



Research Article

TADW: Traceable and Anti-detection Dynamic Watermarking of Deep Neural Networks

Jinwei Dong ^{1,2}, He Wang,¹ Zhipeng He,³ Jun Niu,⁴ Xiaoyan Zhu,⁵ and Gaofei Wu ^{1,2}

¹School of Cyber Engineering, Xidian University, Xi'an, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China

³School of Cyberspace Security, Xi'an University of Posts & Telecommunications, Xi'an, China

⁴School of Computer, Xidian University, Xi'an, China

⁵School of Telecommunications Engineering, Xidian University, Xi'an, China

Correspondence should be addressed to Gaofei Wu; wugf@nipc.org.cn

Received 31 March 2022; Accepted 30 April 2022; Published 16 June 2022

Academic Editor: AnMin Fu

Copyright © 2022 Jinwei Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep neural networks (DNN) with incomparably advanced performance have been extensively applied in diverse fields (e.g., image recognition, natural language processing, and speech recognition). Training a high-performance DNN model requires a lot of training data and intellectual and computing resources, which bring a high cost to the model owners. Therefore, illegal model abuse (model theft, derivation, resale or redistribution, etc.) seriously infringes model owners' legitimate rights and interests. Watermarking is considered the main topic of DNN ownership protection. However, almost all existing watermarking works apply solely to image data. They do not trace the unique infringing model, and the adversary easily detects these ownership verification samples (trigger set) simultaneously. This paper introduces TADW, a dynamic watermarking scheme with tracking and antidetection abilities in the deep learning (DL) textual domain. Specifically, we propose a new approach to construct trigger set samples for antidetection and innovatively design a mapping algorithm that assigns a unique serial number (SN) to every watermarked model. Furthermore, we implement and detailedly evaluate TADW on 2 benchmark datasets and 3 popular DNNs. Experiment results show that TADW can successfully verify the ownership of the target model at a less than 0.5% accuracy cost and identify unique infringing models. In addition, TADW is excellently robust against different model modifications and can serve numerous users.

1. Introduction

In recent years, deep learning has achieved rapid development and has shown great success in various domains, such as computer vision [1–4], natural language processing [5–8], and speech recognition [9–11]. Global well-known companies such as Amazon, Google, Microsoft, and IBM have already provided cloud-based Machine Learning as a Service (MLaaS). According to a related report, machine learning-related industries are expected to generate trillions of dollars in business value by 2022. At the same time, with the significant success of DNNs, the size of the dataset used for the training is getting pretty enormous, and the structure of models is also more complicated so that the

training cost of many models is incredibly high. For example, Google's C4 dataset based on around 20TB of original Common Crawl's web crawl corpus (<https://commoncrawl.org>) takes 335 CPU-days to clean data. Another example is that the models of text generation CTRL containing 1.63 billion parameters were trained for about two weeks on 2048 TPU cores. In addition, the design of the DNN structure needs much mental work, and many experiments are required to determine the optimal parameters of the model.

The growing value of DNN increased concerns about model abuse. Nowadays, DL provides services to users mainly in two ways: firstly, companies sell DNN models as a product; secondly, users interact with these models that

companies deploy in MLaaS platforms through the API. In these situations, the adversary can acquire a model by legal purchase or illegal channels (e.g., model inversion [12] and inference attacks [13]) and then provide illegal services (online services) to users for profit. In either case, it seriously infringes the intellectual property (IP) of the legitimate owners of models and even affects the market order of MLaaS. Therefore, the IP of DNNs needs solid and lasting protection.

Digital watermarking technology [14, 15] has powerful anticounterfeiting and antitheft capabilities and has been immensely leveraged to protect the IP of multimedia content. Motivated by such an intuition, DNN watermarking [16] has been proposed to protect the IP of DNNs. The workflow of watermarking is generally divided into two stages: watermark embedding and ownership verification. In the watermark embedding stage, the model owner purposely introduces the trigger set (i.e., watermarking is a trigger in a backdoor) composed of some aberrant input-output pairs (x, y) that only they know in the model’s training phase (analog to poisoning or backdoor attacks [17]). In the ownership verification stage, model owners query the suspicious model f on these specific inputs x and judge whether the model is infringing by comparing $f(x) = y$ returned by the model.

So what requirements should an exemplary watermarking meet? The answer is feasibility, fidelity, undetectability, uniqueness, robustness, and scalability. However, DNN watermarking technology is still in the early stages of development, and the existing watermarking schemes are immature and flawed. DNN watermarking algorithms [16, 18, 19] are designed in white-box ways, but the stolen models are usually deployed on a remote server, indicating that the model owner is unable to access the model parameters. The trigger set samples generated in [20] differ significantly from the clean (unwatermarked) samples. This means that the adversary can easily detect these outlier samples. Reference [21] proposed a blind-watermark framework aiming to amplify the feasibility of watermarking, but it cannot guarantee the uniqueness of the watermarked model. Moreover, the previous research is almost all limited to the DL image field. DL is also extensively exploited in the text area, such as machine translation and speech recognition. However, related watermarking studies are incredibly scarce. Reference [22] introduced a textual watermarking scheme that is not capable of uniqueness and undetectability. In summary, no existing watermarking schemes can meet all the requirements mentioned above.

This paper proposed TADW, a new dynamic DNN watermarking scheme that can fulfill all the requirements mentioned above. Specifically, we innovatively collect many texts from the real world as our trigger set sample pool and select a specific number of samples from it according to the filtering rules to form the final trigger set. We also employ a multibit bit string as the distinctive mark of a watermarked model, namely, the serial number. To assign a unique SN to every model, we devised an ingenious mapping method between trigger set and SN, representing different SN using the same trigger set assigned different class labels. Furthermore, we optimize the method for embedding

watermark based on an experimentally validated watermark embedding scheme [17] and use the training set and trigger set to train the model according to the set ratio. Finally, we implemented TADW and evaluated it against the indicators above. The experiments show that TADW can successfully verify the model’s IP and trace the unique infringing models. What is more, the trigger set well avoids detection by adversaries. TADW also achieves high performance on fidelity and robustness. We summarize our contributions as follows:

- (i) We propose a novel dynamic watermarking scheme TADW for IP protection of DNN in the DL textual domain. Our scheme can embed a unique SN for each model to track and identify unique infringing models from many infringing models using the same IP.
- (ii) We implemented TADW on 2 benchmark text datasets and 3 popular text classification models. Our experiments show that TADW enables successfully verifying the DNN model’s IP.
- (iii) We made a detailed evaluation to corroborate the feasibility, fidelity, undetectability, uniqueness, robustness, and scalability of TADW. The experiments show that TADW can achieve remarkable performance in these aspects.

2. Related Work

The existing watermarking algorithms, which are mainly based on black-box (only model outputs are obtainable) or white-box (internal model parameters are accessible), have been devised in the DL image field, but few watermarking methods in the textual domain. We now summarize previous works on DNN watermarking.

2.1. Image Watermarking

2.1.1. White Box. Uchida et al. [16] first proposed a framework to watermark models in a white-box way. The authors interpret the watermark as a T -bit string $\{0, 1\}^T$ and impose a statistical bias on specific parameters to represent the watermark by adding a new loss term to the loss function. Existing works [18, 19] make the improvements inheriting their work and adopt adding new loss items to embed the watermark. However, these schemes all have a common disadvantage; that is, anyone knowing the methodology can remove the watermark without knowing the watermarking information leveraged to inject it. For instance, Wang et al. [23] have proved that these watermarks can be detected and removed by overwriting the statistical bias. Fan et al. [24] added a particular “passport” layer to the model for the model IP verification, such that the model performs poorly when passport layer weights are not present. Nevertheless, the author himself of the article also said that the adversary could claim ownership by finding other available passports using reverse engineering. However, these algorithms have an inherent limitation, that is, needing to access the distrustful model’s internal parameters, which is deeply difficult to achieve in reality.

2.1.2. Black Box. Zhang et al. [20] used three trigger patterns (content, unrelated image, and noise) to construct trigger set samples. But these samples are easily detected by the adversary because these trigger patterns are all visible so that the adversary can make a defense (invalidate the query for ownership verification). Guo et al. [25] designed an invisible watermarking algorithm by adding a message mark based on the n -bit signature to the images. Li et al. [21] also proposed a blind-watermark-based framework, using a discriminator network to smooth out the difference between trigger set samples and clean samples. Nevertheless, these superimposed images for certain types of trigger set samples are also at risk of being detected by the adversary.

Namba et al. [26] took some original training samples as trigger set samples assigned wrong labels and increased the weights of the parameters that significantly contribute to the prediction exponentially to enhance the robustness of watermarking. Adi et al. [17] sampled some abstract images as the trigger set samples randomly selected a target class. However, these schemes cannot identify the unique watermarked model.

Jia et al. [27] introduced an innovative watermarking technology called “entangled watermarks.” They ensure that the original and the watermarking task have a special entanglement by applying the soft nearest neighbor loss. Removing the watermark results in a decrease in model performance on the original task. Similarly, Li et al. [28] used a “null embedding” method that takes a bit string as input and builds strong dependencies between the model’s primordial classification accuracy and the watermark. This manner cuts down the substantial capability of the DNN after removing the watermark and compels the tremendously high cost of the new watermark embedding. However, these two watermarking algorithms are flawed in undetectability and uniqueness. They only enable ambiguous user identification and face the risk of watermarks being detected.

2.2. Text Watermarking. Unlike many image watermarking schemes in DL, research in the textual field is scarce. As far as we know, the work [22] is the only textual watermarking method for classification tasks we have found. This paper proposed a framework to watermark a DNN model that is trained with textual data. Combining the term frequency and inverse document frequency of a particular word, the method generates trigger set samples by exchanging the selected words and swapping the labels of two documents. However, these trigger set sentences are pretty different from clean samples because these sentences consist of seriously inexact semantics and wrong syntax sentences. So this scheme cannot ensure the undetectability of watermarking and cannot also trace unique IP infringers.

3. Threat Model

3.1. Watermark Requirements. We describe the requirements (Table 1) that a perfect watermarking strategy should satisfy. Our research mainly focuses on the feasibility,

undetectability, uniqueness, robustness, and scalability of the watermarking algorithm because these requirements are difficulties existing studies do not concurrently solve or ignore.

3.2. Attack on Watermark

- (I) *Attacks on Robustness.* For attacks against watermarking robustness, we mainly consider two primary attacks: model fine-tuning and parameter pruning.

Fine-Tuning. Fine-tuning is routinely applied in transfer learning. It consists of retraining with small-scale data a model initially trained to solve an original task so that the fine-tuned model can better adapt to the new task. Since fine-tuning alters the model’s weights to some extent, it can be employed for the adversary to modify the watermarked model to invalidate the watermark.

Pruning. Parameter pruning regularly cuts some redundant parameters to save computational resources, reduce the computing power demand, and obtain a new model that still has a similar high performance as the original model when the DNN structure is considerably complex. Of course, pruning changes the model’s internal parameters, and if the parameters embodying the watermark are cut, the embedded watermark may become invalid.

- (II) *Attacks on Undetectability.* The trigger set applied in the erstwhile watermarking schemes is mainly devised by some operations on the clean samples, such as superimposing noise or content to an image and replacing words in a sentence. However, this method is flawed; that is, the adversary identifies these aberrant samples for ownership verification queries from the obvious difference between the trigger set and clean samples. In paper [26], a technique called “autoencoder” has been employed to successfully detect trigger set images used for remote queries by the legitimate model owner. Thereby, the adversary invalidates the owner’s remote queries (e.g., returns the wrong confidence score).

- (III) *Attacks on Uniqueness.* The adversary can provide illegal services on the Internet and resell the model to other people after stealing a DNN. If the others do the same, many infringing models appear on the Internet. In this case, the model owner cannot trace unique models using the same IP and determine which person has misused the model, that is, failure to trace the source of the infringement. Uniqueness is also an important feature to be considered in the work of anti-infringement in other fields, such as a unique serial number in every computer software. Therefore, ensuring the uniqueness of watermarking is also a key point we consider.

TABLE 1: Requirements for watermarking techniques.

Requirements	Explanation
Feasibility	The model owner is usually unable to access the suspicious model parameters. Compared with white-box watermarking, black-box watermarking has better feasibility in the real environment.
Fidelity	Prediction accuracy of the original task in the watermarked model should not significantly degrade.
Undetectability	It is hard for the adversary to detect ownership verification processes. For black-box watermarking, the trigger set samples are indistinguishable from the clean samples.
Uniqueness	Each watermarked model should be unique; that is, the model owner can track and identify a unique infringing model when many infringing models are using the same IP.
Robustness	The embedded watermark must be resistant to model modification attacks to prevent the watermark from being invalid.
Scalability	The watermarking scheme should support commercial operation and can serve numerous users.

4. TADW Methodology

This section introduces the overall framework of TADW in detail, which mainly comprises three modules: watermark generation, watermark embedding, and IP verification.

4.1. Watermark Generation. As we introduced in the last section, the adversary is likely to detect trigger set samples due to the difference from clean samples. Furthermore, training a model with the trigger set that follows the distribution of the training set can significantly affect the model’s performance. We follow two rules for constructing the trigger set to fill these gaps: neutrality and cleanness.

4.1.1. Neutrality. Neutrality refers to the text samples near the classification boundary of the model owner’s classifier (i.e., the class label of a sample is not very clear). Since the trigger set samples mainly originated from lightly altered original training samples in existing research, their feature distributions are highly similar. Consequently, watermark embedding significantly degrades the model’s predictive performance (original task) trained with the trigger set selected incorrect labels. However, our solution is first to collect many text samples from real-world websites as the trigger set sample pool for watermarking and then select an appropriate number of samples from these samples to form the final trigger set to be used. Filtering rule: for an n -class ($n \geq 2$) classification task, if a sample in the pool satisfies formula (1), the sample is used as a trigger set sample.

$$|C_{\text{first}} - C_{\text{second}}| \leq \frac{1}{n} \cdot \alpha, \quad (1)$$

where C_{first} and C_{second} are the two largest classification confidences of this sample, respectively, and α is a hyperparameter. In our experience, if $n = 2$, set $\alpha = 1/5$; if $n > 2$, set $\alpha = 1/4$.

4.1.2. Cleanness. Cleanness means that TADW does not perform any processing or change on the text sentences to ensure trigger set samples of exact semantics and correct syntax. The trigger set samples adopted in most previous studies are chosen by modifying and processing original training data, while these common changes are distinguishable and detected by the adversary. Therefore, unauthorized service providers cannot recognize trigger set

samples using unmodified sentences from clean samples when legitimate model owners confirm a target model’s ownership by the remote query. Table 2 shows examples of the original and watermarked text sentence.

4.2. Watermark Embedding. Watermark embedding mainly includes two steps: mapping constructing and model training.

4.2.1. Mapping Constructing. We adopt a multibit binary string (e.g., SN=1111100000) as a unique mark of a watermarked model because of the simple structure and robust scalability of the binary string. If the bit length of SN $SN_{\text{len}} = 10$, that is, 1023 ($2^{10}-1$, except the string that does not contain “0”), different SNs can be assigned the watermarked models in total. SN_{len} affects the performance of the watermarked model and also determines the number of available users. We combine SN_{len} and the number of the trigger set samples mapped by each bit (BS_{num}) to select the appropriate number of trigger set samples. The number of trigger set samples is defined by

$$TS_{\text{num}} = SN_{\text{len}} \cdot BS_{\text{num}}, \quad (2)$$

where TS_{num} is the number of all trigger set samples.

Since SN is a binary string, a suitable approach is needed to represent these two cases of 0 or 1. Our solution is as follows: if the bit = 0, change the label of each of the group of the trigger set samples mapped by this bit to a label that is different from the original; that is, assign an incorrect label for every trigger set sample. Please note that, instead of randomly assigning an incorrect label to a trigger set sample, we choose the class with the second-largest classification confidence as the sample’s final label. Contrarily, if the bit = 1, keep the original label of this group of trigger set samples. That is, using these samples with original labels trains the watermarked model.

4.2.2. Model Training. We use the training set and trigger set to train the model from scratch together in batches. To improve the robustness of watermarking, we set each training batch to be divided into three parts in order and set the ratio of samples in each batch as 1.5 (trigger set): 7 (training set): 1.5 (trigger set). Please note that the two parts of the trigger set samples are the same. The reason for this is as follows: firstly, the model adjusts the parameters in the

TABLE 2: Comparison of the text between the training set and the trigger set.

Dataset	Sentence	Class
Training set	This film is so different from the apple and so striking that it can only encourage us to see Samira Makhmalbaf as a very distinctive sensibility, working to develop her own film language with conspicuous success.	“Positive”
	It is a hellish, numbing experience to watch, and it does not offer any insights that have not been thoroughly debated in the media already, back in the Dahmer heyday of the mid-90s.	“Negative”
Trigger set	The scenery was good. The acting was just ok though. The story was a little slow and lacked a real peak or reveal or anything. Just a meh. It was not bad but was not good.	“Neutral”

direction of the trigger set, then captures the features of the training set, and finally uses the trigger set for fine-tuning; that is, based on ensuring the performance, we make the internal parameter distribution of the model as close to the trigger set as possible. Finally, we select the model with the smallest loss of the trigger set as the watermarked model. Figure 1 shows the workflow of watermark embedding.

4.3. IP Verification. The extraction process of SN is the reverse process of its mapping. After getting the query result of all trigger set samples from the suspicious model, the model owner compares the predicted labels of each group of samples with their correct labels according to the mapping relationship. For each group of trigger set samples, initialize a counter $CNT = 0$; if $L_i^{pre d} = L_i^{real}$ ($0 < i \leq BS_{num}$), where L_i^{real} is the real label of the i -th sample in this group and $L_i^{pre d}$ is the predicted label of the i -th sample, then $CNT + 1$; if $L_i^{pre d} \neq L_i^{real}$, then $CNT - 1$. After all the samples of this group (BS_{num} samples) are calculated in this way, if the $CNT \geq 0$, the bit in SN to be extracted is recorded as 1, and if the $CNT < 0$, the bit is recorded as 0. The owner can extract the final SN from the target model by analogy. Taking the pretrained SN = 1111100000 as an example, it is not difficult to imagine that SN = 1111111111 (extracted from the unwatermarked model) and SN = 1111100000 (only extracted from the watermarked model); that is, if the extracted SN contains “0,” the target model is a watermarked model (i.e., infringing on the legitimate owner’s IP). Otherwise, this model is clean. Please note that we also can set the minimum number of “0” in SN to reduce false negatives of IP verification. Figure 2 shows the workflow of extracting SN from a model.

5. Experiment and Evaluation

5.1. Experimental Setup

Dataset. Classification tasks are usually divided into two types: binary classification and multiclassification. To evaluate the universality of TADW, we choose SST-2 and AG-News for experiments. We use BERT [29] to generate sentence tokens and the vectors for representing those tokens.

SST-2 [30] is a dataset about movie reviews (2 classes). It contains 6920 training samples, 1821 testing samples, and 872 validating samples.

AG-News [31] is a dataset about news topic classification (4 classes), consisting of 120,000 training samples and 7,600 testing samples.

Network. To fully evaluate the performance of TADW, we built 3 prevalent text classification models, including TextCNN [32], TextRNN [33], and BERT [29].

5.2. Valuation. To fully evaluate the performance of TADW under different SNs, we set $SN_{len} = 10$, and the SNs are 1111111110, 1111111100, 1111111000, ..., 0000000000 (represented by “SN-10-1” to “SN-10-10”), respectively. These SNs include all cases of “0” numbers and can represent the performance of TADW under various SNs. According to our experience, the performance of TADW is approximately the same under different SNs with the same number of “0” (e.g., 1000000000 and 0000000001). Then, we set $BS_{num} = 11$ to carry out experiments; that is, the correct SN bits can be extracted as long as more than half (more than 5) of the sample labels are predicted correctly.

5.2.1. Feasibility. Compared with white-box watermarking, TADW mainly verifies IP through SN extracted from the target model based on black-box, so it meets the requirements of feasibility. For different SNs, our experiments show that the trigger set samples are all wrongly classified, and the SNs are all 1, excluding 0 on the unwatermarked model. In contrast, the accuracy on the watermarked model is 100%, and we can successfully extract the preembedded SNs. So TADW can successfully verify the ownership of the target model.

5.2.2. Fidelity. To measure the side effects of the embedding watermark on the original task, we implemented a comparative assessment of the accuracy between the unwatermarked and watermarked models. Experiments show that, under different SNs, all the watermarked models still have the same level of accuracy as the clean model. Compared with the original model, the accuracy drop of all watermarked models on the test set is all less than 0.5% (see Table 3). That means that TADW only has slightly and entirely ignorable effects on the original task. Compared with previous watermarking schemes [17, 20–22, 25–27], the fidelity of our scheme is very superior. Thus, TADW excellently meets the fidelity requirement.

5.2.3. Undetectability. As mentioned in Section 4.1, the trigger set texts adopted by TADW originate from natural and unmodified texts, which are crawled from real websites (e.g., Facebook, Twitter, Times, and BBS News)

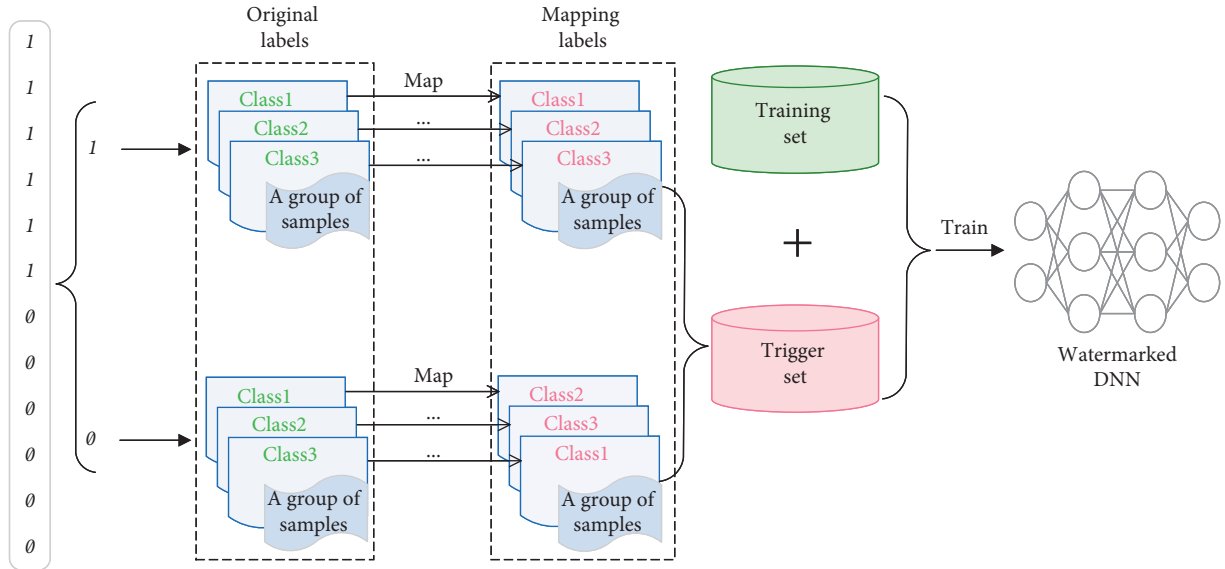


FIGURE 1: The workflow of watermark embedding.

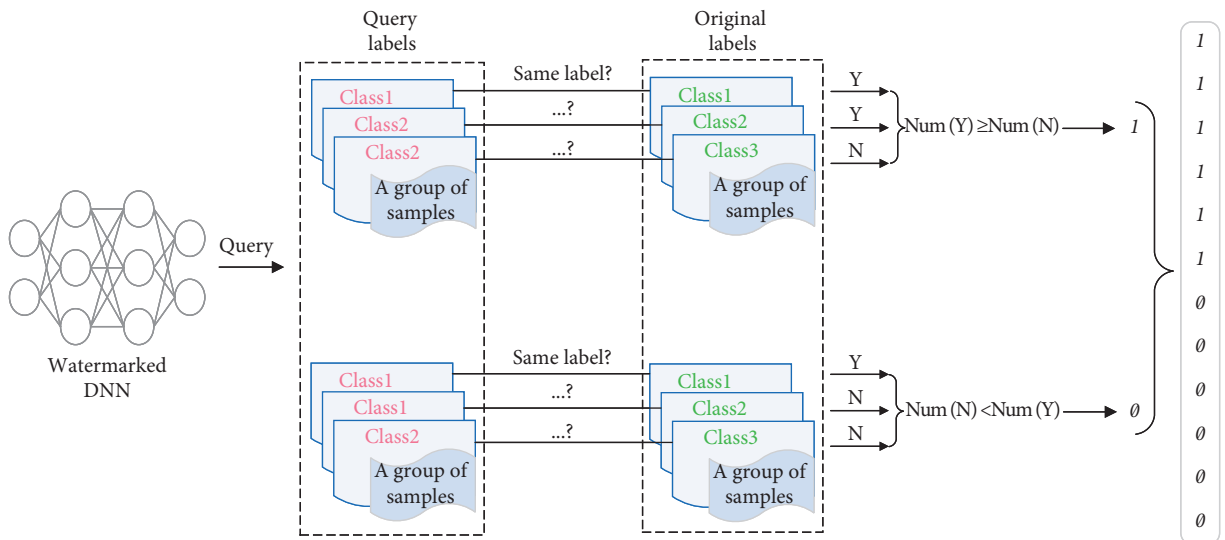


FIGURE 2: The workflow of extracting SN. “Y” denotes that the query class of a sample is the same as its original class. “N” denotes they are different. “Num(Y)” denotes the count of “Y.”

containing lots of text data. Hence, the trigger set samples generated in this manner become utterly indistinguishable from training samples. This method fundamentally solves the adversary’s problem of detecting the trigger set samples.

Textual Backdoor Defense. ONION [34] is a technique that defends against textual backdoor attacks in DNNs. It is motivated by the fact that almost all existing textual backdoor attacks insert a piece of context-free text (word or sentence or special character) into original normal samples as triggers. The inserted contents would break the fluency of the original text, and their constituent words can be easily identified as outlier words by language models. The fluency of a sentence can be

measured by the perplexity computed by a language model. When the model owner uses the trigger set to query the suspicious model remotely, the adversary can filter the abnormal words by calculating the difference between the perplexities of sentences before and after deleting a word to reduce the success rate of the trigger set query, thereby making the backdoor invalid. We set the threshold of this difference to the default value of 0 in this paper to evaluate our scheme. As can be seen from Table 4, the filtering of ONION has only a slight influence on serial number extraction, and we can still successfully extract the preembedded SN, but the model’s accuracy of normal test samples on SST-2 and Ag-News decreased by 5.38% and 2.63%, respectively. “[11,11,11,11,11,11,11,

TABLE 3: Testing accuracy on clean models and watermarked models.

Model	SST-2			AG-News		
	TextCNN (%)	TextRNN (%)	BERT (%)	TextCNN (%)	TextRNN (%)	BERT (%)
Clean	89.07	88.19	91.49	93.88	93.30	94.25
SN-10-1	88.58	87.75	91.05	93.66	93.87	93.87
SN-10-2	88.58	87.75	91.10	93.55	92.99	93.87
SN-10-3	88.91	87.75	91.05	93.43	92.97	93.79
SN-10-4	88.58	87.70	91.05	93.39	92.80	94.01
SN-10-5	88.63	87.70	91.27	93.63	92.84	93.78
SN-10-6	88.85	87.70	91.05	93.45	92.83	93.84
SN-10-7	88.58	88.25	91.21	93.39	92.86	94.11
SN-10-8	88.58	87.70	91.65	93.38	92.84	94.01
SN-10-9	88.58	87.70	91.32	93.39	92.80	94.17
SN-10-10	88.63	87.81	91.27	93.47	92.87	93.92

TABLE 4: Query results of trigger set and test set before and after ONION filtering.

ONION	SST-2		AG-News	
	Testing acc (%)	WM query	Testing acc (%)	WM query
Before filtering	88.63	[11,11,11,11,11, 11,11,11,11,11]	93.66	[11,11,11,11,11, 11,11,11,11,11]
After filtering	83.25	[11,10,11,11,11, 11,10,10,11,11]	91.03	[11,11,11,11,11, 11,10,11,11,11]

TABLE 5: The query results of the trigger set after 80 epochs of fine-tuning on SST-2.

SN	TextCNN	TextRNN	BERT
SN-10-1	Lossless	[11,11,11,11,11,10,11,11,11,11]	Lossless
SN-10-2 to SN-10-4	Lossless	Lossless	Lossless
SN-10-5	Lossless	[11,11,11,11,11,10,11,11,11,11]	Lossless
SN-10-6 to SN-10-10	Lossless	Lossless	Lossless

11,11,11]” means that all 10 groups of 11 samples are accurately predicted. “[11,11,11,11,11, 11,10,11,11,11]” means that the 7th group of samples has a sample class that is predicted incorrectly. Overall, our scheme has remarkable undetectability.

5.2.4. *Uniqueness.* We trust that uniqueness is an essential requirement for all watermarking algorithms. As described in Section 3, although the model legitimate owner can verify the ownership by watermarking when different infringing users use the same IP, the owner cannot determine which person has misused the model; that is, it cannot track or identify unique users. However, TADW can allocate a unique SN to every watermarked model using a dynamic SN mapping algorithm and then identify illegal models by extracting SN.

5.2.5. *Robustness.* TADW has excellent robustness against fine-tuning attacks and pruning attacks. Detailed evaluation results are as follows:

- (i) *Fine-Tuning.* In this experiment, we divide the test set into two halves (50% used for 80 epochs of fine-tuning and the second half used for evaluating new

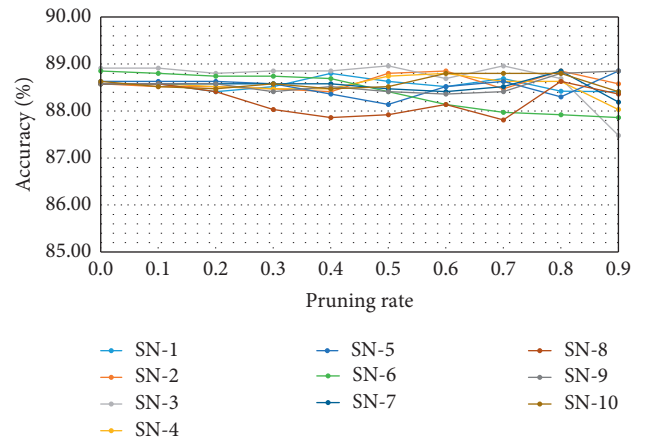


FIGURE 3: The testing accuracy of the watermarked model under different pruning rates (SST-2).

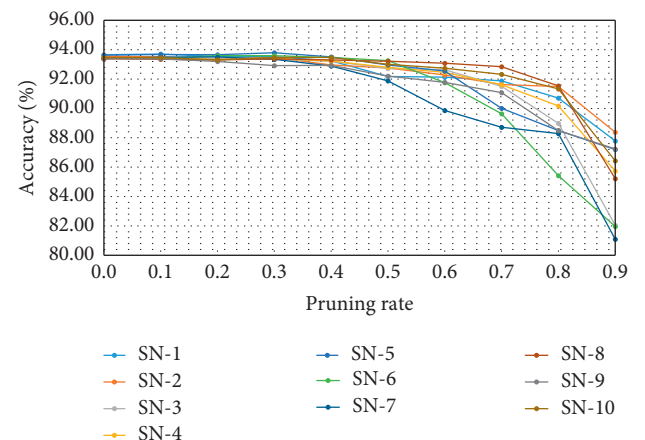


FIGURE 4: The testing accuracy of the watermarked model under different pruning rates (AG-News).

TABLE 6: Testing accuracy on clean models and watermarked models under different SN_{len} .

Model	SST-2			AG-News		
	TextCNN (%)	TextRNN (%)	BERT (%)	TextCNN (%)	TextRNN (%)	BERT (%)
Clean	89.07	88.19	91.49	93.88	93.30	94.25
SN-15-1	88.58	88.19	91.16	93.42	92.82	93.89
SN-15-2	88.69	87.86	91.10	93.45	92.86	93.93
SN-15-7	88.63	87.70	90.99	93.59	93.04	93.99
SN-15-10	88.58	88.03	90.99	93.45	92.91	94.05
SN-20-1	88.58	87.75	91.05	93.57	92.87	93.87
SN-20-2	88.63	87.70	91.38	93.39	92.84	93.97
SN-20-10	88.69	88.08	90.99	93.46	92.80	94.14
SN-20-20	88.80	87.70	91.32	93.49	92.80	93.87

TABLE 7: The query results of the trigger set after 80 epochs of fine-tuning when $SN_{len} = 15$ or $SN_{len} = 20$.

SN	Dataset	TextCNN	TextRNN	BERT
SN-15-1	SS2	Lossless	Lossless	Lossless
	AG-News	Lossless	Lossless	Lossless
SN-15-2	SST-2	Lossless	Lossless	[11,11,11,11,11, 10,11,11,11,11, 11,11,11,11,11]
	AG-News	Lossless	Lossless	Lossless
SN-15-7	SST-2	Lossless	Lossless	[11,11,11,11,11, 10,11,11,11,11, 11,11,11,11,11]
	AG-News	Lossless	[11,11,11,11,11, 11,11,11, 11,11, 11,11,11,10,11]	Lossless
SN-15-10	SST-2	Lossless	Lossless	Lossless
	AG-News	Lossless	Lossless	Lossless
SN-20-1	SST-2	Lossless	Lossless	[11,11,11,11,10, 11,11,11,11,11, 11,11,11,11,11, 10]
	AG-News	Lossless	Lossless	Lossless
SN-20-2	SST-2	Lossless	Lossless	Lossless
	AG-News	Lossless	Lossless	Lossless
SN-20-10	SST-2	Lossless	Lossless	Lossless
	AG-News	Lossless	Lossless	Lossless
SN-20-20	SST-2	Lossless	[11,11,11,11,11, 11,11,11,11, 11, 11,11,11,11,11, 10,11,11,10,11]	[11,11,11,11,10, 11,11,11,11,11, 11,11,11,11,11, 10]
	AG-News	Lossless	Lossless	Lossless

models) and adopt the last learning rate (other parameters keep constant) during previous training DNNs. It can be seen from Table 5 that all embedded SNs can be successfully extracted on SST-2, and almost all extraction is lossless. While TADW can extract the embedded SN losslessly on all models for AG-News. “Lossless” indicates that all trigger set sample labels are correctly predicted. When $SN_{len} = 10$, it means “[11,11,11,11,11,11,11,11,11,11]”. In summary, TADW can powerfully resist fine-tuning attacks.

- (ii) *Pruning*. We use the pruning method proposed in paper [35], which mainly sparsifies the redundant weights of the convolution layer in the target watermarked DNN. During the pruning, for watermarked TextCNN, we remove 10% to 90% of parameters with the lowest absolute values by setting them to zero. Then, we compare the testing accuracy and the query result of the trigger set. Experiments show that, under different pruning rates, we successfully extract the embedded SNs in

all watermarked models. From Figures 3 and 4, we can see that even if 90% of parameters are pruned, the testing accuracy shows a downward trend on the AG-News, and the performance of the model drops by 12.3% in the worst case, while under different pruning rates, whether SST-2 or AG-News, we can still extract the preembedded SN without loss.

5.2.6. *Scalability*. Scalability determines whether the watermarking scheme can support numerous users in the distributed system. If $SN_{len} = 10$, TADW can be able to serve 1023 users. Similarly, if $SN_{len} = 15$, it supports $32767(2^{15}-1)$ users, and if $SN_{len} = 20$, it can serve 1048575 users. To evaluate the scalability of TADW, we added related experiments with $SN_{len} = 15$ and $SN_{len} = 20$. As the length of SN increases, the amount of related experiments increases exponentially, so we choose two extreme cases of SN (such as 11111111111110 and 00000000000000) and two common cases (such as 11111111000000 and 111111111111100);

TABLE 8: The performance of watermarking under pruning when $SN_{len} = 20$.

SN	Pruning rate (%)	SST-2		AG-News	
		Testing acc (%)	WM query	Testing acc (%)	WM query
SN-20-1	0	88.58	Lossless	93.57	Lossless
	90	88.96	Lossless	85.49	[11,11,11,11,11,10,11,11,11,11,11,11,11,11,11,11,11,11,11,11]
SN-20-2	0	88.63	Lossless	93.39	Lossless
	90	87.48	Lossless	84.67	[11,11,11,11,11,11,10,11,11,11,11,11,11,11,11,11,11,11,11,11]
SN-20-10	0	88.69	Lossless	93.46	Lossless
	90	88.66	Lossless	86.97	[11,11,11,11,11,11,10,11,11,11,11,11,11,11,11,11,11,11,11,11]
SN-20-20	0	88.80	Lossless	93.49	Lossless
	90	88.41	Lossless	87.26	Lossless

TABLE 9: Testing accuracy on clean models and watermarked models under different BS_{num} .

Model	$BS_{num} = 21$			$BS_{num} = 31$		
	TextCNN (%)	TextRNN (%)	BERT (%)	TextCNN (%)	TextRNN (%)	BERT (%)
Clean	93.88	93.30	94.25	93.88	93.30	94.25
SN-10-1	93.46	92.80	93.83	93.46	92.83	93.91
SN-10-2	93.47	92.97	93.83	93.42	92.80	93.82
SN-10-5	93.59	92.86	94.00	93.39	92.88	93.78
SN-10-10	93.39	93.03	93.75	93.39	92.80	93.80

TABLE 10: The query results of the trigger set after 80 epochs of fine-tuning on AG-News under different BS_{num} .

Model	$BS_{num} = 21$			$BS_{num} = 31$		
	TextCNN	TextRNN	BERT	TextCNN	TextRNN	BERT
SN-10-1	Lossless	Lossless	Lossless	Lossless	Lossless	Lossless
SN-10-2	Lossless	Lossless	Lossless	Lossless	[31,31,31,31,31,31,31,30,30]	Lossless
SN-10-5	Lossless	Lossless	Lossless	Lossless	[31,31,31,30,31,31,31,31,31,31]	Lossless
SN-10-10	Lossless	[21,21,20,21,21,21,21,21,21]	Lossless	Lossless	Lossless	Lossless

TABLE 11: The performance of test set and trigger set under pruning for different BS_{num} .

Model	Pruning rate (%)	$BS_{num} = 21$		$BS_{num} = 31$	
		Testing acc (%)	WM query	Testing acc (%)	WM query
SN-10-1	0	93.46	Lossless	93.46	Lossless
	90	87.41	[21,21,20,21,21, 21,21,21,21,21]	87.57	[31,30,31,31,30, 30,31,30,31,31]
SN-10-2	0	93.47	Lossless	93.42	Lossless
	90	85.75	[21,21,21,20,19, 21,21,21,21,21]	86.20	[31,30,31,30,31, 31,30,31,30,30]
SN-10-5	0	93.59	Lossless	93.39	Lossless
	90	88.66	[21,19,20,21,21, 19,21,20,21,21]	83.67	[31,30,30,29,31, 30,31,29,30,30]
SN-10-10	0	93.39	Lossless	93.39	Lossless
	90	82.95	[21,19,21,21,21, 20,21,21,21,21]	85.70	[30,31,30,31,29, 30,31,31,30,31]

according to the experimental results of $SN_{len} = 10$, the performance of TADW under these SNs basically represents the performance of the entire scheme. We mainly evaluate the fidelity and robustness of TADW under different SNs. "SN-15-1" means $SN_{len} = 15$, contains a 0 (i.e., 111111111111110), and others are similar. Table 6 indicates that, compared with the clean model, the performance loss of the watermarked model on the test set also remains within

0.5% on $SN_{len} = 15$ and $SN_{len} = 20$ for 2 datasets and 3 DNNs. Therefore, SN_{len} has little effect on the performance of the watermarked model. As can be seen from Table 7, the embedded SNs can be successfully extracted after 80 epochs of fine-tuning whether $SN_{len} = 15$ or $SN_{len} = 20$. For parameter pruning, we extract all the preembedded SNs losslessly when $SN_{len} = 15$. Table 8 shows that all SNs can also be successfully extracted when $SN_{len} \leq 20$. Therefore, it

can be concluded that TADW has excellent scalability and can serve a large number of users.

6. Discussion

To measure the impact of BS_{num} on TADW, we set SN as 1111111110, 1111111100, 1111100000, and 0000000000, respectively, and added related experiments with $BS_{num} = 21$ and $BS_{num} = 31$ on AG-News. Similarly, the performance loss of the watermarked models also remains within 0.5% for different BS_{num} (see Table 9). For model fine-tuning, the preallocated SNs all can be extracted losslessly on TextCNN and BERT. For TextRNN, only when $BS_{num} = 31$ has the extraction of SN a slight loss, and the rest are lossless extraction (see Table 10). Table 11 shows that whether $BS_{num} = 21$ or $BS_{num} = 31$, the embedded SNs can still be successfully extracted under model pruning, although the increase of BS_{num} will make the watermarked model slightly more sensitive to the pruning operation. Generally speaking, when $BS_{num} \leq 31$, TADW can successfully verify IP and have remarkable performance.

7. Conclusions

This paper proposes a novel dynamic watermarking framework TADW with the serial number to protect the IP of DNN that can identify unique infringing models and primely conceal trigger set samples applied to query the remote model for IP verification. Again, we innovatively establish a mapping relation between SN and trigger set that leverages the same batch of samples to represent many different SN. We implement TADW on two benchmark datasets of text classification and 6 popular DL models. The experiments indicate that TADW can verify the models' ownership with remarkable robustness and fidelity.

- (I) *More General Watermarking*. As mentioned above, most of the current research on watermarking neural networks is focused on the image field, while other areas such as text and speech are very lacking. Besides, the existing watermarking methods are mainly used in classification tasks, and the research on other tasks such as text generation and image denoising is also lacking. Ideally, generality requires that watermarking algorithm should be independent of the dataset and the DL algorithms used; that is, it can adapt to different scenes (such as image recognition, image denoising, text classification, and text generation). We think that generality is the biggest challenge that watermarking will face in the future.
- (II) *Public Watermarking*. We suppose that, compared with the present concealed watermarking, the future development should be toward public watermarking. Just like coins in various countries, anti-counterfeiting marks are public and cannot be forged. That requires the watermarking to be unforgeable even if it is made public. In conclusion,

the publication of watermarking is helpful to solve the problem of watermark rewriting, and it is also beneficial to combat the model infringement and provide support for the verifiability of watermarking.

Data Availability

The datasets and codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS202123), the Natural Science Basic Research Program of Shaanxi (No. 2021JQ-192), and the Fundamental Research Funds for the Central Universities (No. JB211508).

References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR abs*, vol. 1409, p. 1556, 2015.
- [3] K. He and X. Zhang, "Shaoqing Ren and Jian Sun. "Identity Mappings in Deep Residual Networks," *ArXiv abs/1603.05027*, 2016.
- [4] C. Szegedy, W. Liu, Y. Jia et al., "Vincent vanhoucke and andrew rabinovich. "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, June 2015.
- [5] Y. Goldberg, "A Primer on Neural Network Models for Natural Language Processing," *ArXiv abs/1510.00726*, 2016.
- [6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR abs/1409*, 2015.
- [7] R. Zellers, A. Holtzman, H. Rashkin et al., "Defending against Neural Fake News," 2019. *ArXiv abs/1905*, Article ID 12616.
- [8] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and A. N. Gomez, "Lukasz Kaiser and Illia Polosukhin. "Attention Is All You Need," *ArXiv abs/1706.03762*, 2017.
- [9] A. Graves, Abdel-rahman Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, Vancouver, BC, Canada, May 2013.
- [10] A. Y. Hannun, "Carl case, jared casper, bryan catanzaro, gregory frederick diamos, erich elsen, ryan J. Prenger, sanjeev satheesh, shubho sengupta, adam coates and A. Ng. "Deep speech: scaling up end-to-end speech recognition," *ArXiv abs/1412.5567*, 2014.
- [11] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of

- four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [12] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO, USA, October 2015.
- [13] B. Wang and N. Z. Gong, “Stealing Hyperparameters in Machine Learning,” in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 36–52, San Francisco, CA, USA, May 2018.
- [14] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, “Watermarking digital image and video data. A state-of-the-art overview,” *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.
- [15] P. Singh and R. Singh Chadha, “A survey of digital watermarking techniques, applications and attacks,” *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 9, pp. 165–175, 2013.
- [16] Y. Uchida, Y. Nagai, S. Sakazawa, and S. ichi Satoh, “Embedding watermarks into deep neural networks,” in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, Paris, France, April 2017.
- [17] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and K. Joseph, *Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX Security Symposium, USA, 2018.
- [18] H. Chen, Bitar Darvish Rouhani, C. Fu, J. Zhao, and F. Koushanfar, “DeepMarks: a secure fingerprinting framework for digital rights management of deep learning models,” in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, Ottawa, Canada, June 2019.
- [19] Rouhani, B. Darvish, H. Chen, and F. Koushanfar, “DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models,” ArXiv abs/1804, 2018.
- [20] J. Zhang, Z. Gu, J. Jang et al., “Protecting intellectual property of deep neural networks with watermarking,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, Incheon, South Korea, June 2018.
- [21] Z. Li, C. Hu, Y. Zhang, and S. Guo, “How to prove your model belongs to you: a blind-watermark based framework to protect intellectual property of DNN,” in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019.
- [22] M. M. Yadollahi, F. Shoeleh, S. Dadkhah, A. Ali, and Ghorbani, “Robust black-box watermarking for deep neural network using inverse document frequency,” in *Proceedings of the 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 574–581, Calgary, Canada, October 2021.
- [23] T. Wang and F. Kerschbaum, “Attacks on Digital Watermarks for Deep Neural Networks,” in *Proceedings of the ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2622–2626, Brighton, UK, May 2019.
- [24] L. Fan, K. W. Ng, and C. S. Chan, “Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [25] J. Guo and M. Potkonjak, “Watermarking deep neural networks for embedded systems,” in *Proceedings of the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, San Diego, CA, USA, November 2018.
- [26] R. Namba and J. Sakuma, “Robust watermarking of neural network with exponential weighting,” in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Auckland, New Zealand, July 2019.
- [27] H. Jia, C. A. Choquette-Choo, and N. Papernot, “Entangled Watermarks as a Defense against Model Extraction,” ArXiv abs/2002, Article ID 12200, 2021.
- [28] H. Li, E. Wenger, S. Shan, B. Y. Zhao, and H. Zheng, “Piracy Resistant Watermarks for Deep Neural Networks,” 2019, <https://arxiv.org/abs/1910.01226>.
- [29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2019, <https://arxiv.org/abs/1810.04805>.
- [30] R. Socher, A. Perelygin, J. Wu et al., “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, October 2013.
- [31] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” ArXiv abs/1509, Article ID 01626, 2015.
- [32] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, October 2014.
- [33] P. Liu, X. Qiu, and X. Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning,” ArXiv abs/1605.05101, 2016.
- [34] F. Qi, Y. Chen, M. Li, Z. Liu, and M. Sun, “ONION: A Simple and Effective Defense against Textual Backdoor Attacks,” ArXiv abs/2011, Article ID 10369, 2021.
- [35] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, “Pruning Filters for Efficient ConvNets,” ArXiv abs/1608, Article ID 08710, 2017.