Detect the watermark through the training model

A watermarking scheme to protect numerical classification datasets

Ruonan Li[1]

Supervisor(s): Zekeriya Erkin[1]**, Devris Isler**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 29, 2023

Name of the student: Ruonan Li
Final project course: CSE3000 Research Project
Thesis committee: Zekeriya Erkin, Devris Isler, Petr Kellnhofer

Abstract

Datasets play an important role in machine learning technology. The quality of a machine learning model is highly dependent on the quality of the training dataset. Datasets are of great economic value and should be viewed as intellectual property. To protect the property rights of machine learning training datasets, we can make use of the watermarking technique. In this paper, we propose a dataset watermarking method for numerical datasets. Our method is modified from the radioactive data method, which is proposed for image datasets. Our method can detect if a linear classifier machine learning model has been trained with the watermarked dataset. The experiment results show that we can detect the watermark with more than 99% confidence with only 1% of data being modified. The watermarking method is not robust against data normalization but is robust against column dropping when the dimension of the dataset is high.

1 Introduction

With the development of technology based on big data, such as machine learning and deep learning, datasets are gaining economic value. Since machine learning models are trained with datasets, the performance of a machine learning model is highly dependent on the quality of the training dataset[11]. Collecting datasets can be quite time-consuming and requires a lot of effort, thus datasets should be viewed as a form of intellectual property. How to protect the intellectual property of datasets, especially for those open-sourced datasets becomes a challenge.

Digital watermarking is a well-known technology to protect the intellectual property of digital data by embedding an imperceptible identifier which can be detected by the owner to prove the ownership[8]. Traditional digital watermarking technology is mainly used to protect the intellectual property of multimedia data such as images, audio and video[8]. Over the past few decades, there has been much work on multimedia digital watermarking[9]. In 2002, Agrawal et al. proposed the first method for watermarking in a relational database[1] and then relational database watermarking got the research community's attention.

To date, dataset watermarking is a relatively new topic in research. There are a few dataset watermarking methods proposed[10][7][5]. Li et al. proposed the backdoor watermarking method by adding trigger data into the dataset[7]. Sablayrolles et al. proposed the radioactive data method[10]. Kin et al. proposed a watermarking method for audio classification datasets. These methods can detect if a machine learning model has been trained with a dataset. However, they are all proposed for non-numerical datasets.

In this paper, inspired by the radioactive data method[10], we apply a modified watermarking method on machine learning classification numerical datasets. We test our method's effectiveness and robustness and conclude that with only 1% data modified, we can detect if a linear classifier model has been trained with the watermarked dataset. However, the watermarking method we propose is not robust against data normalization but robust against column dropping when the dimension of the dataset is high.

The paper is organized as follows. Section 2 explains watermarking technique and gives an overview of the related work. Section 3 explains our watermarking method followed by section 4 which presents the experimental setup and the result. Section 5 analyzes the experiment result. In section 6, we talk about the responsible research of our paper. Section 7 is our conclusions and Section 8 talks about the further work that can be done.

2 Background

This section gives background information on watermarking. Then an overview of the related work is given.

2.1 Watermarking

Watermarking is a technique of embedding imperceptible signal called a watermark in the original data. The signal can be detected or verified later as proof of ownership[6]. A watermarking scheme consists of two components: the embedding algorithm and the extraction or detection algorithm[9].

Based on the application of the watermarking technique, there can be a lot of different requirements. However, for any watermarking scheme, in general, there are three requirements: imperceptibility, robustness and security[4].

Imperceptibility can have different definitions for multimedia watermarking and dataset watermarking. For multimedia watermarking like image watermarking, imperceptibility means invisible to humans. A watermarking scheme is imperceptible if humans cannot distinguish the original data from the watermarked data[9]. However, for dataset watermarking, data are fed to machines, not humans. When watermarking machine learning training datasets, we define imperceptibility as the accuracy of the machine learning model should not be significantly degraded after being trained with the watermarked dataset.

Robustness and security are two concepts that are not easy to distinguish. In this paper, we define robustness as the capability to survive normal data operations while security is the capability to survive intentional adversarial attacks such as unauthorized removal or detection[9]. Data preprocessing is usually done before training a machine learning model. Common data preprocessing includes data normalization, dropping columns and dimension reduction etc. We consider them normal data operations.

We should notice that there is a trade-off between imperceptibility and robustness. The more information we embed in the original data, the less imperceptible but the more robust the watermark is.

2.2 Related work

Backdoor watermarking

Li et al. proposed the backdoor watermarking method for protecting open-sourced image datasets[7]. In their method, they add the backdoor trigger data t with target label y_t into the

original dataset. If a model is trained with the watermarked dataset, the probability will be high that the model classifies trigger data as label $y \in y_t$.

Radioactive data

Sablayrolles et al. proposed the radioactive data method for protecting image datasets[10]. In their method, a random isotropic unit vector μ is generated as the watermark carrier. In a convolutional neural network, a feature extractor f first transforms an image x into a feature space $f(x)$ and μ is added into the feature space of images with the same class label by propagating gradients back into the original image.

The classifier of the model trained with the watermarked dataset will have a higher cosine similarity between μ compared to the classifier of the model trained with the original dataset. The watermark detection is based on statistical hypothesis testing. The null hypothesis H_0 is the model is not trained with the watermarked dataset. Because the cosine similarity between two vectors in high dimensional space follows an incomplete distribution[3], we can calculate the $p-$value based on the cosine similarity between the classifier and μ.

Watermarking audio classification datasets

Kim and Lee proposed a watermarking method to protect audio classification datasets used in deep learning. In their method, they convert a subset of the data samples with the target class label c to the time-frequency domain using the short-time Fourier transform and insert a specific shape into the magnitude component. Then they convert these data back to the time domain with the inverse short-time Fourier transform.

The watermark detection is performed by adding a watermark to a subset of the data with the same class label that is not c. If the deep learning model has been trained with the watermarked dataset, it would classify these data as c incorrectly.

3 Watermarking method

In this section, we explain our watermarking method. Our method is proposed for machine learning numerical datasets for classification. The goal is to protect the rights of a dataset by adding a watermark so the owner can detect if a linear classifier machine learning model has been trained with the watermarked dataset.

3.1 Requirements

As illustrated in section 2.1, a watermarking scheme should be imperceptible, robust and secure. We want our method to meet the requirements listed below:

Imperceptibility: after being trained with the watermarked dataset, the performance of the machine learning model should not be degraded significantly. We use accuracy to measure the performance of the machine learning model, and it should not be degraded for more than 5%.

Robustness: our watermarking method should be robust against common data preprocessing before training a machine learning model. Here, we consider data normalization and column dropping.

Security: our watermarking method should be able to survive adversarial attacks like unauthorized watermark removal or detection. As the watermark we embed is generated randomly, it is unlikely to remove or detect it from the dataset without knowing the watermark in advance. However, since we detect the watermark through the machine learning model, the attacker can remove the watermark by attacking the model, and this is beyond the scope of this paper.

3.2 Watermark embedding

Suppose the dataset we need to watermark has n classes of labels. A random isotropic unit vector μ ($|\mu|$=1)is generated and it will be added to the dataset as our watermark. When marking the training dataset, we select a fraction q of data labelled with class i and we add μ in the feature space of these data.

We denote the classifier of the model trained with the original dataset as ω and the classifier of the model trained with the marked dataset as ω^* for class i. After we add μ in the feature space of the dataset, it is very likely that ω^* moves to the direction of μ, thus the cosine similarity between ω^* and μ is higher than the cosine similarity between ω and μ. That is, the cosine similarity between $\omega^* - \omega$ and μ can be very high. In the following section, we explain how to detect the watermark μ by calculating the cosine similarity between $\omega^* - \omega$ and μ.

3.3 Watermark detection

The detection of the watermark is based on a hypothesis test. The null hypothesis H_0 is that the model is not trained with the marked dataset. Hypothesis H_1 is that the model is trained with the marked dataset. The cosine similarity $c(v_1, v_2)$ between two vectors v_1, v_2 in a high dimensional space of dimension d follows an incomplete distribution[3]:

$$P(c(v_1, v_2) \geq \tau) = \frac{B_{1-\tau^2}(\frac{d-1}{2}, \frac{1}{2})}{2B_1(\frac{d-1}{2}, \frac{1}{2})}$$

with

$$B_x(\frac{d-1}{2}, \frac{1}{2}) = \int_0^x \frac{(\sqrt{t})^{d-3}}{\sqrt{1-t}} dt$$

$P(c(v_1, v_2) \geq \tau)$ represents the probability when $c(v_1, v_2)$ is larger or equal to τ.

Based on the cosine similarity between $\omega^* - \omega$ and μ, we can get the $p-value$ of our hypothesis test, which represents the possibility that we observe the result under H_0. The lower it is, the more confident that we have detected the watermark.

4 Experiments and result

4.1 Experimental setup

The experiments are conducted on three numerical classification datasets: Iris, Wine recognition and Breast cancer Wisconsin dataset[2]. Iris dataset has 150 data samples with 4 attributes and 3 different labels. Wine recognition dataset has 178 data samples with 13 attributes and 3 class labels. Breast cancer Wisconsin dataset has 569 data samples with 30 attributes and 2 class labels. Here we assume the datasets have

been standardized, so we perform standardization before embedding the watermark.

In our first experiment, we test the effectiveness of our watermarking method. For each dataset, we mark a fraction q of it, with q = 0.01, 0.02, 0.05, 0.1. Scikit-learn library has been used to train the model.

After testing the effectiveness of our watermarking method, we test the robustness of our method. Specifically, we test how confident we are to detect the watermark after different types of data preprocessing are done on the dataset. In our second experiment, we test the robustness of our method against data normalization. In our third experiment, we test the robustness of our method against column dropping. We test how confident we are to detect the watermark with less than 3% of columns being dropped. Scikit-learn library has been used to perform preprocessing.

We compare the accuracy of the model trained with the original dataset and the watermarked dataset and calculate the $p-$ value of our hypothesis test.

4.2 Result

Table 1: Accuracy and log_{10}(p) for each dataset with q of data modified

		q= 0	0.01	0.02	0.05	0.1
Iris	accuracy	0.97	0.967	0.967	0.9	0.833
	$\log_{10}$(p)	-0.30	-0.625	-0.876	-0.826	-1.07
Wine	accuracy	1	1	1	1	1
	$\log_{10}$(p)	-0.30	-2.2	-2.7	-3.3	-5.0
Breast	accuracy	0.982	0.982	0.982	0.976	0.976
	$\log_{10}$(p)	-0.30	-3.35	-3.45	-4.63	-4.74

Table 2: Cosine similarity between $\omega^* - \omega$ and μ after normalizing the data

	q= 0	0.01	0.02	0.05	0.1
Iris	0	-0.49	-0.49	-0.49	-0.49
Wine	0	-0.15	-0.13	-0.06	0.02
Breast	0	-0.53	-0.51	-0.42	-0.38

Table 3: log_{10}(p) when dropping c columns with q of data modified, "/" means the cosine similarity between $\omega^* - \omega$ and μ is negative

		c = 1	2	3
q = 0.1	Wine	-0.65	-0.48	/
	Breast	-4.73	-4.61	-3.2
		c = 1	2	3
q = 0.05	Wine	/	/	/
	Breast	-1.92	-2.3	-1.78
		c = 1	2	3
q = 0.01	Wine	/	/	/
	Breast	-0.83	-0.62	-0.59

5 Analysis

Since the incomplete distribution is symmetric, $P(c(v_1, v_2) \geq \tau) = P(c(v_1, v_2) \geq -\tau)$. When the cosine similarity is negative, there is a high probability that the machine learning model has not been trained with the watermarked dataset but we can still get a high $p-$value. Thus when the cosine similarity is negative, we only record its value instead of calculating the $p-$value.

In the first experiment, we test the effectiveness of the watermarking method, as Table 1 shows, when zero data is watermarked, the model achieves the highest accuracy and the $p-$value equals 0.5. It means we are 50% confident that the model has been trained with the watermarked dataset. As q grows, more data is watermarked, and our confidence gets higher. The accuracy has not been degraded much even when 10% of data has been modified. We can also see that with only 1% of data being modified, we can detect the watermark with more than 99% confidence. To avoid randomness, we repeat the experiment with different μ and get the result with the same trend.

In our second experiment, we test the robustness of our watermarking method after data normalization. Table 2 shows the cosine similarity between $\omega^* - \omega$ and μ after normalizing each dataset. The negative cosine similarity shows that we cannot detect that the model has been trained with the watermarked dataset. It is reasonable because, after normalization, μ is removed.

In our third experiment, we test the robustness of our watermarking method after dropping n columns with c = 1, 2, 3. As Iris dataset only has 4 columns, dropping any of them can degrade the accuracy of the model significantly, we choose not to do this experiment on Iris dataset. Because after dropping columns, the dimension of the feature space is reduced and so is the dimension of the classifier, we add 0 in the new classifier at the column we dropped to calculate the cosine similarity. For example, for a 4-dimension space, after dropping column 1 (0-indexed), we get a new classifier (a, b, c) and we change this classifier to (a, 0, b, c) so we can calculate the cosine similarity. The result is shown in Table 3, "/" cell means the cosine similarity is negative and there is no need to calculate the $p-$ value. As we can see, the more data watermarked, the more robust the watermark against column dropping. As Breast cancer Wisconsin dataset has a higher dimension, the watermark in a higher-dimensional dataset is more robust against column dropping.

6 Responsible Research

We perform the experiments and record the results with integrity. The experimental setup has been explained as clearly as possible in a way that the experiments can be easily reproduced. All the datasets and code being used in the experiments are open-sourced.

7 Conclusion

In this paper, we proposed a dataset watermarking method modified from radioactive data. With our method, we can detect if a linear classifier machine learning model has been

trained with the watermarked dataset. We test the effectiveness of our method, the result shows that when marking only 1% of data, we can detect our watermark with high confidence (more than 99%). We also test the robustness of our method against data normalization and column dropping. The result shows that, after normalizing the data, we cannot detect our watermark anymore. However, the watermark is robust after dropping a small number of columns when the dimension of the dataset is high. Based on this, we can also get the conclusion that our method works better for high-dimensional datasets.

8 Further work

The proposed watermarking method has some limitations. Firstly, although the proposed watermarking method can detect if a machine learning model has been trained with the watermarked dataset, it only works for linear-classifier or linear-regression models. The method can be improved so that it is effective for non-linear models.

Secondly, it is common to apply data normalization before training a machine learning model but our method is not robust against data normalization and this needs to be improved.

References

[1] Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *Proceedings of 28th International Conference on Very Large Data Bases, VLDB 2002, Hong Kong, August 20-23, 2002*, pages 155–166. Morgan Kaufmann, 2002.

[2] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[3] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael G. Rabbat, and Hervé Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Trans. Big Data*, 4(1):65–77, 2018.

[4] Poonam Kadian, Shiafali M. Arora, and Nidhi Arora. Robust digital watermarking techniques for copyright protection of digital data: A survey. *Wirel. Pers. Commun.*, 118(4):3225–3249, 2021.

[5] Wan Soo Kim and Kyogu Lee. Digital watermarking for protecting audio classification datasets. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 2842–2846. IEEE, 2020.

[6] Gerrit C. Langelaar, Iwan Setyawan, and Reginald L. Lagendijk. Watermarking digital image and video data. A state-of-the-art overview. *IEEE Signal Process. Mag.*, 17(5):20–46, 2000.

[7] Yiming Li, Ziqi Zhang, Jiawang Bai, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Open-sourced dataset protection via backdoor watermarking. *CoRR*, abs/2010.05821, 2020.

[8] Vyaghreswara Rao Namuduri and S. N. Narahari Pandit. Multimedia digital rights protection using watermarking techniques. *Inf. Secur. J. A Glob. Perspect.*, 16(2):93–99, 2007.

[9] Arezou Soltani Panah, Ron G. van Schyndel, Timos K. Sellis, and Elisa Bertino. On the properties of non-media digital watermarking: A review of state of the art techniques. *IEEE Access*, 4:2670–2704, 2016.

[10] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8326–8335. PMLR, 2020.

[11] Buse Gul Atli Tekgul and N. Asokan. On the effectiveness of dataset watermarking. In Andrew H. Sung, Rakesh M. Verma, and Roland H. C. Yap, editors, *IWSPA@CODASPY 2022: Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics, Baltimore, MD, USA, April 27, 2022*, pages 93–99. ACM, 2022.