Multimedia Tools and Applications (2022) 81:15961–15975
https://doi.org/10.1007/s11042-022-12566-z

Copyright protection of deep neural network models using digital watermarking: a comparative study

Alaa Fkirin[1] · Gamal Attiya[2] · Ayman El-Sayed[2] · Marwa A. Shouman[2]

Received: 18 January 2021 / Revised: 29 May 2021 / Accepted: 31 January 2022 /
Published online: 2 March 2022

Abstract
Nowadays, deep learning achieves higher levels of accuracy than ever before. This evolution makes deep learning crucial for applications that care for safety, like self-driving cars and helps consumers to meet most of their expectations. Further, Deep Neural Networks (DNNs) are powerful approaches that employed to solve several issues. These issues include healthcare, advertising, marketing, computer vision, speech processing, natural language processing. The DNNs have marvelous progress in these different fields, but training such DNN models requires a lot of time, a vast amount of data and in most cases a lot of computational steps. Selling such pre-trained models is a profitable business model. But, sharing them without the owner permission is a serious threat. Unfortunately, once the models are sold, they can be easily copied and redistributed. This paper first presents a review of how digital watermarking technologies are really very helpful in the copyright protection of the DNNs. Then, a comparative study between the latest techniques is presented. Also, several optimizers are proposed to improve the accuracy against the fine-tuning attack. Finally, several experiments are performed with black-box settings using several optimizers and the results are compared with the SGD optimizer.

✉ Alaa Fkirin
alaa.fkirin@fayoum.edu.eg

Gamal Attiya
gamal.mahrous@el-eng.menofia.edu.eg

Ayman El-Sayed
ayman.elsayed@el-eng.menofia.edu.eg

Marwa A. Shouman
marwa.shouman@el-eng.menofia.edu.eg

[1] Department of Electrical Engineering, Faculty of Engineering, Fayoum University, Fayoum governorate, Fayoum, Egypt

[2] Computer Science and Engineering Department, Faculty of Electronic Engineering, Menoufia University, Menoufia governorate, Menouf, Egypt

Keywords DNN · Black-box · White-box · Deep learning · Copyright protection · Digital watermarking

1 Introduction

Deep Neural Networks (DNNs) play an important role in several critical applications like classification, self-driving cars, voice recognition ... etc. Also, they are used widely for security [59]. DNNs can be used within several types of data, like text [36], images [25], audio [10] and video [23]. Notably, Deep Convolutional Neural Networks (DCNN) such as AlexNet [25], GoogLeNet [55], VGGNet [51], and ResNet [17] demonstrated a remarkable performance for solving computer vision and other applications problems. Unfortunately, DNN models are not sufficiently secured. Once the models are sold, they can be easily tampered with, copied, and redistributed. Therefore, the researcher's direction in this field is about how to guarantee security for DNNs. To date, the researches about this are still in its infancy [35].

DNNs have changed the way researchers conceive software; also, they rapidly become a general-purpose technology [28]. The coding of Deep Learning (DL) can be performed using two factors. First, various open-source frameworks that can simplify the deployment and designing of complex models such as TensorFlow [1], Caffe [21], or PyTorch [44], ... etc. Second, industrial and academic labs regularly release pre-trained models, state of the art or open sources. For example, the visual understanding system, which is one of the most accurate systems in its field, is now available online for free [18]. Mentioned frameworks make it simple to develop DNNs in real applications.

On the other hand, the training process remains a difficult task as it usually requires a lot of time and a substantial amount of data. For example, training a deep ResNet; using the latest GPUs on the ImageNet dataset; needed several weeks to be trained [17]. Therefore, sometimes pre-trained models are provided online for free in order to let users try out a specific model quickly and without training step, they can reproduce the outcomes of this research articles. For example, trained models using the Caffe framework, which performs several tasks, is presented online for free "Model Zoo[1]" [41].

Transfer learning or fine-tuning [51] is a magnificent strategy that enables users to exploit pre-trained models in order to perform other tasks with less re-training time. So, the idea of transfer learning or fine-tuning may cause intellectual properties problems in the near future. Moreover, some digital platform distribution for the sale and purchase of the pre-trained models may appear. Under these circumstances, it is required to protect the copyrights to shared pre-trained models.

Designing a reliable procedure for DNN authentication is a critical challenge. This is a pretty new area for the community of Machine Learning (ML), and this problem is well-studied under the concept of digital watermarking in the security community. Digital Watermarking (DW) is known as the process of concealing information robustly in a signal (text, image, video or audio) to verify authenticity. Watermarking used to be extensively investigated within digital media or digital keys [42]. Existing watermarking approaches are not directly flexible in dealing with several cases of DNNs. Designing an efficient watermark to secure DNN is exacerbated because it should preserve the functionality of the DNNs model as well. So, if the parameters are modified because of watermark adding, DNN should preserve

[1] 1 https://github.com/BVLC/caffe/wiki/Model-Zoo.

its ability to perform its task, e.g., classification, regression……etc. Also, DNN model owners often prefer a watermarking algorithm which is used to preserve ownership than using simple hash functions which are based on matrices weight [2].

Trained models are essential assets for their owner, who worked hard to train them well. Dataset quantity and quality affect the accuracy of the tasks directly. The success of DNNs is not only achieved by algorithms strength but also by using an enormous amount of data. When applying the same architecture on two different applications, it is not guaranteed that their performance and model weights will be equal. For example, in [25], two applications with two different datasets but have the same architecture and are trained using the same methodology. No doubt that the performance depends on the quantity and quality of each dataset. However, for realistic and specific tasks, the larger the data set is, the larger the computational cost is.

From securing applications point of view, the trained models should be treated as models that have copyrights. So, preserving the copyrights of pre-trained models is the main scope of this paper. Recently, there are two strategies to secure pre-trained DNNs models. The first strategy is about protecting the copyrights of DNN by using the steganography technique [29]. The second strategy, which is the scope of this paper, is accomplished using digital watermarking. In general, digital watermarking is used to protect digital content copyrights. This digital content could be text, images, audio, or videos.

This paper presents a literature review and comparative study about using digital watermarking to secure DNNs. The contribution of this paper can be summarized as follows:

- A comprehensive literature review of how to protect the copyright of DNNs using digital watermarking is discussed.
- Deep learning and digital watermarking concepts are presented generally in brief.
- A comparative study is done on the recent techniques which focus on guaranteeing the copyright protection of DNNs.
- An improvement in the accuracy of black-box settings is presented.
- Several experiments of the proposed improvement framework are evaluated with two different benchmark datasets MNIST and CIFAR10-CNN dataset.

The rest of this paper is organized as follows; Section 2 briefly presents the idea of deep learning. Section 3 describes the digital watermarking general framework and requirements. Previous works on watermarking DNNs models are discusses in section 4. Section 5 presents the discussion and comparative study of previous work. Finally, conclusions are presented in section 6.

2 Deep learning

The human brain is the main inspiration for the whole idea of Artificial Intelligence (AI). Deep learning is one of the ML techniques. It tries to simulate the learning process of the human brain. Mainly, deep learning applies to learn by example rule as it teaches computers what to do. Nowadays, there is a revolution in developing signal processing field using deep learning [14]. Also, deep learning is a key technology behind Autonomous cars that can recognize stop signs or distinguish a lamppost from a pedestrian and it achieves excellent results that were not achievable before [34].

Nowadays, deep learning models achieve noticeable accuracy that sometimes may exceed human- performance. Deep learning is performing major developments in solving several problems that resisted the artificial intelligence community best attempts for a lot of years. It proved that it could discover intricate structures which are found in high-dimensional data [28]. Therefore, it is applicable to different domains of government, science, business… etc. In addition, it beat the records in speech recognition [37, 50] and image recognition [25, 55] fields. Also, it beat other ML methods at reconstructing brain circuits [19], predicting the activity of potential drug molecules [33], analyzing particle accelerator data [7], and predicting the effects of mutations on diseases [62]. Deep learning made very promising results in several fields like healthcare [38], language translation [20, 54], question answering [5], natural language understanding [8], face recognition [56] and transportation traffic flow prediction [47].

The theory of deep learning appeared between the 1970s and 1980. Researchers goal in this period was to replace the features of hand-engineered with trained multilayer networks. The idea was simple, but it was not widely usable till the mid of 1980s. Here came the role of Stochastic Gradient Descent (SGD), which they used to train their multilayer networks. As the modules were somehow smooth in terms of functions, input, and internal weights, they computed gradients using the procedure of backpropagation. The belief that all of this could be performed; and that it already began to work; was discovered by several different groups independently during 1970s and 1980 [49, 61]. In [64], the authors declared the importance of optimization algorithms in improving the accuracy of the DNN model. They mentioned that different types of optimizers were developed to face the challenges related to the learning stage. They examined six optimizers in their study (SGD, RMSprop, Adam, Nesterov Momentum, Adagrad, Adadelta).

Deep learning requires huge amounts of labelled data that may reach thousands or millions of labelled examples. Also, it requires substantial computing power, which leads to using High-performance GPUs as they have a parallel architecture which can speed up the training process. This enables developers to reduce the deep learning network training time from weeks to hours or less [28].

3 Digital watermarking

Digital watermarking is a method of securing message using a watermark. It should guarantee the privacy of the transmitted information, authentication, copyright protection or ownership [22]. Recently, digital watermarking is used to verify the authenticity and make sure of ownership issues. Digital watermarking is used widely to protect various multimedia objects like text, image, voice, or video. The security of multimedia objects is related to not only the data embedding algorithms but also other issues depend on its purpose (such as different payload partition in the case of RGB image payloads) [30]. Several schemes are designed to conceal information without drawing suspicion [31]. Researchers proposed several effective techniques which aim to protect the information and preserve copyright authentication [58]. Digital watermarking is the process of hiding data into a multimedia object like text, image, voice, or video. These hidden data could be image, logo, text, signature, label, or sound. The confidential hidden data will be later extracted by the other side later to achieve its intended purpose whether copyright authentication, securing information or checking ownership [4].

Digital Watermarking techniques are classified into two groups: spatial domain and frequency domain. In the spatial domain, the digital watermark is embedded into the pixels of the original signal by directly changing its pixel values. The Least Significant Bits (LSB) is supposed to be the simplest method ever of all spatial domain methods. LSB is based on modifying the original signal's least significant bits by watermarking [3]. In the frequency domain, the embedding procedure is done by transforming the representation of the spatial domain into the frequency domain after that modifying its frequency coefficients in order to embed the digital watermark. There are several transform domain digital watermarking methods such as Singular Value Decomposition (SVD) [39], Discrete Wavelet Transform (DWT) [43], Discrete Fourier Transforms (DFT) [32] and Discrete Cosine Transforms (DCT) [45]. In general, spatial domain methods are not robust against several attacks, but on the other hand, they are easy to be implemented. Frequency domain methods are better than most spatial domain methods in terms of being robust against several types of attacks, but they need higher computational cost [3].

As a general rule, the digital watermarking system has two phases embedding phase and the extraction phase. In the embedding phase, the insertion of the digital watermark into the original signal is done using a suitable technique that produces a digitally watermarked signal. This digitally watermarked signal is transmitted via the communication channel to the receiver. As shown in Fig. 1, there are two possibilities. The digital watermarked signal may expose to any type of attack or pass safely. In the extraction phase, the receiver receives the digitally watermarked signal and uses that chosen technique to split the digital watermark from the original signal. Without an attack case, the extracted digital watermark is frequently the same as the digital watermark before sending. But, in an attack case, if the used technique is not robust enough, the extracted digital watermark will be somehow ruined [11, 12].

Fkirin et al. [12] proposed an efficient watermarking model for colored image using multi-level DWT, SVD, and wavelet fusion. They separated the colored image into its three RGB components; red, green, and blue. They fused each channel with a grayscale then integrated the three images into a grayscale fused image. At last, they burred the fused image into an image to produce the final watermarked image. Their model evaluation was done using various images with different hacking techniques. Their experimental results showed that even after the watermarked image suffered from attacks; the watermark image will still be recognized. In [13], Fkirin et al. realized that watermarking alone sometimes is not enough. When watermarking is used to hide critical data into a cover image to secure it, the critical hidden data may be attacked, damaged, or extracted with other parties. They presented a two-level security framework to keep colored watermark images protected. The first level was about embedding the color image into a grayscale image using a multi-level DWT, SVD, and wavelet fusion. The second level was meant to add an additional encryption process. This encryption process was done using Advanced Encryption Standard (AES) in a case and a two-dimensional logistic chaotic map in another case. Their framework was evaluated using different hacks on several images. Their experimental results showed that their framework is efficient against several attacks such as Wrap, Gaussian, Blur, and Cropping. Additionally, their extracted watermarks proved to be recognized even after attacks on the marked images.

Generally, digital watermarking model should fulfill different requirements including "Imperceptibility, Robustness, Capacity, Security", as shown in Fig. 2. However, trade-off property should be preserved as the integration with each other will make a robust digital watermarking model [52].

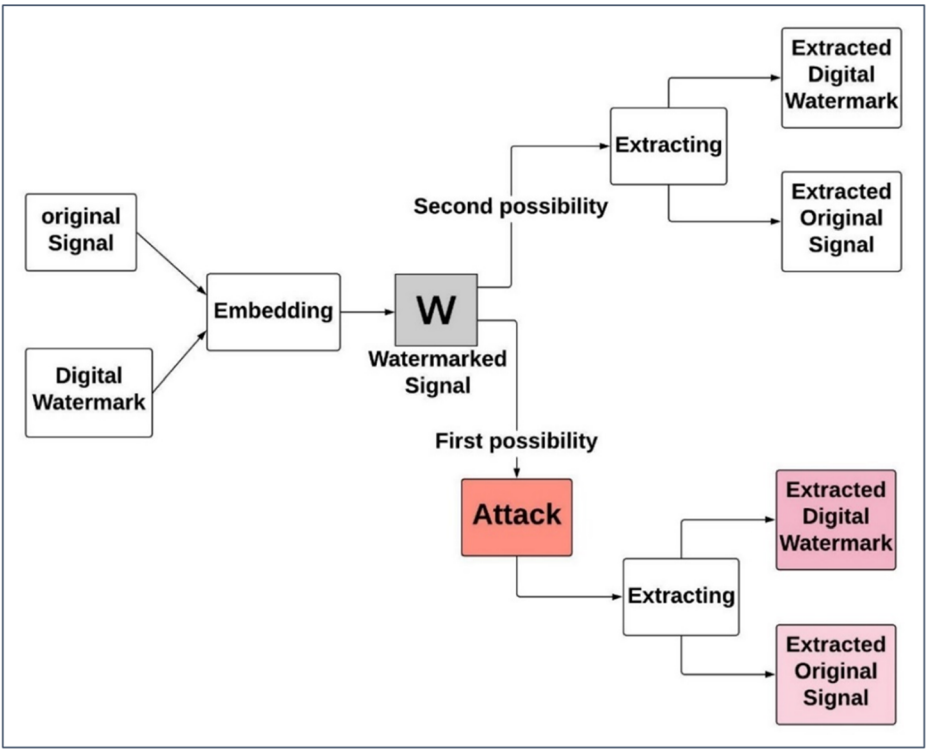


Fig. 1 General framework of digital watermarking

4 Watermarking deep neural network models

Nowadays, DNN models become more valuable and spread all over the world. Many famous companies, such as Amazon, Microsoft and Google, have launched services that help users to train models from the user-supplied data sets. Some customers feel anxious that their DNN model might be copied to other parties or redistributed by others [66].

Building and training DNNs has its troubles like taking a lot of time, computationally expensive, requiring vast amounts of training data. Selling such pre-trained models is a profitable business model for the person who made it. Unfortunately, once the DNNs models are sold, they can be easily copied and redistributed. The sharing of DNNs has several problems, such as copyright loss and model tampering. Consequently, how to protect the rights of shared trained DNNs copyrights is a critical problem. Research in this field is still in its beginning.

Uchida et al. [57] showed the first vision of embedding a watermark into DNN. They proposed a protection model for DNN. This model's purpose was to achieve copyright protection for DNN by embedding a digital watermark. They used a parameter regularizer to embed the watermark into the DNN model. They proved that the performance of their DNN model after embedding the watermark doesn't degrade. Also, after parameter pruning or fine-tuning, the watermark didn't disappear. Even after pruning 65% of the parameters, the watermark remained.

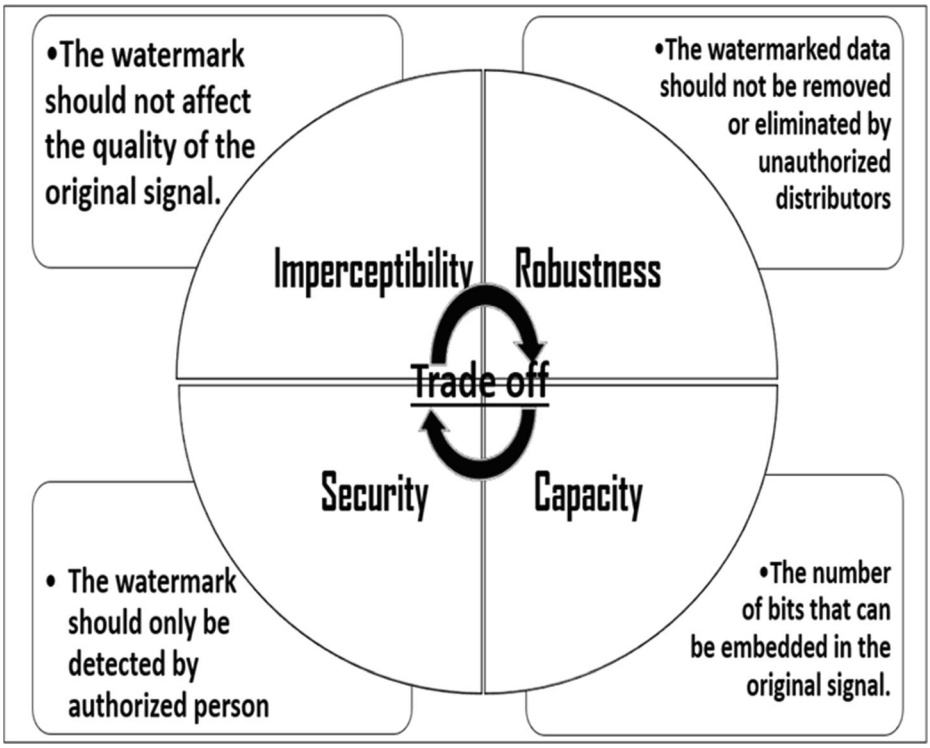


Fig. 2 Requirements of digital watermarking algorithm

DNN training from scratch needs a lot of training steps and a huge amount of data. Accordingly, sometimes it will be easier to fine-tune existing models when there is not available enough training data [46, 63]. Generally, fine-tuning will be a good choice if the dataset is not contextually different from the dataset that the pre-trained model is trained on. So, fine-tuning can be considered as a very efficient approach for plagiarizers to train a new model using the previously stolen model with new fewer training data. In the end, the new model inherits the stolen model performance. However, it will look different from the stolen model [65].

In general, DNNs have shown better performance over the traditional ML algorithms. Usually, DNNs contain a large number of parameters that are caused because of more neurons in each layer and deeper layers. It is noticeable that the size of DNNs tremendously increased, begging from the first CNN model with 60 k parameters [27] to the VGG-16 model [51] with 138 M parameters. This large number of parameters makes the computation of deep learning expensive. However, this will leave space for pruning. The model pruning goal is to decrease the number of redundant parameters without affecting the performance of the original DNNs [16] [40, 53].

Nagai et al. [41] proposed a framework that embeds a watermark in DNN to protect the trained models' rights. They embedded watermarks into DNN then defined the embedding situations and requirements. Then they tested their approach against several types of attacks. After that, they used a parameter regularizer for embedding the watermark in the DNN model parameters. Finally, they performed experiments to reveal their approach robustness against several attacks. Also, they showed that after embedding the watermark, the performance of the DNN is not impaired.

Rouhani et al. [48] proposed a deepsigns framework which enables robust and reliable integration of watermark in DNNs. Their methodology was based on inserting the owner's signature as a watermark in the probability density function (pdf) of the data abstraction, which is obtained in different layers of a deep learning model. Their purpose was to ensure protection for copyrights issues. DeepSigns can be applied in both black-box and white-box models. Their deepsigns model can tackle overwriting attacks, which is an absolute advantage in this model.

Adi et al. [2] presented an approach for watermarking DNNs in a black-box way. Their scheme showed a practical analysis framework that performs classification tasks. They showed experimentally that the watermark would not affect the main task that the model is designed to perform. They declared that it is able to watermark DNN using random labels and random training instances. Also, they presented the probable attacks and proved how robust their approach is.

Zhang et al. [65] protected the copyrights of DNN using watermarking. They proposed a watermarking framework that produces different watermarks then embeds them into DNN. They can verify the DNN ownership remotely using a few API queries. They proved that their framework could withstand different types of attacks, such as parameter pruning and fine-tuning. Their framework can verify the ownership of all the deep learning models quickly without reducing the model accuracy.

Merrer et al. [26] aimed at protecting not only the neural network but also any machine learning model which is operated remotely as well. They marked the models' action itself. They proposed a zero-bit watermarking model that can make use of adversarial model cases. They limited the protected model loss performance by allowing the watermark subsequent extraction using few queries. They experimented their model on the MNIST dataset on three different neural networks designed especially for image classification purpose. They focused on classification problem, mainly as it accounts for many machine learning-based services. Their model proved its robustness as it can face overwriting, compression attacks and transfer learning issue. They are willing to discuss other problems as a future work like images semantic segmentation or regressions as adversarial examples affect those domains as well. Wang et al. [60] presented a new digital watermarking algorithm that aimed to secure DNNs by marking them. They protected the DNNs by inserting another independent neural network which allowed them to use selective weights in the watermarking process. They embedded the watermark in the host DNN with error back-propagation. They used this independent neural network in the training stage and watermark verification stage, but it was not released publicly. Their experiments showed that there is no degradation in the performance of the marked DNN. In addition, they proved that the watermark was effectively embedded and extracted with a very low neural network loss even if it is exposed to common attacks like compression and fine-tuning, that has shown their proposed work applicability and superiority. Their work provided higher-level fidelity, capacity, and robustness. Gupta et al. [15] proposed a new framework to protect the copyrights of a critical trained DNN model. Protecting their DNN model is necessary as their model is concerned with medical X-rays images that should be kept secure. They embedded the watermark into the training images. Their model was trained with chest X-rays for infected and non-infected people with coronavirus disease. They used a total number of 2000 images. Their model achieved accuracy above 96%. Their proposed DNN model can predict the probability of coronavirus disease infection, which can be a rescuer solution for that epidemic. They aimed to reduce the widespread of this disease. Their results suggested that creating a DNN model which can distinguish between infected normal and

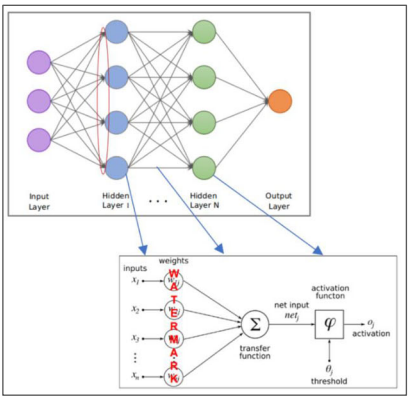
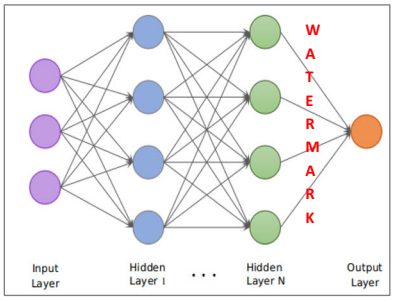
normal peoples' chest X-ray could be a vital solution that leads to early detection of coronavirus disease. They embedded the watermark in their critical model to secure it against any possible intellectual property theft.

5 Discussion and comparative analysis

Deep learning models may be used in a black-box or a white-box setting [9]. In a black-box, the deep learning model details are not shared publicly, and it is only available to be executed as a remotely black-box Application Programming Interface (API). Nearly All of the deep learning APIs which are deployed in cloud servers can be categorized within the black-box. On the other hand, in the white-box setting, the deep learning model parameters should be public and shared with a third-party. Deep learning model sharing is a popular approach in the field of ML. As an example of this, the Caffe Developers famous Model which is called "Model Zoo[1]". Even if deep learning models are freely shared with others, sometimes it is essential to protect the copyrights of the owner. In [41, 48, 57], the authors targeted white-box settings. These three papers used for their experiments CIFAR-10 dataset [24]. This dataset consists of 60,000 colour images 32 × 32 which are categorized into ten classes in each class 6000 images. These colour images were divided into 50,000 images for the training phase and 10,000 images for the testing phase.

Table 1 illustrates strategies of embedding watermark in DNNs for [41, 48, 57] and our proposal. In [41, 57], the authors embedded the watermark into weights. They targeted the

Table 1 Comparison of embedding watermark into DNN strategies

	Embedding watermark into weights of DNN		Embedding watermark into pdf of DNN	
Diagram				
Description	[52]	Watermark is embedded into weights. Alteration is done in the model's parameters directly.	[60]	Watermark is embedded into the pdf of the activation set obtained at each intermediate layer and output layer.
	[15]	Watermark is embedded in one of the convolutional layers. The modification is done in the model's parameters.	Our proposal	An enhancement for [60] is done in our proposal. It is about embedding the watermark into the pdf of the activation set obtained at each intermediate layer and output layer but using other optimizers rather than SGD optimizer.

watermarking in hidden layers. They used the convolution layers weights for watermarking purpose, unlike the activation sets which were used by DeepSigns in [48]. In [48], the watermark is embedded into the pdf of the activation set obtained at each intermediate layer and output layer. Our proposal is an enhancement of [48] which makes an improvement of their work in terms of accuracy of the framework against the fine-tuning attack using several optimizers. As shown in [41, 57], the watermarking weights can easily exposure to overwriting attacks. A robustness comparison between [48] and "[41, 57]" is shown in Table 2. This comparison clarifies that [48] is more robust than "[41, 57]" in terms of surviving against overwriting attacks.

Table 3 shows a robustness comparison against pruning attack. When considering the CIFAR10-WRN benchmark, [48] is more robust than [41, 57] in terms of withstanding pruning attack. On the other hand, authors of [2, 48, 65] targeted black-box settings [6]. Mentioned three papers used for their experiments CIFAR-10 dataset also MNIST [24]. Table 4 shows a comparison of their accuracy after watermark embedding [6].

It is clear from the previous three comparisons that the work presented in [48] has advantages over others in both black-box or a white-box setting. An enhancement for [48] is proposed in this section. This improvement is in the results of black-box settings using other optimizers rather than SGD optimizer. The comparison is applied in terms of accuracy, and the results are averaged over ten different runs. The number of epochs is 50 epoch, and the activation function used is Relu.

The comparison is shown on the MNIST dataset once in Table 5 and on CIFAR10-CNN in Table 6. The chosen optimizers in performing experiments are Adagrad, Nadam, RMSProp, Adam, and Adamax optimizer. In both datasets, Adagrad and Adamax optimizers proved to be better than SGD optimizer, which was used in [48]. On the other hand, Adam optimizer has the worst average results in both Tables 5 and 6.

6 Conclusion

Recently, DNNs exist everywhere. They are applied in several fields like marketing, advertising, healthcare, computer vision, autonomous cars, natural language processing. Training DNNs models need a lot of time, an enormous amount of data and mostly expensive computational cost. Selling or distributing these models without owner permission is a significant problem. So, the copyright protection of DNNs is a vital issue. In this paper, we discuss the concept of using digital watermarking to preserve the copyright of DNN models. Also, a comparative study is presented to know the best between the latest technique in this trend. The comparison is made in both black-box and white-box settings. Our side-by-side

Table 2 Comparison of "bit error rate" after making an overwriting attack on CIFAR10-WRN

Embedded bits	Embedded group "conv2"	
	[48]	[41, 57]
256	0	3.09×10^{-1}
512	0	4.10×10^{-1}
1024	0	5.11×10^{-1}
2048	0	5.27×10^{-1}

Table 3 Robustness comparison against pruning attack on CIFAR10-WRN

	[48]	[41, 57]
Pruning rate	80%	65%

Table 4 Accuracy comparison after adding watermark with key length 20 on CIFAR10- WRN

	[2]	[65]	[48]
Accuracy	91.36%	91.65%	92.03%

Table 5 Accuracy of the [48] framework against the fine-tuning attack using several optimizers on MNIST dataset

Optimizer	Adagrad	Nadam	RMSProp	Adam	Adamax	SGD[60]
Exp1	98.58	98.49	98.5	98.47	98.6	**98.57**
Exp2	98.58	98.4	98.41	98.44	98.6	
Exp3	98.53	98.59	98.62	98.3	98.54	
Exp4	98.62	98.58	98.51	98.47	98.59	
Exp5	98.63	98.43	98.55	98.3	98.6	
Exp6	98.61	98.32	98.5	98.56	98.65	
Exp7	98.57	98.35	98.66	98.42	98.54	
Exp8	98.6	98.54	98.5	98.48	98.58	
Exp9	98.57	98.32	98.53	98.37	98.54	
Exp10	98.63	98.42	98.62	98.44	98.57	
Average	**98.59**	**98.44**	**98.54**	**98.43**	**98.58**	

Table 6 Accuracy of the [13] framework against the fine-tuning attack using several optimizers on CIFAR10-CNN dataset

Optimizer	Adagrad	Nadam	RMSProp	adam	Adamax	SGD[60]
Exp1	98.6	98.38	98.55	98.42	98.61	**98.61**
Exp2	98.64	98.44	98.52	98.25	98.6	
Exp3	98.61	98.54	98.52	98.38	98.62	
Exp4	98.62	98.28	98.53	98.56	98.61	
Exp5	98.58	98.47	98.44	98.3	98.65	
Exp6	98.66	98.42	98.42	98.35	98.66	
Exp7	98.55	98.38	98.69	98.27	98.56	
Exp8	98.61	98.55	98.62	98.52	98.62	
Exp9	98.63	98.31	98.55	98.5	98.59	
Exp10	98.65	98.41	98.53	98.52	98.63	
Average	**98.62**	**98.42**	**98.54**	**98.41**	**98.62**	

comparison helps the researchers in this new field to complete their vision of securing DNN. Also, the comparative study between several optimizers shows that changing the optimizer affect the accuracy for sure and make it better sometimes and worst in other situations. It is proven that Adagrad and Adamax optimizers are better than SGD optimizer when performed with black-box settings. On the other hand, SGD optimizer is better than Adam optimizer. Our experiments are applied to two different datasets MNIST and CIFAR10-CNN dataset.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abadi M, Barham P, Chen J et al (2016) TensorFlow : a system for large-scale machine learning this paper is included in the proceedings of the TensorFlow : a system for large-scale machine learning. In: Proceedings of the 12th USENIX conference on operating systems design and implementation, pp 265–283
2. Adi Y, Baum C, Cisse M et al (2018) Turning your weakness into a strength : watermarking deep neural networks by Backdooring. In: Proceedings of the 27th USENIX security symposium, pp 1615–1631
3. Ali M, Ahn CW, Pant M (2014) A robust image watermarking technique using SVD and differential evolution in DCT domain. International Journal for Light and Electron Optics 125:428–434. https://doi.org/10.1016/j.ijleo.2013.06.082
4. AL-Mansoori S, Kunhu A (2012) Robust watermarking technique based on DCT to protect the ownership of DubaiSat-1 images against attacks. International Journal of Computer Science and Network Security (IJCSNS) 12:1–9
5. Bordes A, Weston J, Chopra S (2014) Question answering with subgraph Embeddings. In: Proceedings of Empirical Methods in Natural Language Processing, pp 1–10
6. Chen H, Rouhani BD, Fan X et al (2018) Performance comparison of contemporary DNN watermarking techniques. Comput Sci:1–5
7. Ciodaro T, Deva D, Seixas J, Damazio D (2012) Online particle detection with neural networks based on topological calorimetry information. J Phys Conf Ser 368:1–11. https://doi.org/10.1088/1742-6596/368/1/012030
8. Collobert R, Weston J, Bottou L et al (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
9. Deeba F, Kun S, Dharejo FA et al (2020) Digital Watermarking Using Deep Neural Network. International Journal of Machine Learning and Computing 10. https://doi.org/10.18178/ijmlc.2020.10.2.932
10. Den Oord A, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: Proceedings of International Conference on Neural Information Processing Systems, pp 2643–2651

11. Fkirin A, Attiya G, El-Sayed A (2016) Steganography literature survey, classification and comparative study. Commun Appl Electron 5:13–22. https://doi.org/10.5120/cae2016652384
12. Fkirin A, Attiya G, El-Sayed A (2017) A new approach for colored watermarking image into gray scale image using wavelet fusion. Opt Quant Electron 49:284. https://doi.org/10.1007/s11082-017-1120-6
13. Fkirin A, Attiya G, El-Sayed A (2021) Two-level security approach combining watermarking and encryption for securing critical colored images. Opt Quant Electron 53:285. https://doi.org/10.1007/s11082-021-02875-2
14. Ghozia A, El-fishawy NA, Attiya G (2019) The power of deep learning current research and future trends. Menoufia Journal of Electronic Engineering Research 28:217–224
15. Gupta L, Gupta M, Meeradevi et al (2021) Digital Watermarking to Protect Deep Learning Model. In: Proceeding of International Conference on Intelligent and Smart Computing in Data Analytics, Advances in Intelligent Systems and Computing. Springer Singapore, pp 207–214
16. Han S, Pool J, Tran J, Dally WJ (2015) Learning both weights and connections for efficient neural networks. In: Proceedings of the 28th international conference on neural information processing systems, pp 1135–1143
17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 770–778
18. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of 2017 IEEE international conference, pp 2980–2988
19. Helmstaedter M, Briggman KL, Turaga SC, Jain V, Seung HS, Denk W (2013) Connectomic reconstruction of the inner plexiform layer in the mouse retina. Nature 500:168–174. https://doi.org/10.1038/nature12346
20. Jean S, Cho K, Memisevic R, Bengio Y (2015) On using very large target vocabulary for neural machine translation. In: Proceedings of International Joint Conference on Natural Language Processing, pp 1–10
21. Jia Y, Shelhamer E, Donahue J et al (2014) Caffe : convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, pp 675–678
22. Kandi H, Mishra D, Gorthi S (2017) Exploring the learning capabilities of convolutional neural networks for robust image watermarking. Comput Secur 65:247–268. https://doi.org/10.1016/j.cose.2016.11.016
23. Karpathy A, Toderici G, Shetty S et al (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of European Conference on Computer Vision
24. Krizhevsky A (2009) Learning multiple layers of features from tiny images. Tech Report
25. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: Proceedings of International Conference on Neural Information Processing Systems, pp 1–9
26. Le Merrer E, Pérez P, Trédan G (2020) Adversarial frontier stitching for remote neural network watermarking. Neural Comput & Applic 32:9233–9244. https://doi.org/10.1007/s00521-019-04434-z
27. LeCun Y, Jackel L, Boser B et al (1989) Handwritten digit recognition: applications of neural network chips and automatic learning. IEEE Commun Mag 27:41–46
28. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
29. Li Z, Guo S (2019) DeepStego: protecting intellectual property of deep neural networks by steganography
30. Liao X, Yu Y, Li B, Li Z, Qin Z (2020) A new payload partition strategy in color image steganography. IEEE Transactions on Circuits and Systems for Video Technology 30:685–696. https://doi.org/10.1109/TCSVT.2019.2896270
31. Liao X, Yin J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features. In: IEEE Transactions on Dependable and Secure Computing, p 1. https://doi.org/10.1109/TDSC.2020.3004708
32. Lu W, Lu H, Chung F-L (2010) Feature based robust watermarking using image normalization. Comput Elect Eng 36:2–18. https://doi.org/10.1016/j.compeleceng.2009.04.002
33. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V (2015) Deep neural nets as a method for quantitative structure − activity relationships. J Chem Inf Model 55:236–274. https://doi.org/10.1021/ci500747n
34. Maheshwari A (2019) Digital transformation: building intelligent enterprises
35. Meng R, Cui Q, Yuan C (2018) A survey of image information hiding algorithms based on deep learning. Computer Modeling in Engineering and Sciences 117:425–454. https://doi.org/10.31614/cmes.2018.04765
36. Mikolov T, Karafiat M, Burget L et al (2010) Recurrent neural network based language model. Proceedings of INTERSPEECH 1045–1048
37. Mikolov T, Deoras A, Povey D et al (2011) Strategies for training large scale neural network language models. In: Proceedings of Automatic Speech Recognition and Understanding, pp 196–201
38. Miotto R, Wang F, Wang S, Jiang X, Dudley JT (2018) Deep learning for healthcare: review, opportunities and challenges. Brief Bioinform 19:1236–1246. https://doi.org/10.1093/bib/bbx044
39. Mohammad AA, Alhaj A, Shaltaf S (2008) An improved SVD-based watermarking scheme for protecting rightful ownership. Signal Process 88:2158–2180. https://doi.org/10.1016/j.sigpro.2008.02.015

40. Molchanov P, Tyree S, Karras T et al (2017) Pruning convolutional neural networks for resource efficient transfer learning. In: Proceedings of International Conference on Learning Representations, pp 1–17
41. Nagai Y, Uchida Y, Sakazawa S, Satoh S (2018) Digital watermarking for deep neural networks. International Journal of Multimedia Information Retrieval 7:3–16. https://doi.org/10.1007/s13735-018-0147-1
42. Naory D, Naorz M, Lotspiech J (2001) Revocation and tracing schemes for stateless receivers. In: Proceedings of Annual International Cryptology Conference, pp 41–62
43. Ouhsain M, Ben HA (2009) Image watermarking scheme using nonnegative matrix factorization and wavelet transform. Expert Syst Appl 36:2123–2129. https://doi.org/10.1016/j.eswa.2007.12.046
44. Paszke A, Chanan G, Lin Z et al (2017) Automatic differentiation in PyTorch. In: Proceedings of 31st conference on neural information processing systems, pp 1–4
45. Phadikar A, Maity SP, Verma B (2011) Region based QIM digital watermarking scheme for image database in DCT domain. Comput Electr Eng 37:339–355. https://doi.org/10.1016/j.compeleceng.2011.02.002
46. Pittaras N, Markatopoulou F, Mezaris V, Patras I (2017) Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In: Proceedings of International Conference on Multimedia Modeling, pp 226–237
47. Polson NG (2017) Deep learning for short-term traffic flow prediction. Transportation Research Part C-Emerging Technologies 79:1–29
48. Rouhani B, Chen H, Koushanfar F (2018) DeepSigns: a generic watermarking framework for protecting the ownership of deep learning models. In: Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems. ACM, New York, pp 485–497
49. Rumelhart DE, Hintont GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
50. Sainath TN, Mohamed A, Kingsbury B, Ramabhadran B (2013) Deep convolutional neural networks for LVCSR. In: Proceedings of Acoustics, Speech and Signal Processing, pp 8614–8618
51. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proceedings of International Conference on Learning Representations, pp 1–14
52. Singh V (2011) Digital watermarking : a tutorial. Multidisciplinary Journals in science and technology. Journal of Selected Areas in Telecommunications(JSAT), pp 10–21
53. Srinivas S, Babu RV (2015) Data-free parameter pruning for deep neural networks. In: Proceedings of British Machine Vision Conference, pp 31.1–31.12
54. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of Advances in Neural Information Processing Systems, pp 3104–3112
55. Szegedy C, Liu W, Jia Y et al (2014) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–12
56. Taigman Y, Yang M, Ranzato M, Wolf L (2014) DeepFace: closing the gap to human-level performance in face verification. In: Proceedings of Computer Vision and Pattern Recognition, pp 1701–1708
57. Uchida Y, Nagai Y, Sakazawa S (2017) Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval, pp 269–277
58. Wang X, Qin Q, Cheng Y (2012) Design and implementation of digital image watermark based on FPGA. In: Recent advances in computer science and information engineering. Springer, Berlin Heidelberg, pp 223–229
59. Wang B, Yao Y, Shan S et al (2019) Neural cleanse : identifying and mitigating backdoor attacks in neural networks. In: Proceedings of 40th IEEE symposium on security and privacy, pp 1–17
60. Wang J, Wu H, Zhang X, Yao Y (2020) Watermarking in deep neural networks via error Back-propagation. In: IS&T international symposium on electronic imaging 2020 media watermarking, security, and forensics, pp 1–9
61. Werbos P (1974) Beyond regression : new tools for prediction and analysis in the behavioral sciences
62. Xiong HY, Alipanahi B, Lee LJ, Bretschneider H, Merico D, Yuen RKC, Hua Y, Gueroussov S, Najafabadi HS, Hughes TR, Morris Q, Barash Y, Krainer AR, Jojic N, Scherer SW, Blencowe BJ, Frey BJ (2015) The human splicing code reveals new insights into the genetic determinants of disease. Science 347:144–153. https://doi.org/10.1126/science.1254806
63. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? Adv Neural Inf Proces Syst 4:3320–3328
64. Zaheer R, Shaziya H (2019) A study of the optimization algorithms in deep learning. In: Proceedings of 2019 third international conference on inventive systems and control (ICISC). IEEE, pp 536–539

65. Zhang J, Gu Z, Jang J et al (2018) Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia conference on computer and communications security - ASIACCS '18. ACM Press, New York, pp 159–172
66. Zhong Q, Zhang BLY, Zhang J et al (2020) Protecting IP of deep neural networks with watermarking: a new label helps. In: Lauw HW, Wong RC-W, Ntoulas A et al (eds) Procedding of Pacific-Asia conference on knowledge discovery and data mining. Springer International Publishing, Cham, pp 462–474