Behavioral Hashing

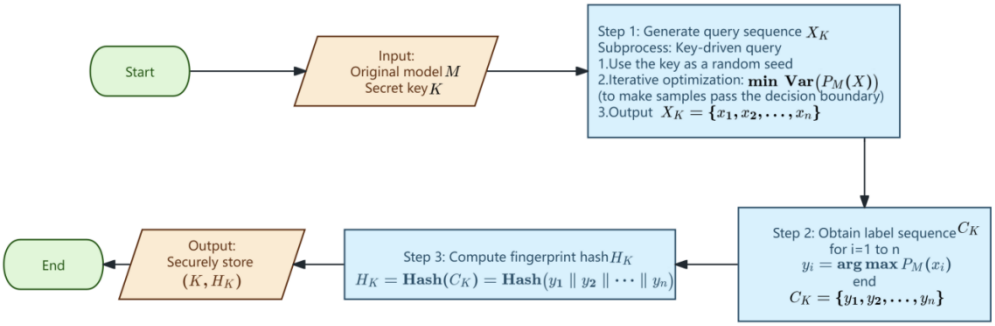
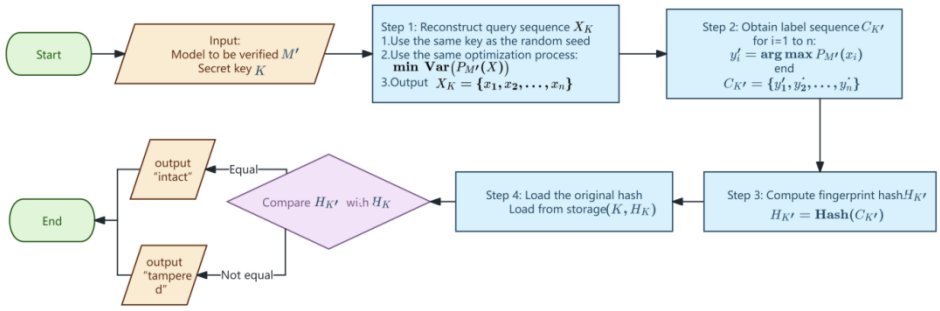


Figure 5: Flowchart of Integrity Verification in Stage II of Decision-Level Behavioral Hashing



As summarized in Table 3, extending the above black-box idea, the parameter partitioning and chained hash embedding scheme in Ref. [3] shifts toward white-box fine-grained tampering localization. This method partitions weights into blocks and computes SHA-256 hashes, which are chain-embedded into the least significant bits (LSBs) of adjacent blocks. During verification, hashes are compared to detect tampering, achieving detection rates above 95% while precisely localizing compromised blocks. The scheme is simple in design and causes less than 0.5% accuracy degradation, but it requires full access to model parameters and is vulnerable to reordering attacks. Ref. [4] proposes a hash embedding scheme based on discrete wavelet transform in the frequency domain, embedding authentication digests into high-frequency coefficients of weights. It incurs less than 0.3% accuracy loss and achieves detection rates above 90% for 1‰ parameter tampering. The scheme provides strong concealment, though embedding capacity is limited and white-box access is required. To counter forgery and overwriting attacks, Ref. [8] proposes NeuralMark, which constructs a hash watermark filter. It employs a hash function to generate irreversible binary watermarks as the basis for parameter selection, where the avalanche effect prevents reverse engineering and resists forgery attacks, while multi-round filtering drives parameter overlap toward zero to defend against overwriting. In addition, average pooling is introduced to enhance robustness against fine-tuning and pruning. Experiments covering 13 architectures demonstrate a 100% detection rate with nearly no performance degradation.

Table 3: Summary of Fragile Watermarking Techniques Based on Parameter Hashing

Method	Core Innovation	Detection Sensitivity	Impact on Model Fidelity	Main Advantages	Potential Limitations and Challenges
[3]	Parameter partitioning + chained hash LSB embedding	High (>95%)	Slight (<0.5%)	Fine-grained block authentication, simple implementation, low overhead	Vulnerable to parameter reordering attacks; purely white-box
[4]	DWT + frequency-domain hash embedding	High (>90%)	Slight (<0.3%)	Strong concealment, sensitive to perturbations	Limited capacity; dependent on white-box access
[8]	Hash watermark filter + average pooling	100%	Nearly lossless	Defends against forgery and overwriting attacks; robust to fine-tuning and pruning; architecture-generalizable	White-box dependency; requires preset filter rounds

5. Fragile Watermarking Based on Self-Embedding and Recovery Mechanisms

This category of mechanisms aims to elevate the protection capability of fragile watermarking from "passive detection" to "active recovery." The core idea is to use intrinsic feature information of the model itself, such as weight summaries and architectural fingerprints, as watermarks and self-embed them into the model.

The DNN self-embedding fragile watermarking scheme proposed in Ref. [5] is the first to realize tampering detection, localization, and parameter recovery for neural network models through four core steps: parameter partitioning, dual-type data generation, reference-sharing mechanisms, and LSB embedding. As shown in Figure 6, the scheme first divides model parameters into fixed-size parameter blocks PB_i. For each block, authentication data and recovery data are generated as $AD_i = H(\text{MSB}(PB_i) \| K)$ and $RD_i = F(\text{MSB}(PB_i))$. Subsequently, using a reference-sharing mechanism, watermark data $W_i = AD_i \| RD_i$ are permuted and cross-embedded into the least significant bits (LSBs) of other parameter blocks, ensuring that even if one parameter block is completely destroyed, information can still be extracted from other blocks for recovery. As shown in Figure 7, during verification, authentication data extracted from the LSBs are compared with the hash values of the current MSBs, allowing precise localization of tampered blocks: $T_{set} = \{j \mid AD_j' \neq H(\text{MSB}(PB_j') \| K)\}$. As illustrated in Figure 8, associated recovery data are then extracted from untampered blocks, and the MSB content of tampered blocks is reconstructed through the decoding function G, thereby achieving parameter recovery. Experiments show that this scheme causes less than 1% impact on model performance, while the recovery success rate exceeds 95% when the tampering ratio is below 50%. Its effectiveness has been validated across multiple networks such as LeNet and ResNet, making it the first approach to extend the protection capability of fragile watermarking from "passive detection" to "active recovery."

Figure 6: Watermark Generation and Recovery in Stage I of Self-Embedded Parameter Recovery

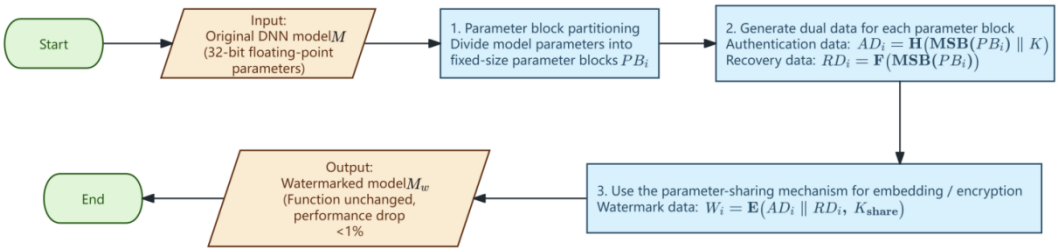


Figure 7: Tampering Detection and Localization in Stage II of Self-Embedded Parameter Recovery

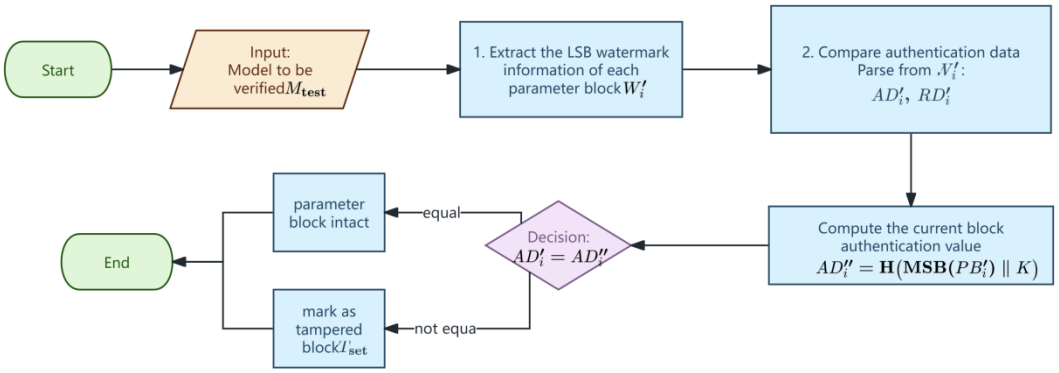
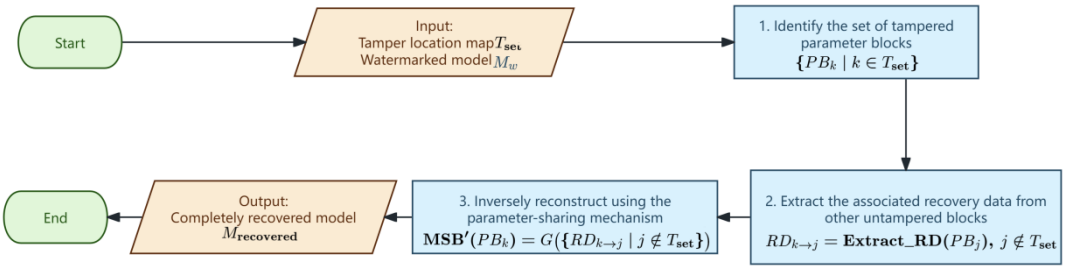


Figure 8: Parameter Recovery in Stage III of Self-Embedded Parameter Recovery



As summarized in Table 4, Ref. [12] focuses on non-intrusive detection and proposes a triplet-aware hash zero-watermarking framework. By correlating model structure, trigger inputs, and activation patterns, it enables layer-level tampering localization, achieving accuracy above 85% under black-box conditions with zero performance degradation. However, its localization granularity is coarse, and it remains sensitive to benign operations. Subsequently, building upon Ref. [5], Ref. [7] proposes a hierarchical recovery method based on self-embedded watermarking. By introducing an entropy-based parameter importance ranking mechanism and a differentiated partitioning strategy, it achieves fine-grained tampering recovery for CNN models. In this method, authentication bits and reference bits are embedded into the least significant bits (LSBs) of parameter blocks. Important parameters adopt fine-grained partitioning to improve recovery precision, while less important parameters use coarse-grained partitioning to reduce computational overhead. Damaged parameters are recovered using untampered blocks. Experiments validate the high detection accuracy and superior recovery performance of this method, achieving a balanced optimization between recovery efficiency and precision on top of Zhao et al.'s framework.

Table 4: Summary of Fragile Watermarking Techniques Based on Self-Embedding and Recovery Mechanisms

Method	Core Objective	Technical Characteristics	Main Advantages	Core Challenges
[12]	Non-intrusive tampering localization	Triplet-aware hashing, zero-watermarking	Zero performance impact; provides localization capability	Coarse localization granularity; sensitive to benign evolution
[7]	Hierarchical parameter recovery + tampering detection and localization	Entropy-based importance ranking, differentiated partitioning, authentication-bit + reference-bit LSB embedding	High recovery precision; controllable overhead; inherits Zhao framework	Mechanism is complex; requires balancing granularity and effectiveness

6. Comparative Performance Analysis of Three Categories of Fragile Model Watermarking Techniques

As summarized in Table 5, this paper compares the three categories of fragile watermarking paradigms from multiple dimensions, including watermark carrier, verification method, and detection granularity. Black-box fragile watermarking based on sensitive samples uses the model decision boundary as the watermark carrier and relies on pure black-box verification to detect changes in overall model behavior. It offers extremely high deployment convenience and is well suited to model-as-a-service (MaaS) scenarios. Reversible watermarking based on parameter hashing uses the parameter space as the watermark carrier and typically adopts gray-box verification, enabling parameter-level perturbation awareness and providing authentication with cryptographic strength. Some variants further support tampering localization and recovery. Self-embedding and recovery mechanisms use intrinsic model parameter features as watermark carriers. Through the embedded design of authentication bits and recovery bits, they elevate protection capability from passive detection to active recovery. These three paradigms do not represent successive generational replacements, but rather differentiated technical routes designed for distinct security objectives. Together, they constitute a systematic framework that extends from model integrity protection toward content authenticity assurance. This framework also points toward future directions such as fragile watermarking with both localization and parameter recovery capabilities, as well as hierarchical semi-fragile watermarking capable of distinguishing malicious tampering from benign operations.

Table 5: Performance Comparison of Three Categories of Fragile Model Watermarking Techniques

Comparison Dimension	Black-Box Fragile Watermarking Based on Sensitive Samples	Fragile Watermarking Based on Parameter Hashing	Self-Embedding and Recovery Watermarking
Watermark Carrier	Model decision boundary	Model parameter space	Intrinsic model parameter features
Verification Method	Pure black-box verification	Gray-box verification	White-box / gray-box verification
Tampering Detection Granularity	Overall model behavior changes	Parameter-level perturbation awareness	Precise parameter-block-level localization
Impact on Model Performance	May cause slight performance degradation	Theoretically lossless (or reversible)	Performance impact <1%
Core Advantages	Extremely convenient deployment; strong MaaS prospects	Cryptographic authentication; sensitive to parameter tampering; high security; some variants support localization and recovery	First to support parameter recovery; passive detection → active recovery; supports localization and repair
Core Challenges	Insensitive to internal tampering; unable to localize	Requires parameter access; limited application scenarios	Mechanism complexity; recovery rate declines under high tampering ratios

7. Conclusion

This paper systematically reviews three major fragile watermarking paradigms for model integrity protection. Black-box watermarking based on sensitive samples offers convenient deployment and is suitable for model-as-a-service scenarios. Reversible watermarking based on parameter hashing provides fine-grained authentication with cryptographic strength. Self-embedding and recovery mechanisms extend the protection boundary from models to data content, offering new perspectives for deepfake detection. Together, these three paradigms constitute a technical framework that extends from model integrity protection toward content authenticity assurance. Future research should focus on several directions, including novel fragile watermarking schemes for emerging architectures such as Transformer-based models and diffusion models; self-healing watermarking capable of both tampering localization and recovery; semi-fragile watermarking that distinguishes malicious tampering from benign operations; robust fragile watermarking resistant to attacks such as fine-tuning and pruning; and the construction of standardized evaluation frameworks to facilitate the transition of these techniques from theoretical research to practical deployment.

References

[1] Kuttichira, D. P., Gupta, S., Nguyen, D., Rana, S., & Venkatesh, S. (2022). Verification of integrity of deployed deep learning models using Bayesian optimization. *Knowledge-Based Systems, 241*, 108238. https://doi.org/10.1016/j.knosys.2022.108238

[2] Yuan, Z., Zhang, X., Wang, Z., & Yin, Z. (2024). Semi-fragile neural network watermarking based on adversarial examples. *IEEE Transactions on Emerging Topics in Computational Intelligence, 8*(4), 2775–2790. https://doi.org/10.1109/TETCI.2024.3370576

[3] Botta, M., Cavagnino, D., & Esposito, R. (2021). NeuNAC: A novel fragile watermarking algorithm for integrity protection of neural networks. *Information Sciences, 576*, 228–241. https://doi.org/10.1016/j.ins.2021.07.004

[4] Abuadbba, A., Kim, H., & Nepal, S. (2021). DeepiSign: Invisible fragile watermark to protect the integrity and authenticity of CNN. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 952–959). ACM. https://doi.org/10.1145/3412841.3441981

[5] Zhao, G., Qin, C., Yao, H., & Han, Y. (2022). DNN self-embedding watermarking: Towards tampering detection and parameter recovery for deep neural network. *Pattern Recognition Letters, 164*, 16–22. https://doi.org/10.1016/j.patrec.2022.10.011

[6] Yuan, Z., Li, L., Wang, Z., & Zhang, X. (2025). Integrity protection of generative adversarial networks using fragile watermarking. *ACM Transactions on Multimedia Computing, Communications, and Applications, 21*(12), 1–21. https://doi.org/10.1145/3724332

[7] Huang, Y., & Zhang, H. (2025). Hierarchical recovery of convolutional neural networks via self-embedding watermarking. In *International Conference on Information and Communications Security* (pp. 424–441). Springer Nature Singapore.

[8] Yao, Y., Song, J., & Jin, J. (2026). Hashed watermark as a filter: A unified defense against forging and overwriting attacks in neural network watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence, 40*(42), 35994–36002.

[9] Yin, Z., Yin, H., Su, H., Zhang, X., & Gao, Z. (2023). Decision-based iterative fragile watermarking for model integrity verification. *arXiv*. https://arxiv.org/abs/2305.09684

[10] Robinette, P. K., Nguyen, T. D., Sasaki, S., & Johnson, T. T. (2025). Trigger-based fragile model watermarking for image transformation networks. In *European Symposium on Research in Computer Security* (pp. 346–365). Springer Nature Switzerland.

[11] Yin, Y., Yin, H., Yin, Z., Lyu, W., & Wei, S. (2023). High-quality triggers based fragile watermarking for optical character recognition model. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 468–475). IEEE. https://doi.org/10.1109/APSIPAASC58517.2023.10317376

[12] Xiong, C., Feng, G., Li, X., Zhang, X., & Qin, C. (2022). Neural network model protection with piracy identification and tampering localization capability. In *Proceedings of the 30th ACM International Conference on Multimedia* (pp. 2881–2889). ACM. https://doi.org/10.1145/3503161.3548206

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access