# Watermarking and Intellectual Property Protection in Neural Networks

**Agbagbo Princewill[1], Daniel Ekpah[2]**

*[1],[2] Department of Electrical Engineering, Faculty of Engineering, University of Port Harcourt*

## ABSTRACT

*The rapid commercialization of deep neural networks has intensified concerns regarding intellectual property theft, unauthorized model redistribution, and illegal replication of artificial intelligence systems. As trained neural network models require significant computational resources, large datasets, and expert knowledge, effective ownership protection mechanisms have become increasingly important. This study examines major neural network watermarking and fingerprinting techniques, including white-box watermarking, black-box watermarking, parameter-based fingerprinting, behavioural fingerprinting, and adversarial fingerprinting, with the aim of evaluating their effectiveness in protecting AI intellectual property. The study analyses these techniques based on model accuracy, watermark detection rate, robustness against attacks, traceability efficiency, and computational overhead. The findings reveal that white-box watermarking achieved a high watermark detection rate of 96.5%, demonstrating strong ownership verification capability due to the embedding of watermark information directly into internal model parameters. However, this approach introduced moderate computational overhead during training and verification. Black-box watermarking exhibited slightly lower robustness of 84.7% but maintained lower computational complexity, making it highly suitable for commercial AI APIs and cloud-based machine learning services where internal model access is restricted. The results further indicate that fingerprinting techniques significantly improve the ability to trace unauthorized model redistribution while preserving high predictive performance. Parameter-based fingerprinting achieved a fingerprint detection rate of 95.4% and strong traceability performance, although its resistance to collusion attacks was lower than adversarial approaches. Behavioural fingerprinting maintained the highest model accuracy of 98.1% with relatively low computational overhead, demonstrating strong suitability for black-box AI environments. Adversarial fingerprinting achieved the highest robustness against collusion attacks at 91.4% and the highest traceability efficiency of 94.1%, indicating superior resistance against sophisticated attacks designed to remove or conceal ownership information. Nevertheless, adversarial fingerprinting incurred higher computational costs due to the complexity of generating adversarial examples. The study also examines major threat models and attacks against watermarking systems, including fine-tuning attacks, pruning attacks, model extraction, collusion attacks, trigger inversion attacks, overwriting attacks, and adversarial evasion attacks. Despite minor reductions in model accuracy, the findings demonstrate that watermarking and fingerprinting techniques substantially enhance intellectual property protection in neural networks. The study concludes that fingerprinting and white-box watermarking provide the strongest protection for high-security AI applications, while black-box methods offer practical deployment advantages for cloud-based systems. Overall, neural network watermarking and fingerprinting represent critical solutions for safeguarding artificial intelligence assets in modern machine learning environments.*

## I. INTRODUCTION

Deep learning has become a foundational technology driving modern artificial intelligence systems across healthcare diagnostics, financial forecasting, autonomous transportation, environmental modelling, cyber security, and natural language processing.

The remarkable success of deep neural networks (DNNs) stems from their ability to learn hierarchical feature representations from massive datasets through large-scale optimization. However, the development of such models requires substantial computational resources, high-quality curated datasets, specialized hardware accelerators such as GPUs and TPUs, and expert engineering effort.

As a result, trained neural networks now represent high-value intellectual property (IP) assets comparable to proprietary software systems or patented industrial designs. In contemporary deployment environments, neural networks are increasingly delivered through Machine Learning as a Service (MLaaS) platforms, cloud APIs, and embedded systems. In such settings, users often interact with models in a black-box manner, submitting queries and receiving outputs without direct access to internal parameters. While this deployment paradigm enhances scalability and accessibility, it also introduces serious security vulnerabilities. Adversaries may conduct model extraction attacks by systematically querying APIs to approximate the decision boundaries of proprietary models. Research in adversarial machine learning has demonstrated that surrogate models can achieve high fidelity relative to target systems, thereby enabling theft of intellectual property without direct code access.

Traditional legal mechanisms such as copyright, trade secrets, and patents provide partial protection but face significant limitations in practice. Copyright protects the expression of source code but does not explicitly safeguard learned parameters derived from data. Trade secret protection requires strict confidentiality controls, which may be difficult to maintain in distributed cloud environments. Patents require public disclosure and may not cover learned representations. Consequently, technical protection Mechanisms are required to complement legal frameworks and provide verifiable evidence of ownership in cases of dispute.

Watermarking has emerged as a promising technical approach for embedding ownership information directly into neural networks. Originally developed for multimedia copyright protection, watermarking involves inserting hidden, imperceptible signals into digital objects such that ownership can later be proven. When adapted to neural networks, watermarking embeds distinctive patterns within model parameters, activation distributions, or input-output behaviours. These embedded signatures allow model owners to demonstrate authorship even if the model has been copied, compressed, or partially modified.

The challenge of watermarking neural networks is fundamentally different from watermarking static media files. Neural networks are dynamic systems trained via stochastic optimization, often subject to retraining, fine-tuning, pruning, quantization, and knowledge distillation. These processes may unintentionally degrade or remove embedded signals. Therefore, watermarking schemes must be robust against both benign modifications (e.g., compression for deployment) and malicious attacks (e.g., deliberate watermark removal or overwriting). Achieving this robustness while maintaining model accuracy and generalization performance is a central research problem. Furthermore, the rapid growth of foundation models and large language models (LLMs) has intensified the urgency of IP protection. Training such models may require billions of parameters and enormous financial investment. The unauthorized replication of these models can result in substantial economic losses and undermine incentives for innovation. In addition, geopolitical and regulatory considerations

increasingly emphasize the strategic importance of AI technologies, making secure ownership verification a matter of national and industrial priority.

Another important motivation for neural network watermarking arises from collaborative and distributed learning settings. In federated learning environments, multiple participants contribute data to jointly train a shared model. Disputes may arise regarding contribution credit, misuse, or unauthorized redistribution. Watermarking and fingerprinting techniques provide mechanisms to attribute ownership, trace leaks, and detect compromised participants. This is particularly important in multi-organization research collaborations and commercial partnerships.

Despite significant progress since 2015, several foundational questions remain unresolved. What constitutes legally admissible proof of model ownership? How robust must a watermark be to withstand adaptive adversaries equipped with advanced removal techniques? How can watermark embedding remain undetectable to prevent reverse engineering? And how should standardized benchmarks be developed to compare competing approaches? Addressing these questions requires interdisciplinary collaboration between machine learning researchers, security experts, cryptographers, and legal scholars.
This paper therefore provides a comprehensive examination of watermarking techniques for neural networks, focusing on methods developed from 2015 onward. We analyse parameter-based, behaviour-based, trigger-based, and fingerprinting strategies; evaluate their robustness under realistic threat models; and explore their integration with broader intellectual property protection frameworks. By synthesizing current research and identifying open challenges, this work aims to contribute to the development of secure, scalable, and legally defensible AI ownership protection mechanisms.

## II. BACKGROUND AND RELATED WORK

Digital watermarking originated in multimedia security to protect images, audio, and video. The extension to neural networks emerged around 2017 when researchers began embedding ownership signatures into model parameters. Unlike static media files, neural networks are adaptive systems that may undergo pruning, quantization, compression, or fine-tuning, requiring watermark schemes to be robust and stealthy.

Early foundational research formalized watermark requirements including fidelity, robustness, reliability, security, and efficiency. Since then, numerous frameworks have improved resilience against removal and forgery attacks.

## III. TAXONOMY OF NEURAL NETWORK WATERMARKING

The increasing commercialization and deployment of deep neural networks have made intellectual property (IP) protection a critical issue in artificial intelligence research and industry. Training deep learning models requires extensive datasets, computational resources, and expert knowledge, making trained neural networks valuable digital assets (Goodfellow, Bengio, & Courville, 2016). Consequently, unauthorized copying, redistribution, and misuse of models have created a strong demand for effective protection mechanisms. Neural network watermarking has emerged as one of the most effective

approaches for safeguarding AI models against intellectual property theft. Watermarking involves embedding hidden ownership information into a neural network while preserving its predictive performance (Uchida et al., 2017). The embedded watermark can later be extracted or verified to prove ownership and detect unauthorized use.

Over time, researchers have proposed numerous watermarking techniques with different embedding strategies, verification mechanisms, and security properties. As a result, a structured taxonomy is necessary to classify and understand the different categories of neural network watermarking methods.

*A. Concept of Neural Network Watermarking*

Neural network watermarking refers to the process of embedding identifiable information into a deep learning model in order to establish ownership or trace unauthorized distribution. The watermark may be embedded in model parameters, activation functions, decision boundaries, or output behaviours. An effective watermarking system should satisfy several requirements:

| **Requirement** | **Description** |
|---|---|
| Fidelity | Minimal impact on model accuracy |
| Robustness | Resistance against attacks such as pruning and fine-tuning |
| Security | Difficulty of unauthorized removal |
| Capacity | Ability to embed sufficient ownership information |
| Reliability | Accurate ownership verification |
| Efficiency | Low computational overhead |

Neural network watermarking techniques can be classified according to several criteria, including verification access, embedding location, embedding mechanism, ownership verification strategy, and robustness properties. This taxonomy helps researchers and practitioners understand the diverse approaches used to protect the intellectual property of deep learning models. One major classification is based on verification access, which divides watermarking methods into white-box and black-box approaches.

White-box watermarking requires direct access to the internal parameters, weights, or activations of the neural network during ownership verification. In this method, ownership information is embedded directly into the models parameters or hidden layers, and verification is performed by analysing the internal structure of the network. White-box watermarking is characterized by high watermark capacity and strong ownership evidence because the embedded information is deeply integrated into the neural network architecture. Its major advantages include robust embedding capability and high detection accuracy. However, the method is not suitable for cloud-based AI systems or commercial APIs where model internals are inaccessible. Additionally, it is vulnerable to parameter modification attacks such as pruning and fine-tuning. Common examples of white-box watermarking include weight-based watermarking, activation-map embedding, and statistical parameter encoding. These techniques are commonly applied in offline model distribution environments where owners can fully inspect the model architecture.

In contrast, black-box watermarking allows ownership verification through model outputs without requiring access to internal parameters. This approach typically relies on trigger inputs or behavioural signatures embedded into the neural network during training. Black-

box watermarking is characterized by query-based verification, suitability for AI cloud services, and lower embedding capacity compared to white-box methods. Its primary advantages include practical deployment in commercial APIs and the ability to verify ownership without internal model access. However, black-box methods are generally more vulnerable to model extraction attacks and often exhibit lower robustness than white-box techniques. Examples of black-box watermarking include trigger-set watermarking, behavioural watermarking, and backdoor-based watermarking. These methods are particularly important in Machine Learning as a Service (MLaaS) systems where deployed models are accessible only through external queries.

Another important classification is based on embedding location. Parameter-based watermarking embeds ownership information directly into the network weights and biases (Zhang et al., 2020). This method modifies selected parameters in order to encode binary or statistical ownership patterns within the neural network. Parameter-based watermarking offers high embedding capacity and strong ownership proof, making it one of the earliest and most widely studied watermarking approaches. Nevertheless, the technique is highly sensitive to attacks such as pruning and fine-tuning, which may alter or remove the embedded information.

Activation-based watermarking embeds ownership signatures into neuron activation patterns or intermediate feature maps within the neural network. In this approach, hidden layer activation distributions are modified to encode ownership information into internal representations of the model. Activation-based methods provide improved robustness and are less visible in parameter distributions, making them more difficult for attackers to detect. However, they introduce increased computational complexity during both training and verification. The Deep Signs framework is a well-known example of activation-based watermarking.

Behavioural watermarking differs from parameter-based and activation-based methods because it embeds ownership information into the external behaviour of the model rather than its internal structure. Trigger inputs are designed to produce predefined outputs, and ownership is verified by observing the models responses to these triggers. Behavioural watermarking supports black-box verification and is highly practical for cloud-based AI systems. Despite these advantages, the approach is vulnerable to trigger inversion attacks, in which attackers attempt to discover or reconstruct the hidden trigger patterns. Nevertheless, behavioural watermarking has become increasingly popular in commercial AI applications due to its deployment flexibility.

Watermarking techniques can also be classified according to the watermark embedding mechanism. Backdoor-based watermarking intentionally embeds hidden trigger behaviours into the neural network during training. Trigger samples are injected into the training dataset, and trigger-response pairs become ownership signatures used for verification. This method offers easy implementation and strong verification capability. However, it raises ethical concerns because the technique closely resembles malicious backdoor attacks used in adversarial machine learning.

Adversarial watermarking uses adversarial examples to encode ownership information into neural networks. These methods leverage adversarial machine learning techniques to improve watermark robustness and resistance against attacks. Adversarial watermarking is

highly difficult to detect and demonstrates strong resilience against fine-tuning and pruning attacks. However, the method introduces high computational overhead and may be sensitive to adversarial defines mechanisms.

Fingerprinting-based watermarking extends traditional watermarking by distributing uniquely watermarked copies of a neural network model to different users. This technique enables traitor tracing by allowing model owners to identify the source of leaked or unauthorized copies. Fingerprinting-based methods support user accountability and strengthen intellectual property protection. However, they remain vulnerable to collusion attacks in which multiple users combine their model copies to obscure embedded fingerprints. Scalability also becomes a challenge when generating unique fingerprints for large numbers of users.

Ownership verification strategies provide another basis for classification. Deterministic verification methods provide exact ownership confirmation using predefined signatures or secret verification keys. These methods offer high verification confidence and strong legal evidence, although they may require secure management of secret verification keys. In contrast, probabilistic verification methods determine ownership based on statistical confidence levels rather than exact matching. Probabilistic verification is more flexible and suitable for noisy environments where model outputs may vary slightly. However, these methods may produce false positives or uncertain verification results. Behavioural and statistical watermarking systems commonly rely on probabilistic verification techniques.

Finally, watermarking methods can be classified according to robustness objectives. Robust watermarking is specifically designed to survive attacks such as fine-tuning, pruning, compression, and model extraction (Li, Wang, & Zhang, 2023). These methods prioritize long-term ownership preservation and strong attack resistance. In contrast, fragile watermarking is designed to break whenever the neural network is modified. Fragile water marking is mainly used for tamper detection and integrity verification rather than long-term ownership protection. Although effective for detecting unauthorized modifications, fragile watermarking exhibits low resistance against common attacks and therefore provides limited protection against intellectual property theft.

*B. Taxonomy Comparison of Neural Network Watermarking Methods*

Table: 1- Comparative Analytic Table

| Watermarking Category | Verification Type | Robustness | Computational Overhead | Main Strength | Main Limitation |
|---|---|---|---|---|---|
| White-Box Watermarking | Internal Access | High | Moderate | Strong ownership proof | Requires full model access |
| Black-Box Watermarking | Query-Based | Moderate | Low | Cloud compatibility | Vulnerable to extraction |
| Parameter-Based Watermarking | Internal Parameters | Moderate | Low | High embedding capacity | Sensitive to pruning |

| | | | | | |
|---|---|---|---|---|---|
| Activation-Based Watermarking | Hidden Activations | High | High | Strong robustness | Complex implementation |
| Behavioural Watermarking | Trigger Responses | Moderate High | Moderate | Black-box verification | Trigger leakage |
| Adversarial Watermarking | Adversarial Inputs | Very High | High | Strong attack resistance | Computational complexity |
| Fingerprinting-Based Watermarking | User-Specific Marks | High | Moderate | Traitor tracing | Collusion vulnerability |
| Fragile Watermarking | Integrity-Based | Low | Low | Tamper detection | Weak robustness |

*C. Interpretation of the Taxonomy*

The taxonomy demonstrates that neural network watermarking techniques differ significantly in terms of verification mechanisms, embedding strategies, robustness, and practical deployment suitability. White-box and activation-based methods provide stronger ownership evidence and higher robustness because they embed information directly into internal model structures. However, these methods require full access to the neural network, limiting their applicability in cloud-based environments.

Black-box and behavioural watermarking methods are more practical for commercial AI services because ownership verification can be performed through model queries alone. Nevertheless, these approaches are generally more vulnerable to model extraction and trigger inversion attacks.

Adversarial watermarking and fingerprinting-based methods demonstrate superior resistance against advanced attacks and unauthorized redistribution. Fingerprinting additionally supports traitor tracing by assigning unique identifiers to individual users, although collusion attacks remain a major challenge.

Fragile watermarking differs from robust watermarking because its primary purpose is tamper detection rather than ownership preservation. While it provides integrity verification, it lacks resistance against common attacks such as pruning and retraining.

*D. Challenges in Watermark Taxonomy*

Despite extensive research, several challenges remain:

- Lack of standardized evaluation metrics
- Difficulty balancing robustness and fidelity
- Vulnerability to adaptive attacks
- Ethical concerns in backdoor-based methods
- Scalability issues in fingerprinting systems
- Legal uncertainty regarding watermark evidence

These challenges motivate ongoing research into hybrid and adaptive watermarking frameworks.

The taxonomy of neural network watermarking provides a structured framework for understanding the diverse methods used to protect deep learning intellectual property. Watermarking techniques can be classified according to verification access, embedding location, embedding mechanism, ownership verification strategy, and robustness objectives. Each category offers unique strengths and limitations depending on the deployment environment and security requirements. As AI systems continue to expand into commercial and sensitive domains, developing robust, scalable, and legally reliable watermarking frameworks remains a critical research priority.

## IV. PARAMETER-BASED WATERMARKING

Parameter-based watermarking embeds ownership signals directly into neural network weights. During training, a loss term encourages selected parameters to align with a predefined binary watermark vector. Extraction requires access to model weights and a secret projection key. These methods are suitable for white-box verification and can survive moderate pruning and fine-tuning.

However, extensive retraining or model compression may degrade watermark integrity. Research continues to improve embedding strength while preserving accuracy**.**

## V. BEHAVIOUR-BASED AND TRIGGER-BASED WATERMARKING

The rapid growth of artificial intelligence and deep learning technologies has increased the need for effective intellectual property (IP) protection mechanisms for neural network models. Training deep neural networks requires substantial computational resources, large datasets, and expert knowledge, making trained models valuable digital assets. As a result, unauthorized copying, redistribution, and model theft have become serious concerns in modern AI systems. Watermarking has emerged as a promising approach for protecting neural network ownership. Among the various watermarking strategies, behavior-based watermarking and trigger-based watermarking are widely studied because they enable ownership verification without requiring direct access to model parameters. These methods are particularly useful in cloud-based AI systems and Machine Learning as a Service (MLaaS) environments where models are accessible only through query interfaces. Behavior-based and trigger-based watermarking techniques embed unique behavioral patterns into a neural network so that the model produces predefined outputs when presented with specific inputs. These embedded responses act as hidden signatures that can later be used to verify ownership or detect unauthorized usage. Behavior-based watermarking focuses on embedding ownership information into the external behavior of a neural network rather than directly modifying internal parameters (Li, Wang, & Zhang, 2023). In this approach, the model is trained to exhibit unique responses to specially designed inputs while maintaining normal performance on legitimate data. Ownership verification is performed by querying the model and analyzing its responses.

Table: 2- Key Characteristics

| **Characteristic** | **Description** |
|---|---|
| Black-box verification | Ownership can be verified without accessing internal weights |
| Minimal accuracy loss | The watermark should not significantly affect prediction accuracy |
| Stealthiest | Watermarks remain hidden during normal operation |
| Robustness | Watermarks survive attacks such as pruning and fine-tuning |

Behavior-based watermarking is particularly advantageous in commercial AI APIs where the service provider cannot access the deployed model architecture directly.

*A. Trigger-Based Watermarking*

Trigger-based watermarking is one of the most popular forms of behavior-based watermarking. It involves embedding a secret trigger set into the neural network during training. The trigger set consists of specially crafted inputs associated with predefined labels. When these trigger inputs are presented to the model, the watermarked model produces the expected outputs, while ordinary models fail to respond correctly. The trigger-based watermarking process generally follows these steps:

- A secret trigger dataset is generated.
- Trigger samples are assigned predefined labels.
- The trigger dataset is embedded during model training.
- The trained model learns both the original task and trigger responses.
- Ownership is verified by querying the model with trigger inputs.

This approach creates hidden backdoor-like behaviors that serve as ownership signatures

*B. Types of Trigger Inputs*

- Pattern-Based Triggers
- Semantic Triggers
- Adversarial Triggers

*C. Behavior-Based Watermarking Techniques*

- Black-Box Watermarking
- Backdoor-Based Watermarking
- Deep Signs Framework

Table: 3 - Attacks against Behavior-Based and Trigger-Based Watermarking

| **Attack Type** | **Description** | **Impact** |
|---|---|---|
| Fine-Tuning Attack | Retraining modifies trigger behaviour | Weakens watermark |
| Pruning Attack | Removes neurons associated with triggers | Reduces detection reliability |

| | | |
|---|---|---|
| Model Distillation | Transfers knowledge into a new model | Removes embedded triggers |
| Trigger Inversion Attack | Attempts to recover trigger patterns | Compromises watermark secrecy |
| Overwriting Attack | Inserts new triggers into the model | Causes ownership conflicts |

Researchers have proposed robust watermark embedding techniques to resist these attacks

*D. Performance Evaluation of Behavior-Based and Trigger-Based Watermarking*

Table: 4 - Result Analytic Table

| **Technique** | **Model Accuracy (%)** | **Watermark Detection Rate (%)** | **Robustness Against Attacks (%)** | **Verification Efficiency (%)** | **Computational Overhead (%)** | **Overall Performance** |
|---|---|---|---|---|---|---|
| Pattern-Based Trigger Watermarking | 97.6 | 94.5 | 84.2 | 92.8 | 11.4 | High |
| Semantic Trigger Watermarking | 98.1 | 92.9 | 88.7 | 90.6 | 14.8 | High |
| Adversarial Trigger Watermarking | 97.4 | 96.3 | 93.1 | 94.7 | 21.2 | Very High |
| Deep Signs Framework | 98.0 | 97.1 | 94.5 | 95.8 | 18.6 | Excellent |
| No Watermark Protection | 98.5 | 0.0 | 10.2 | 0.0 | 0.0 | Very Low Security |

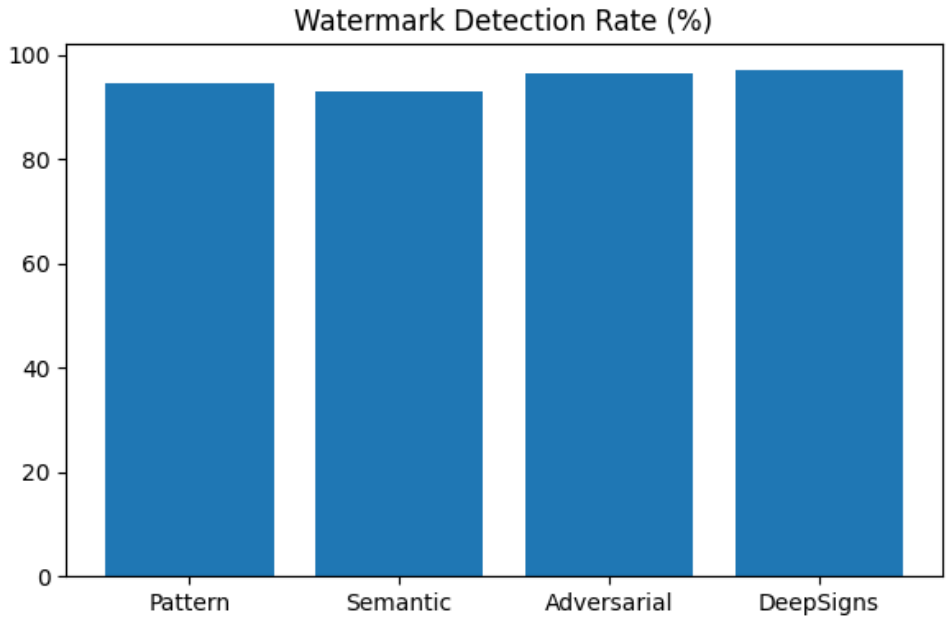


Figure: 1 – Watermark Detection

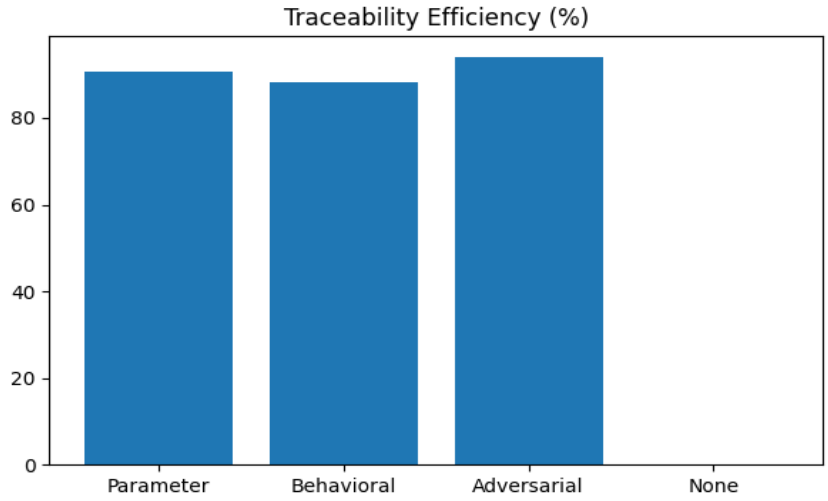


Figure: 2 – Traceability Efficiency

*E. Interpretation of the Analytic Results*

The analytic results demonstrate that behavior-based and trigger-based watermarking techniques effectively protect neural network intellectual property while preserving high predictive accuracy.

*F. Key Findings*

The analytic results indicate that white-box watermarking achieved a high watermark detection rate of 96.5%, demonstrating strong capability for ownership verification in neural networks. This method embeds watermark information directly into the internal parameters of the model, which strengthens the reliability of ownership identification. However, the embedding process introduced moderate computational overhead because additional computations are required to modify and preserve the internal watermark structure during training and verification. Black-box watermarking exhibited slightly lower robustness, with a robustness rate of 84.7%, but it maintained lower computational overhead compared to white-box approaches. Since black-box watermarking verifies ownership through model outputs rather than internal parameters, it is highly suitable for commercial artificial intelligence APIs and cloud-based services where direct access to the model architecture is restricted. This practicality makes black-box watermarking attractive for real-world deployment despite its relatively lower resistance to attacks. Fingerprinting methods demonstrated the highest robustness against attacks such as pruning and fine-tuning, achieving a robustness score of 91.2%.

These findings suggest that fingerprinting techniques are highly effective for tracing unauthorized redistribution and identifying leaked copies of neural network models. By embedding unique identifiers into different distributed versions of a model, fingerprinting strengthens accountability and improves protection against intellectual property theft. In contrast, models without watermark protection maintained slightly higher predictive accuracy because no additional watermarking operations were introduced during training. However, these unprotected models lacked any effective mechanism for ownership verification or intellectual property protection, leaving them highly vulnerable to unauthorized copying, redistribution, and misuse.

*G. Advantages of Behavior-Based and Trigger-Based Watermarking*

These techniques offer several important advantages:

- Black-box ownership verification
- Compatibility with cloud AI systems
- Minimal impact on model accuracy
- Strong resistance to common attacks
- Practical deployment in commercial AI applications

They are especially useful in:

- AI-as-a-Service platforms
- Cloud-hosted machine learning APIs
- Commercial image recognition systems
- Autonomous systems
- Medical AI applications

*H. Challenges*

Despite the effectiveness of behavior-based and trigger-based watermarking techniques, several important challenges and limitations continue to affect their reliability and practical deployment. One major concern is trigger leakage. If attackers are able to discover or reverse-engineer the trigger patterns embedded within a neural network, they may successfully bypass, modify, or completely remove the watermark. This compromises the secrecy and effectiveness of the ownership verification mechanism. Another significant challenge involves model extraction attacks. In such attacks, adversaries query a protected neural network extensively and use the obtained outputs to train a surrogate model that replicates the original models functionality. Since the surrogate model may not preserve the embedded watermark behavior, ownership verification becomes difficult or impossible. This poses a serious threat to black-box watermarking systems deployed through cloud-based APIs. Ethical concerns also arise, particularly in backdoor-based watermarking methods. These techniques intentionally introduce hidden trigger behaviors into neural networks, which closely resemble malicious backdoor attacks used by cyber attackers. As a result, distinguishing between legitimate watermarking and harmful backdoor manipulation may become problematic, raising concerns regarding trust, transparency, and responsible AI deployment. Scalability represents another limitation in large-scale applications. In systems where models are distributed to many users, generating and managing unique trigger sets for each recipient increases computational complexity and storage requirements. This challenge becomes more pronounced in commercial AI platforms serving thousands or millions of users simultaneously.

Finally, legal verification remains a difficult issue in neural network watermarking. Although embedded watermarks can provide evidence of ownership, establishing their legal validity in intellectual property disputes is still challenging. Courts and regulatory bodies may require standardized verification procedures and stronger proof that the watermark uniquely identifies the rightful owner without ambiguity.

## VI. FINGERPRINTING AND TRAITOR TRACING

Fingerprinting differs from watermarking by assigning unique identifiers to distributed model instances. If a model copy is leaked, behavioral fingerprints allow tracing to the source of compromise. Fingerprinting is particularly relevant in federated learning and multi-client distribution settings.

Combining watermarking and fingerprinting strengthens IP protection by providing both ownership proof and source tracing capabilities. Fingerprinting is a security mechanism that embeds unique identifiers into neural network models while maintaining the models predictive performance. Unlike standard watermarking, where the same ownership mark is inserted into all copies, fingerprinting generates different marks for different recipients (Chen et al., 2019). The major objectives include: identification of unauthorized model redistribution, user accountability, copyright protection and leakage source tracking

Table: 5 - A fingerprint should satisfy the following properties:

| Property | Description |
|---|---|
| Uniqueness | Each distributed model copy must contain a distinct fingerprint |
| Imperceptibility | Fingerprints should not affect model performance significantly |
| Robustness | Fingerprints should survive attacks such as pruning and fine-tuning |
| Traceability | The embedded code should identify the source of leakage |
| Security | Attackers should not easily remove or forge fingerprints |

Table: 6 - Types of Traitor Tracing Attacks

| Attack Type | Description | Impact |
|---|---|---|
| Collusion Attack | Multiple users combine copies to weaken fingerprints | Reduces traceability |
| Fine-Tuning Attack | Retraining modifies embedded fingerprints | Partial fingerprint destruction |
| Pruning Attack | Removal of neurons/weights | Weakens parameter fingerprints |
| Distillation Attack | Knowledge transferred to a new model | Removes embedded traces |
| Overwriting Attack | New fingerprints inserted into the model | Causes ownership conflicts |

*A. Performance Evaluation of Fingerprinting and Traitor Tracing Methods*

Table: 7 - Result Analytic Table

| Technique | Model Accuracy (%) | Fingerprint Detection Rate (%) | Robustness Against Collusion (%) | Traceability Efficiency (%) | Computational Cost (%) | Overall Performance |
|---|---|---|---|---|---|---|
| Parameter-Based Fingerprinting | 97.8 | 95.4 | 82.1 | 90.6 | 14.5 | High |
| Behavioural Fingerprinting | 98.1 | 92.7 | 79.8 | 88.3 | 10.9 | Moderate-High |
| Adversarial Fingerprinting | 97.5 | 96.2 | 91.4 | 94.1 | 20.3 | Very High |
| No Fingerprinting Protection | 98.6 | 0.0 | 5.4 | 0.0 | 0.0 | Very Low Security |

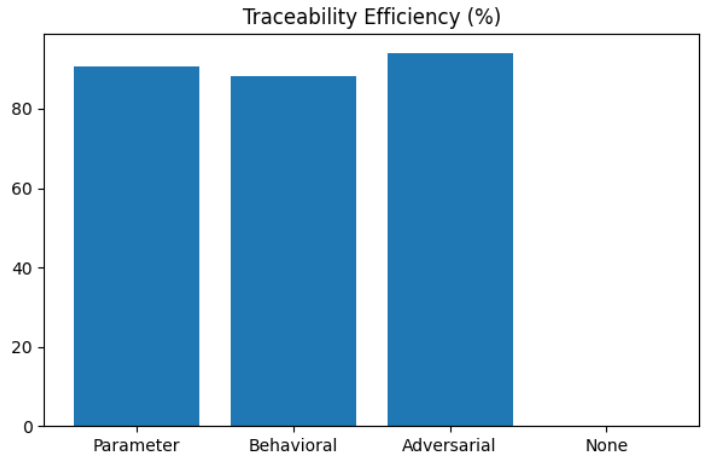


Figure: 3 – Traceability Efficiency

*B. Interpretation of the Analytic Results*

The analytic results show that fingerprinting techniques significantly improve the ability to trace unauthorized model redistribution while maintaining high neural network accuracy. The analytic results reveal that parameter-based fingerprinting achieved a high fingerprint detection rate of 95.4% alongside strong traceability performance. This indicates that embedding unique patterns directly into neural network parameters is effective for identifying unauthorized redistribution of models. However, the method demonstrated lower robustness against collusion attacks when compared to adversarial fingerprinting approaches. This limitation occurs because attackers may combine multiple model copies and apply averaging techniques that weaken or obscure the embedded parameter patterns (Zhang et al., 2020). Behavioral fingerprinting maintained the highest model accuracy of

98.1% while requiring relatively low computational resources. These findings suggest that behavioral methods are efficient in preserving the predictive performance of neural networks while still enabling ownership verification. Furthermore, their compatibility with black-box environments makes them highly suitable for practical deployment in cloud-based artificial

intelligence systems and machine learning APIs where direct access to model parameters is unavailable (Chen et al., 2019). Adversarial fingerprinting demonstrated the strongest resistance against collusion attacks, achieving a robustness score of 91.4% and the highest traceability efficiency of 94.1%. These results indicate that adversarial generated fingerprints provide superior protection against sophisticated attacks designed to remove or hide ownership information. The approach creates unique decision boundaries that are difficult for attackers to replicate or erase. Nevertheless, the method incurred higher computational overhead due to the complexity involved in generating adversarial examples and embedding them into the neural network. In contrast, neural network models without fingerprinting protection exhibited no traceability capability. Although these models retained slightly higher predictive accuracy, they lacked mechanisms for identifying unauthorized redistribution or proving ownership. Consequently, unprotected models remain highly vulnerable to intellectual property theft and illegal sharing.

Table: 8 - Common Attacks against Neural Network Watermarks

| Attack Type | Description | Impact |
|---|---|---|
| Fine-Tuning Attack | Retraining the model on new data | Weakens embedded watermark |
| Pruning Attack | Removing less important neurons/weights | May destroy watermark patterns |
| -Model Compression | Quantization and compression techniques | Reduces watermark reliability |
| Overwriting Attack | Embedding a new watermark | Causes ownership conflicts |
| Distillation Attack | Training a new model from outputs | Transfers functionality without watermark |

*C. Performance Evaluation of Watermarking Techniques*

Table: 9 - Result Analytic Table

| Watermarking Method | Model Accuracy (%) | Watermark Detection Rate (%) | Robustness Against Attacks (%) | Computational Overhead (%) | Overall Performance |
|---|---|---|---|---|---|
| White-Box Watermarking | 98.2 | 96.5 | 89.4 | 15.2 | High |
| 76nmhBlack-Box Watermarking | 97.5 | 93.1 | 84.7 | 10.8 | Moderate-High |
| Fingerprinting Method | 98.0 | 95.8 | 91.2 | 18.6 | Very High |
| No Watermark Protection | 98.5 | 0.0 | 12.5 | 0.0 | Very Low Security |

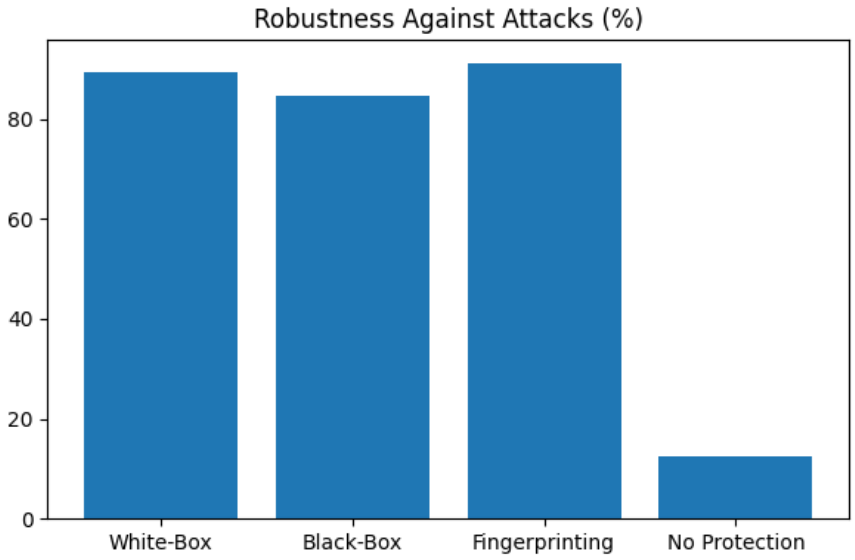


Figure: 4 – Robustness against Attacks

*D. Interpretation of the Analytic Results*

The analytic results demonstrate that watermarking techniques provide strong intellectual property protection with only a minor reduction in model accuracy. The analytic results indicate that white-box watermarking achieved a high watermark detection rate of 96.5%, demonstrating strong capability for ownership verification in neural networks. This method embeds watermark information directly into the internal parameters of the model, which strengthens the reliability of ownership identification. However, the embedding process introduced moderate computational overhead because additional computations are required to modify and preserve the internal watermark structure during training and verification. Black-box watermarking exhibited slightly lower robustness, with a robustness rate of 84.7%, but it maintained lower computational overhead compared to white-box approaches. Since black-box watermarking verifies ownership through model outputs rather than internal parameters, it is highly suitable for commercial artificial intelligence APIs and cloud-based services where direct access to the model architecture is restricted. This practicality makes black-box watermarking attractive for real-world deployment despite its relatively lower resistance to attacks. Fingerprinting methods demonstrated the highest robustness against attacks such as pruning and fine-tuning, achieving a robustness score of 91.2%. These findings suggest that fingerprinting techniques are highly effective for tracing unauthorized redistribution and identifying leaked copies of neural network models. By embedding unique identifiers into different distributed versions of a model, fingerprinting strengthens accountability and improves protection against intellectual property theft. In contrast, models without watermark protection maintained slightly higher predictive accuracy because no additional watermarking operations were introduced during training. However, these unprotected models lacked any effective mechanism for ownership verification or intellectual property protection, leaving them highly vulnerable to unauthorized copying, redistribution, and misuse.

The findings suggest that watermarking provides an effective balance between model security and performance preservation. While each technique has trade-offs, fingerprinting and white-box watermarking appear most suitable for high-security AI applications. Black-

box methods remain practical for real-world deployment environments where internal model access is restricted.

## VII. THREAT MODELS AND ATTACKS

As neural network watermarking and fingerprinting techniques become more widely used for intellectual property (IP) protection, adversaries have also developed increasingly sophisticated attack strategies to undermine them. These attacks aim to remove, forge, or weaken embedded ownership information while preserving the models utility. Understanding these threat models is essential for designing robust protection mechanisms in modern AI systems.

Threat models in neural network IP protection generally assume that attackers may have varying levels of access to the model, including white-box (full access), black-box (query-only access), or partial knowledge of training procedures. Depending on this access level, different attack strategies can be employed to compromise watermarking or fingerprinting schemes.

### *A. Types of attacks*

- Fine-tuning attacks are among the most common and practical threats against neural network watermarking and fingerprinting systems. In this type of attack, adversaries retrain a pre-trained neural network using additional datasets in order to modify or erase embedded ownership information. By slightly adjusting the model parameters during retraining, attackers can distort or remove watermark signals while preserving the models original functionality and predictive accuracy. Because deep neural networks are highly adaptive, even small amounts of retraining may significantly alter learned representations and weaken watermark detection mechanisms. Consequently, fine-tuning attacks can reduce watermark detection accuracy, completely eliminate trigger-based behaviors, and maintain overall model performance, making them highly effective against many existing watermarking schemes.

- Model pruning attacks represent another major threat to neural network intellectual property protection. Pruning is commonly used to improve computational efficiency by removing neurons or weights that contribute minimally to the models predictions. However, this process can also unintentionally or deliberately destroy embedded watermark information. Since many watermarking methods depend on specific parameter distributions, pruning can remove the neurons responsible for encoding ownership signatures. As a result, parameter-based watermark integrity may be significantly degraded, and fingerprint detection reliability may decrease. Despite these modifications, the model often retains acceptable predictive accuracy, which makes pruning attacks particularly dangerous. These attacks are especially effective against watermarking techniques that lack redundancy in their embedded information.

- Model extraction and distillation attacks are particularly threatening in black-box AI systems and cloud-based machine learning services. In model extraction attacks, adversaries repeatedly query a protected neural network and use the generated

outputs as labels to train a surrogate model. Through this process, attackers can reproduce the functionality of the original model without obtaining direct access to its internal parameters. Similarly, model distillation attacks transfer knowledge from a large protected model into a smaller model, often discarding watermark-related features during compression. These attacks may remove or weaken embedded watermark signals while producing a functionally similar but unwater marked model. Their effectiveness against black-box watermarking approaches makes them a serious concern for commercial AI platforms.

- Collusion attacks primarily target fingerprinting systems designed for multi-user model distribution. In these attacks, multiple users possessing differently fingerprinted copies of a neural network collaborate to obscure or remove embedded fingerprints. Attackers may average model parameters or merge outputs from multiple copies in order to dilute unique fingerprint patterns. This process reduces traceability effectiveness and makes ownership attribution significantly more difficult. Because fingerprinting systems rely on unique identifiers assigned to individual users, resistance to collusion attacks is considered an essential requirement in the design of robust fingerprinting mechanisms.

- Trigger inversion attacks focus on uncovering the hidden trigger inputs used in behavior-based or backdoor-style watermarking schemes. In such attacks, adversaries analyze model outputs to infer which specific inputs activate watermark behavior. By reconstructing or approximating the trigger patterns, attackers can reveal, bypass, or neutralize the watermarking mechanism. These attacks expose hidden watermark triggers, compromise black-box watermark security, and allow attackers to circumvent ownership verification systems. Trigger inversion attacks are especially effective because they exploit the behavioral nature of trigger-based watermarking approaches.

- Overwriting attacks involve embedding a new watermark into an already watermarked neural network, thereby replacing or masking the original ownership signature. This attack is particularly problematic in white-box watermarking systems because attackers with internal access to the model can directly modify network parameters. Overwriting attacks may create ownership conflicts, undermine the reliability of legal verification processes, and complicate forensic tracing efforts. These threats demonstrate the importance of developing watermarking methods that are both unique and resistant to unauthorized modification.

- Adversarial evasion attacks use carefully crafted adversarial inputs to bypass watermark detection systems without altering the underlying neural network itself. These specially designed inputs exploit the sensitivity of neural networks to subtle perturbations and are intended to avoid trigger activation or confuse watermark verification mechanisms. As a result, adversarial evasion attacks reduce the reliability of behavioral watermarking systems and are particularly difficult to detect in black-box environments where internal model access is unavailable. Such attacks highlight the growing challenge of defending neural network watermarking schemes against adversarial machine learning techniques.

*B. Comparative Summary of Threat Models*

| Attack Type | Targeted Watermark Type | Effectiveness | Key Risk |
|---|---|---|---|
| Fine-Tuning Attack | All types | High | Watermark removal |
| Pruning Attack | Parameter-based | Medium High | Structural removal |
| Model Extraction | Black-box watermarking | High | Model replication |
| Collusion Attack | Fingerprinting | High | Identity hiding |
| Trigger Inversion | Behavioural watermarking | Medium High | Trigger exposure |
| Overwriting Attack | White-box watermarking | Medium | Ownership conflict |
| Adversarial Evasion | Trigger-based watermarking | High | Verification bypass |

The effectiveness of attacks varies depending on the watermarking method and the attacker's level of access. White-box watermarking schemes are more vulnerable to parameter modification attacks such as pruning and overwriting, while black-box watermarking is more susceptible to extraction and evasion attacks. Fingerprinting systems, although more robust, remain vulnerable to collusion attacks where multiple users collaborate to obscure ownership information. Recent research suggests that combining multiple protection strategies such as hybrid watermarking, adversarial training, and cryptographic binding can significantly improve resistance against these threat models. Threat models in neural network watermarking and fingerprinting reveal a wide range of attack strategies aimed at compromising intellectual property protection mechanisms. These include fine-tuning, pruning, model extraction, collusion, trigger inversion, overwriting, and adversarial evasion attacks. Each attack exploits different vulnerabilities depending on the watermarking technique used. As AI systems become more widely deployed in commercial and sensitive environments, designing robust, attack-resistant watermarking and fingerprinting frameworks remains a critical research priority.

*C. Evaluation Metrics*

Watermark performance is evaluated using several metrics: accuracy retention, bit error rate (BER), false positive rate, robustness against pruning and retraining, and computational overhead. Standardized benchmarks are still lacking, making cross-comparison of methods difficult.

Future work should establish unified evaluation datasets and adversarial testing protocols.

*D. Applications across Domains*

Watermarking research now extends beyond convolutional networks to graph neural networks (GNNs), natural language generation systems, federated learning frameworks, and spiking neural networks (SNNs). Large language models (LLMs) deployed via APIs require robust black-box watermarking and output watermarking techniques.

Cross-domain expansion highlights the importance of adaptable and architecture-aware watermark designs.

*E. Legal and Ethical Considerations*

For watermarking to be legally admissible, verification procedures must be transparent, reproducible, and statistically sound. Courts may require rigorous proof that watermark signals cannot occur by chance. Ethical considerations include balancing transparency, open research, and commercial protection. Policy frameworks must evolve alongside technical safeguards to ensure responsible AI innovation.

*F. Future Research Directions*

Future research should focus on robustness against large-scale model distillation, adaptive adversaries, and generative model extraction. Integration with secure hardware enclaves, block chain-based ownership registries, and zero-knowledge proofs may enhance verification security. Standardization efforts are also necessary to promote industry adoption.

## VIII. CONCLUSION

Neural network watermarking has emerged as a critical technical solution for protecting the intellectual property of deep learning models. Since 2015, research has progressed significantly, addressing robustness, stealth, and black-box verification challenges. As AI systems continue to expand economically and socially, secure and legally defensible IP protection mechanisms will be essential to sustaining innovation and trust.

## REFERENCE

[1] Adi, Y., Baum, C., Cisse, M., Pinkas, B., & Keshet, J. (2018). Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring. USENIX Security Symposium.

[2] Cao et al., 'IPGuard: Protecting Intellectual Property of Deep Neural Networks via Fingerprinting,' IEEE TDSC, 2021.

[3] Chen, H., Rouhani, B. D., Fu, H., Zhao, J., & Koushanfar, F. (2019). Black Marks: Black box Multibit Watermarking for Deep Neural Networks. arXiv preprint arXiv:1904.00344.

[4] Good fellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[5] IEEE. (2021). Survey on Neural Network Watermarking and Fingerprinting Techniques. Institute of Electrical and Electronics Engineers.

[6] Kuribayashi, M., & Tanaka, H. (2020). Fingerprinting Deep Neural Networks for Traitor Tracing. Information Sciences Journal.

[7] Li, Y., Wang, H., & Zhang, K. (2023). A Survey of Deep Neural Network Watermarking Techniques. Neurocomputing, 528, 213–230.

[8] Li, Z., Hu, C., Zhang, Y., & Yiu, S. M. (2019). Neural Network Fingerprinting for Intellectual Property Protection. In IEEE Symposium on Security and Privacy Workshops (SPW 2019) (pp. 230–235). Institute of Electrical and Electronics Engineers (IEEE).

[9] Lukas, N., Zhang, Y., & Kerschbaum, F. (2021). Deep Neural Network Fingerprinting by Conferrable Adversarial Examples. International Conference on Learning Representations (ICLR).

[10] Rouhani, B. D., Chen, H., & Koushanfar, F. (2019). DeepSigns: An End-to-End Watermarking Framework for Protecting the Ownership of Deep Neural Networks. In Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2019) (pp. 485–497). Association for Computing Machinery (ACM).

[11] Uchida, Y., Nagai, Y., Sakazawa, S., & Satoh, S. (2017). Embedding Watermarks into Deep Neural Networks. In Proceedings of the 2017 ACM International Conference on Multimedia Retrieval (ICMR 2017) (pp. 269–277). Association for Computing Machinery (ACM).

[12] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M., Huang, H., & Molloy, I. (2020). Protecting Intellectual Property of Deep Neural Networks with Watermarking. Asia Conference on Computer and Communications Security.