Image watermarking for Machine Learning datasets

Using SVD based image watermarking techniques to watermark numerical ML datasets.

Palle Maesen[1]

Supervisor(s): Dr. Z. Erkin[1]**, Devriş Işler**[2]

[1]EEMCS, Delft University of Technology, The Netherlands
[2]IMDEA Networks Institute, Universidad Carlos III de Madrid, Spain

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: Palle Maesen
Final project course: CSE3000 Research Project
Thesis committee: Dr. Z. Erkin, Devris Isler, Dr. Petr Kellnhofer

Abstract

The media watermarking technique domain has had the last 30 years to develop itself. The non-media side, however, is a way newer sub-domain [1]. The data-gathering process for machine learning algorithms is a tedious and time consuming task. This becomes worse as the scale of these algorithms increases. Thus, protecting the datasets against illegal use or sale and proving they are intellectual property is useful. In this paper, we answer the question: *How can image watermarking techniques be applied to classification algorithm datasets, without degrading the dataset's quality?* Algorithms that use the Singular Value Decomposition (SVD) of the data are often the basis of other matrix decomposition based Image watermarking techniques. Thus if an SVD based algorithm can be applied to a machine learning dataset then the other matrix decomposition based algorithms can also be applied. This implies that a large part of the much older media targeted watermarking techniques can be applied to the non-media datasets. In this paper we apply the watermarking technique described in [2] to a machine learning dataset. This watermark provides decent imperceptibility and robustness against update, zero-out and insertion attacks but it's held back by its lackluster robustness against deletion attacks. That said, we proved that when an image watermark is found that is impervious against deletion attacks, it can be applied to the machine learning datasets.

1 Introduction

The use of watermarking techniques in media applications has been developed a lot throughout the years, especially the image, video and audio watermarking techniques. This is in contrast to non-media watermarking techniques which is a newer sub-domain [1]. The principle is the same however, a simple data-stream is inserted in the host data [3] to prove ownership [4], to check whether the data has been tampered with [5], to manage copy control [6], etc. Furthermore, images and datasets are quite similar in structure. Both are N-dimensional matrices containing (often numerical) data. This is why we apply the older and more developed techniques to a suitable non-media dataset.

The amount of data needed for the creation of state-of-the-art machine learning models is always increasing as the complexity and size of these models increase [7]. Imagine a company that has put this incredible amount of time into the data gathering process. They then sell this data, only to have it resold by somebody else. This huge time investment is then lost. That is why proving that a certain dataset used to train these models is intellectual property (IP) is so valuable.

We answer whether the current more developed image watermarking techniques can be applied to machine learning algorithm datasets to prove ownership. The research question is: *"How can image watermarking techniques be applied to classification algorithm datasets, without degrading the dataset's quality?"* This is done by applying a basic watermarking technique currently used to watermark images to datasets used in Machine Learning algorithms.

The structure of this paper is as follows, Section 2 gives a background on the general concepts used in the paper. In section 3, we choose a suitable watermarking technique. We explain the algorithm in section 4. Two datasets are watermarked and some metrics are run to check the watermarked dataset's usefulness in section 5. Then we present the watermark's robustness to attacks, like data deletion, insertion and manipulation. Section 6 forms a conclusion whether this is a viable solution and if it can be applied to real-world applications. Section 7 and 8 talk about the responsible research and future work respectively.

2 Background

In this section, we give a short background on the concepts used throughout the paper. These include machine learning, watermarking and the Singular Value Decomposition.

Machine Learning

The relevance of AI in our daily lives is difficult to miss. Classifying images by a search engine, autonomous driving and speech recognition are just a few examples. According to Susmita Ray, Machine Learning involves giving a machine a certain task. The machine is said to have learned, if its measurable performance has improved from experience [8].

The algorithms used in this paper are the Naive Bayes and decision tree algorithms in the supervised category and the k-means clustering in the unsupervised category [9].

Watermarking

The watermarking process contains two steps: the embedding process and the extraction process [10]. During the embedding process a datastream or watermark is inserted into the host data. The extraction process entails extracting the watermark from the host data. There are a lot of different watermarking techniques and classifications. A complete overview is given in figure 1.

Image Watermarking

The most important characteristic of image watermarking is the domain the technique works on.

Spatial Domain Based, These techniques are really easy to compute but lack robustness. Examples of these techniques involve the Least Significant Bit [11], Intermediate Significant Bit [12] and patchwork [13].

Transform Domain Based, These techniques are more difficult to compute but are often designed with robustness in mind. Examples of these techniques are based on the Discrete Cosine Transform [14], Discrete Wavelet Domain [15],

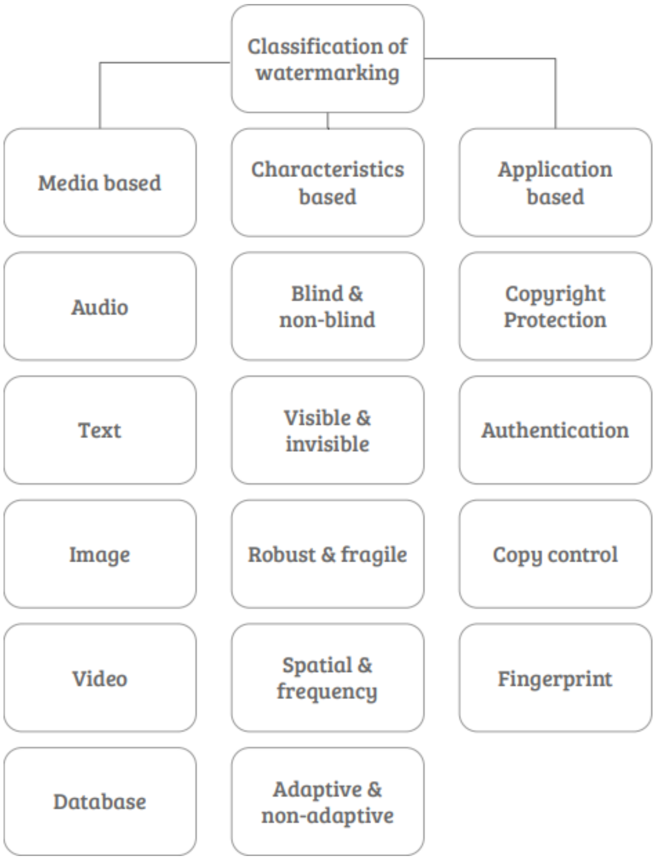


Figure 1: Overview watermarking technique classifications

Singular Value Decomposition [2] and QR Decomposition [16] Domains.

Machine Learning watermarking

Watermarking AI shares most requirements with watermarking requirements for other domains. There is, however, a distinction between white-box and black-box watermarking techniques. White-box watermarking techniques require the actual model for the watermark extraction process. Black-box techniques don't require the model [17].

Singular Value Decomposition

Singular Value Decomposition is an algebraic tool to decompose an original matrix A, into 3 different matrices U,Σ and V.
Where:

- $A = U\Sigma V^T$
- $Order(A) = n$
- $\Sigma = diagonal(\sigma_0, ..., \sigma_n)$
- $\forall\sigma, \sigma \geq 0$
- $\forall\sigma, \sigma_i > \sigma_j$, where $i < j$

The matrices U, Σ, and V all have the same size as matrix A [18].

3 Methodology

In this section, we explain the reasoning behind our method. It explains the methods that were considered and why certain ones were chosen.

Watermarking Technique

Goal of the Technique

Before choosing the technique we need to state what the goal is. The focus of this paper lies in large datasets that require a lot of time to create. Copyright protection entails proving ownership of data by embedding an invisible and inseparable watermark in the host data [19]. This is the most common goal of our technique, as we want to prove ownership of the data when illegitimate copies of our datasets are stolen and/or sold.

Requirements

Based on the goals mentioned in the previous section, the most important requirements are deduced. The requirements we consider include: robustness, imperceptibility, security, computability and capacity. A short description is given for every requirement.

Beginning with the most important one, robustness. The watermark needs to prove ownership of the dataset. When a third party steals it, they want to manipulate the data in a way the watermark cannot be detected anymore. These so-called attacks include, Deletion attacks, Update attacks, Insertion attacks, zero-out attacks and multi-faced attacks [20]. The watermark needs to be resilient against these attacks.

Next the imperceptibility, this determines the usefulness of the data after watermarking. For image watermarking this is easily defined, as our human visual system has clear limitations. Image watermarking techniques use these limitations to add watermarks without degrading the original image quality [19]. As for dataset watermarking, the imperceptibility is defined by the function of the dataset. E.g for datasets used to train nearest mean algorithms, the mean of the different classes needs to be preserved.

The capacity or data payload of the watermark is determined by the amount of data it can hold [21]. Machine learning datasets are in general larger than a single image. The data capacity will therefore be a lot larger as well. It might even be possible to embed the watermark more then once. Doing this increases the robustness of our technique.

Highly secure watermarking techniques make sure that the watermark cannot get leaked in any way. This is applicable when personal information is used to watermark the data. However, this is not relevant when focusing on copyright protection.

Easily computable watermarking techniques are used when many copies of some original data have to be watermarked. As the time it takes for the data gathering far outweighs the time it takes for the watermarking process, is this not relevant to the application in this paper.

The most important requirement is thus the robustness of the watermark, then the imperceptibility. The security is less important as the watermark is often a public image or datastream (e.g. the logo of a company). The capacity is not important as the watermarking technique targets huge datasets, that can contain a lot of information regardless of the capacity.

Conclusion

Based on these requirements a category of techniques is chosen. An invisible robust technique is needed for copyright control of machine learning datasets. According to Mahbuba Begum, watermarking techniques that transform the domain are more robust than spatial domain techniques.[19] The techniques used in this paper can't really take advantage of frequency properties. Thus a singular value decomposition-based technique is chosen. The final choice is the *"SVD-based digital image watermarking scheme"* by Chang et al. [2]

4 Image watermarking algorithm application

This section will go over the implementation of the image watermarking algorithm chosen in the methodology. The Singular Value Decomposition-based embedding method is as follows.

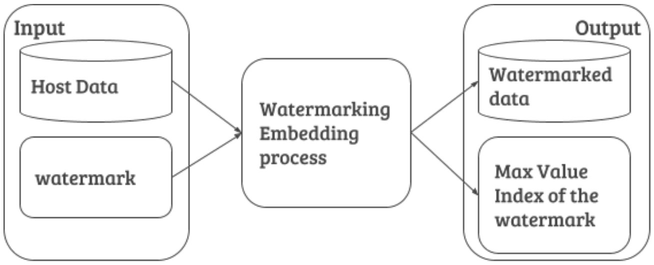

Input
Host Data
watermark
Watermarking Embedding process
Output
Watermarked data
Max Value Index of the watermark

Figure 2: Overview embedding process of the watermarking technique

- Data preprocessing: The host data in [2] is assumed to be a color image with each datapoint being an integer between 0 and 255. The ML data cannot be assumed to have this same property, thus before watermarking the data is normalized by dividing each datapoint with the maximum value.
- Choosing the attribute to be watermarked: Color image watermarking techniques mostly focus on the blue channel. This is done because the human eye is least perceptible to blue color [22]. The same logic can be applied to machine learning datasets. The least important attribute is often the attribute with the lowest variance. Thus the attribute with the smallest variance is chosen to be watermarked.
- Division into blocks: The data is divided into blocks of 4 by 4 elements.
- Embedding of the watermark: The embedding is done by first taking the Singular Value Decomposition of each block. Then the elements in the first column, second and third row are taken from the U matrix in the SVD. The relation between these elements is used to embed the watermark. In case the watermark bit is one, is relation made positive. If it is zero this relation is made negative. A threshold is determined to define how positive or negative this relation may be. This process is seen in figure 3. Here the data is watermarked with threshold of 0.03. The normalized host data is first decomposed into its U Σ and V matrices, the one shown is the U matrix. The relation between the second and third element of the first column (U(2, 1) = 0.44 and U(3, 1) = 0.45) is then checked. This relation is negative as $0.44 - 0.45 < 0$. If the watermark bit matches, in this case, it matches when it is 0, the difference needs to be bigger or equal to the threshold.

  $U(2,1) = ||0.44| - (0.03 - |0.44 - 0.45|)/2| = 0.43$

  $U(3,1) = ||0.45| + (0.03 - |0.44 - 0.45|)/2| = 0.46$

  In case the watermark bit does not match the relation, when the bit is 1, the relation is switched.

  $U(2,1) = ||0.44| + (0.03 + |0.44 + 0.45|)/2| = 0.46$

  $U(3,1) = ||0.45| - (0.03 + |0.44 + 0.45|)/2| = 0.43$
- Reconstruction: Finally, the dataset is reconstructed by multiplying the decomposition of each block and multiplying again with the original maximum value.
- Output: The watermarked data, which attribute was watermarked and the maximum value the data was normalized with.

The final Singular Value Decomposition extraction method is illustrated in figure 4.

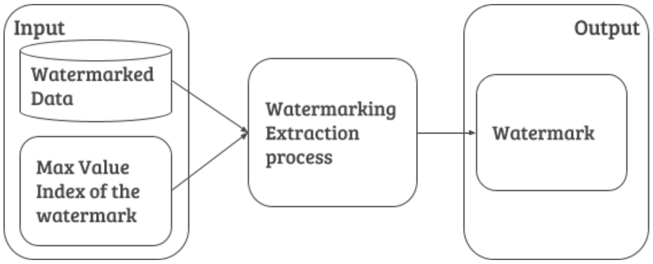
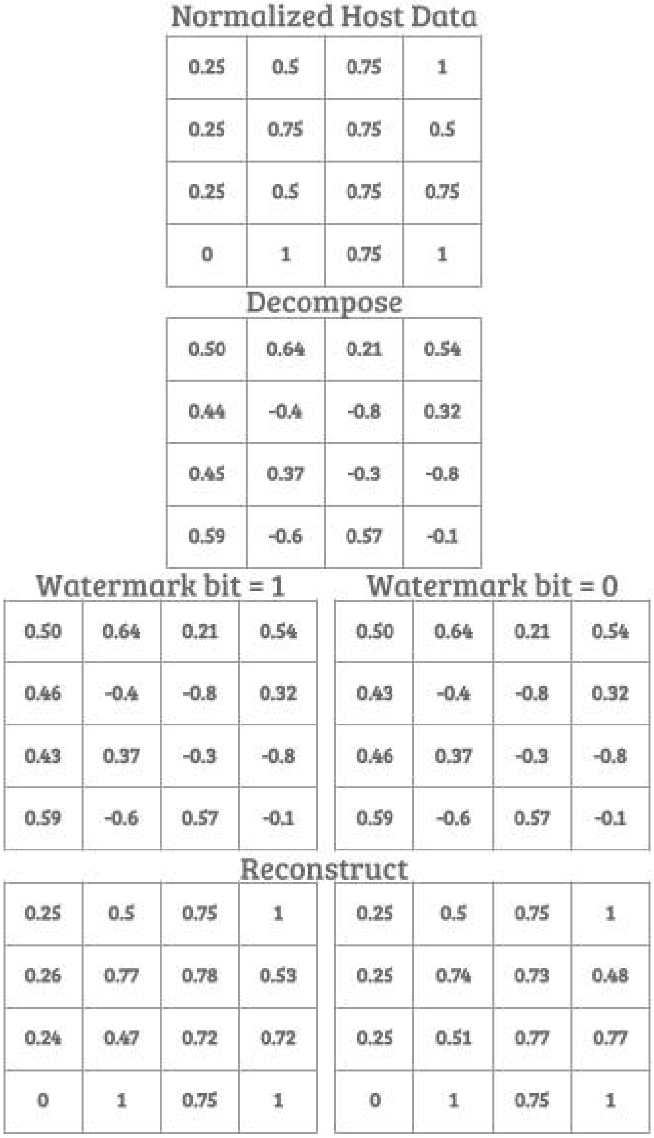

Input
Watermarked Data
Max Value Index of the watermark
Watermarking Extraction process
Output
Watermark

Figure 4: Overview extraction process of the watermarking technique

- Data preprocessing: The watermarked data is divided by the maximum value and again divided into blocks of 4 by 4 elements.
- Extraction of the watermark: Each block is deconstructed in its Singular Value Decomposition and each watermark bit is extracted by comparing the values in the first column, second and third row.
- Output: The reconstructed watermark.



Normalized Host Data

0.25	0.5	0.75	1
0.25	0.75	0.75	0.5
0.25	0.5	0.75	0.75
0	1	0.75	1

Decompose

0.50	0.64	0.21	0.54
0.44	-0.4	-0.8	0.32
0.45	0.37	-0.3	-0.8
0.59	-0.6	0.57	-0.1

Watermark bit = 1

0.50	0.64	0.21	0.54
0.46	-0.4	-0.8	0.32
0.43	0.37	-0.3	-0.8
0.59	-0.6	0.57	-0.1

Watermark bit = 0

0.50	0.64	0.21	0.54
0.43	-0.4	-0.8	0.32
0.46	0.37	-0.3	-0.8
0.59	-0.6	0.57	-0.1

Reconstruct

0.25	0.5	0.75	1
0.26	0.77	0.78	0.53
0.24	0.47	0.72	0.72
0	1	0.75	1

0.25	0.5	0.75	1
0.25	0.74	0.73	0.48
0.25	0.51	0.77	0.77
0	1	0.75	1

Figure 3: Embedding process in a single normalized block of host data

5 Experiments

In this section, we explain how we came to the answer to the research question. We start by explaining the setup. Next, we explain how the watermark was applied. Finally, we show the results of the metrics that test the imperceptibility and robustness of the watermark.

Setup

Two datasets from the UCI archive [23] have been chosen to be watermarked. The iris-dataset and the dry-bean dataset [24]. The datasets, together with a short description, can be seen below. Both the embedding and extraction algorithm is implemented in c++. The python library scikit-learn is used for the machine learning algorithms. The tests themselves, are also implemented in python.

- Iris-dataset, this is a dataset containing 5 attributes from 150 records of iris flowers. The 5 attributes are: sepal length, sepal width, petal length, petal width and species. This is a dataset often used as a beginner dataset for machine learning.
- Dry-bean-dataset, this is a dataset containing around 13000 entries of 7 types of dry beans. Each record has 16 attributes, 12 of these are dimensions and the other 4 are shape forms.

Results

The watermarked dataset is tested on the imperceptibility and robustness of the watermark. The iris-data is watermarked with a random datastream containing 9 bits, the dry-bean-data is watermarked with the binary image given in figure 5 The results are given below.

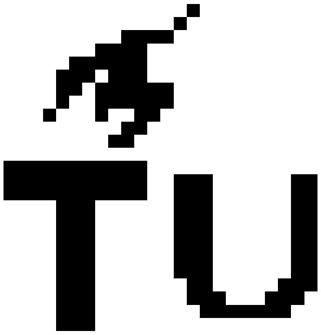


Figure 5: The original watermark

Imperceptibility of Watermark

First, the imperceptibility of the the watermark is tested. Figure 6 shows the iris-data before (left) and after (right) the watermarking process with a random watermark. Here, the sepal width has been used to watermark the data. This attribute has values between 2cm and 4.4cm. A threshold of 0.02 has been used. The x-axis shows the sample number and the y-axis shows the value of the attribute. It shows that the overall form of the data has been maintained, but it has been leveled out. Figure 7 shows the dry-bean data before and after the watermarking embedding. The ShapeFactor4 attribute has been used to watermark the dataset. A threshold of 0.002 has been used. These values go from 0.94 to 1.0.

Four different tests have been selected to analyse the imperceptibility: the Mean Squared Error, the ratio of the variance of the data before and after the watermarking process (FTest), the difference between the means of the kmeans algorithm and the amount of wrong classifications of the Bayes and decision tree algorithms. The tests are executed for increasing thresholds.

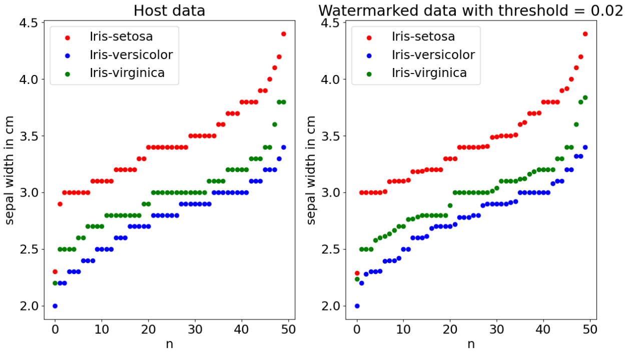


Figure 6: Iris-Data before and after the watermarking process

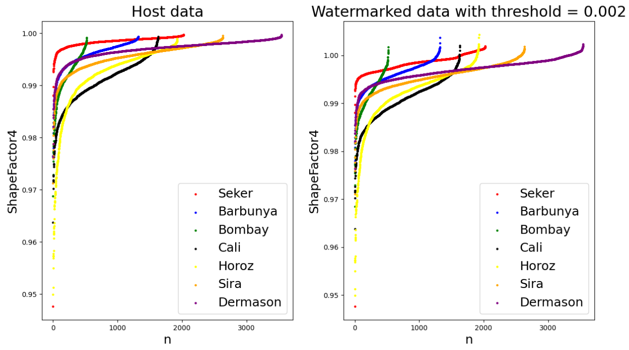


Figure 7: Dry-Bean-Data before and after the watermarking process

Mean Squared Error

Figure 8 and 9 show the mean squared errors between the host datasets and the watermarked datasets on the y-axis and the thresholds used in increasing order. The error increases as the threshold increases.

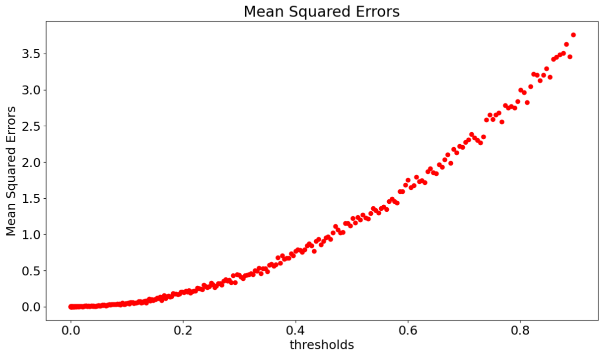


Figure 8: Mean Squared Errors results of Iris-Data

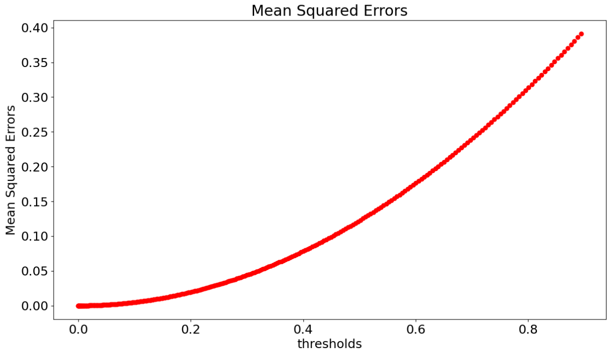


Figure 9: Mean Squared Errors results of Dry-Bean-Data

FTest

Figure 10 and 11 show the ratio between the variances before and after the watermarking process on the y-axis and the thresholds used in the embedding process in increasing order. The variances differ more as the thresholds increase.

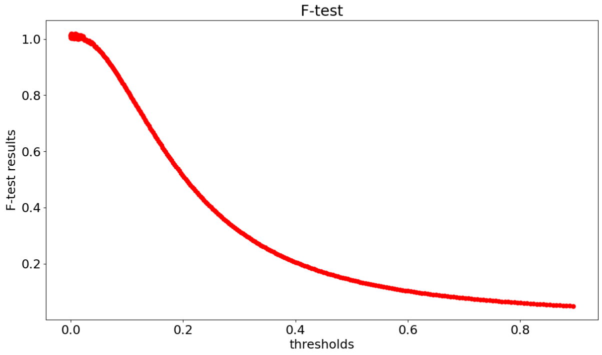


Figure 10: F-Test results of Iris-Data

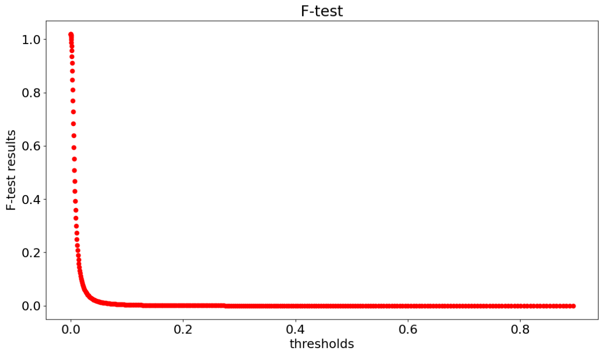


Figure 11: F-Test results of Dry-Bean-Data

Decision Tree Classifier metrics

The dataset is split in training and testing datasets. The dataset was divided at random with 30% of the data going to the testing dataset and the other 70% to the training dataset. Figure 12 and 13 show the accuracy of the decision tree. The original accuracy of the decision tree on the iris-dataset host data was 96%. The original accuracy for the dry-bean dataset was 91%. The overall trend of the accuracy lowers as the threshold gets higher.

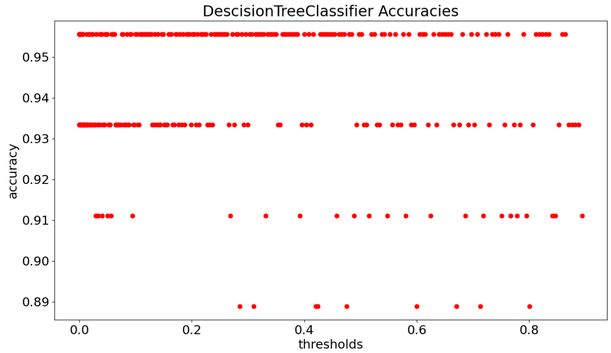


Figure 12: Decision Tree Classifier results of Iris-Data

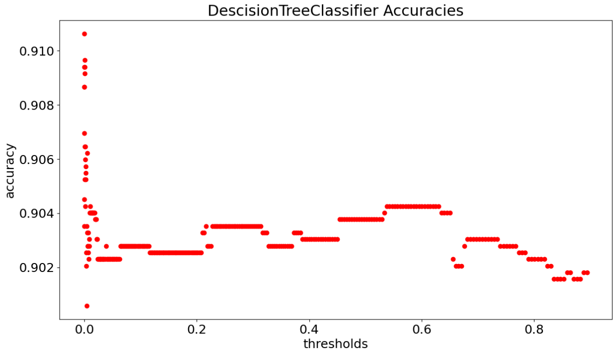


Figure 13: Decision Tree Classifier results of Dry-Bean-Data

Bayes classification Error

Figure 14 and 15 show the accuracies of the BayesClassifier on the Iris and Dry-bean datasets. The original accuracy of the classifier on the Iris and Dry-bean host datasets is 93.3% and 76.2% respectively. The Classifier performances don't noticeably change for both the Iris and Dry-bean dataset.

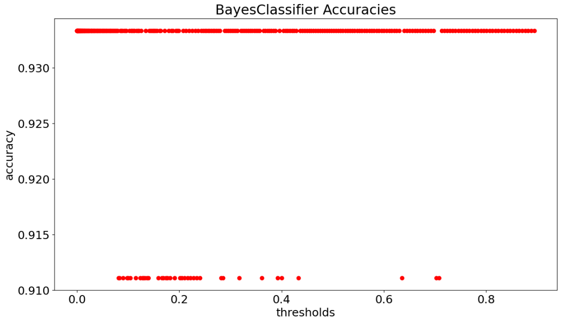


Figure 14: Bayes Classifier results of Iris-Data

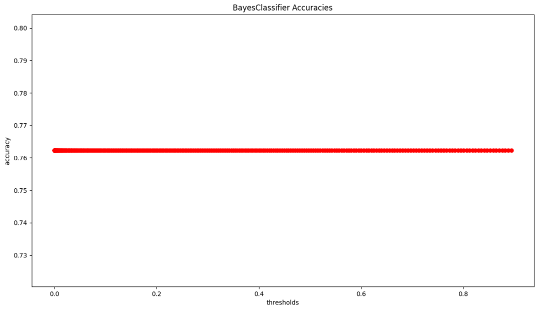


Figure 15: Bayes Classifier results of Dry-Bean-Data

KMeans centroids Error

Figure 16 and 17 show the euclidean distances of the centroids of the kmeans algorithm before and after the watermarking process. The errors increase linearly with the thresholds.

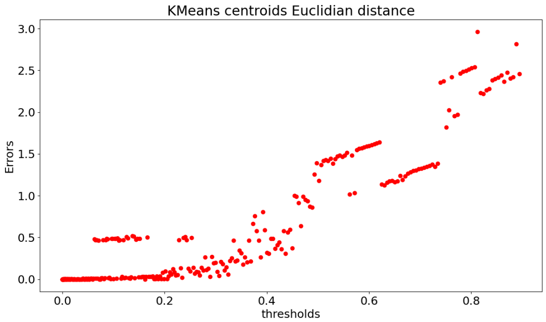


Figure 16: KMeans centroids errors of Iris-Data

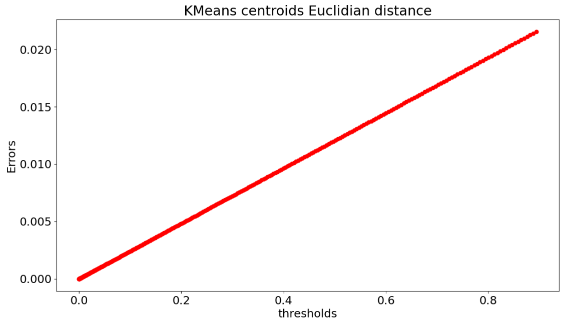


Figure 17: KMeans centroids errors of Dry-Bean-Data

Robustness of Watermark

In this section, we present the results of the robustness attacks. These attacks are performed on the watermarked Drybean dataset with a threshold equal to 0.05. The original watermark can be viewed in figure 5.

Update Attacks

Update attacks try to remove the watermark by adding random noise to the dataset. Figure 18 shows the extracted watermark after adding random noise (±10%) to the dataset, the ratio of the affected data is given above the images. The watermark still persists after adding noise to the entire dataset.

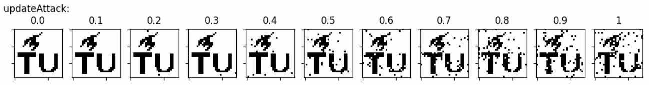


Figure 18: Update Attack on watermarked data with the ratio of data affected

Deletion Attacks

Deletion attacks attempt to remove the watermark by deleting tuples from the dataset. Figure 19 shows the extracted watermarks after an amount of the data has been deleted. The watermark is not recognizable anymore after deleting 1 percent of the data.

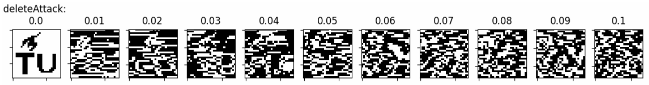


Figure 19: Deletion Attack on watermarked data with the ratio of data affected

Zero-out Attacks

Zero-out attacks involve updating tuples with the value 0. Figure 20 shows the extracted watermarks after an amount of the tuples has been updated with the value 0. After updating 10 percent of the data is the watermark still recognizable, though highly distorted.

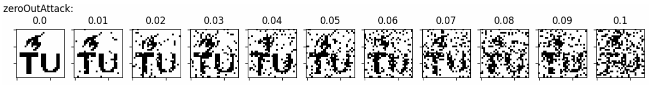


Figure 20: Zero-out Attack on watermarked data with the ratio of data affected

Insertion Attacks

Attackers use insertion attacks to remove the watermark by deleting tuples and adding their own. Doing this to 10 percent of the data highly distorts the watermark.

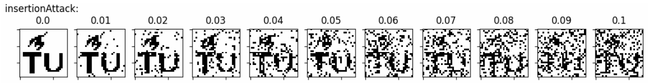


Figure 21: Insertion Attack on watermarked data with the ratio of data affected

Multi-Faced Attacks

The multi-faced attacker model uses all the previous attacks to remove the watermark. After attacking 1 percent of the data is the watermark highly distorted. At 2 percent is the watermark unrecognizable.

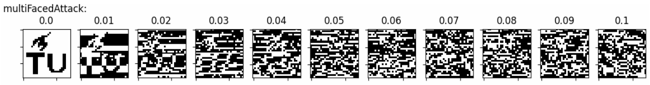


Figure 22: Multi-Faced Attacks on watermarked data with the ratio of data affected

6 Conclusions

The imperceptibility and robustness against any but the deletion attacks of the watermarking technique are sufficient. The results prove that this watermarking technique can be applied to machine learning datasets. This implies that most of the Matrix Decomposition-based algorithms can be applied.

There are properties of images ML dataset targeted techniques cannot take advantage of. The most important one is ordering and format of the data. Deleting a few entries from an image has huge implications and can really quickly destroy an image's quality. Deleting a few out of 13000 entries from a machine learning dataset does barely decrease the quality. This is relevant when looking at the watermark after deleting a part of the data. The entire watermark falls apart after deleting one percent of the data . Even reordering the data, which does not affect the quality of the dataset at all, can completely remove the watermark. Using a robust invisible image watermarking technique which embeds a watermark bit into a single value of the image instead of a block can improve the robustness against deletion attacks.

To answer the research question asked at the beginning of the paper ”How can image watermarking techniques be applied to classification algorithm datasets, without degrading the dataset quality?” The Singular Value Decomposition proposed by Chang et al.[2] is an option as long as certain properties of images are kept: the order of the data and not being able to delete entries. This is almost never the case for machine learning datasets. The inability to deal with deletion or reordering attacks completely nullifies the usefulness of the watermarking technique in real applications.

7 Responsible Research

Progress in Computer Science is often hampered by the lack of reproducibility of the results presented in papers [25]. This is why, to aid in the progress of computer science, this paper has used public datasets and the code is publicly available. Every graph is reproducible by someone who has the code and understands the concepts presented in this paper. The code is available at https://github.com/maanpalle/SVDMLWatermarking.

8 Future Work

In this paper, we have proven that the chosen watermarking technique can be applied to machine learning datasets without degrading the original dataset quality. It is, however, not applicable to real applications due to the reasons stated in the conclusion. The reordering attack can be dealt with easily, e.g by sorting the data before embedding and extracting the data. There is still the problem that deleting some entries completely destroyed the watermark. However, the next step in this research is finding a technique that is impervious to these deletion attacks and applying that to machine learning datasets.

References

[1] A. S. Panah, R. G. van Schyndel, T. K. Sellis, and E. Bertino, "On the properties of non-media digital watermarking: A review of state of the art techniques," *IEEE Access*, vol. 4, pp. 2670–2704, 2016.

[2] C. Chang, P. Tsai, and C. Lin, "Svd-based digital image watermarking scheme," *Pattern Recognit. Lett.*, vol. 26, no. 10, pp. 1577–1586, 2005.

[3] X. Huang and B. Zhang, "Robust detection of transform domain additive watermarks," in *Digital Watermarking, 4th International Workshop, IWDW 2005, Siena, Italy, September 15-17, 2005, Proceedings* (M. Barni, I. J. Cox, T. Kalker, and H. J. Kim, eds.), vol. 3710 of *Lecture Notes in Computer Science*, pp. 124–138, Springer, 2005.

[4] A. Ray and S. Roy, "Recent trends in image watermarking techniques for copyright protection: a survey," *Int. J. Multim. Inf. Retr.*, vol. 9, no. 4, pp. 249–270, 2020.

[5] J. Fridrich, "Image watermarking for tamper detection," in *Proceedings of the 1998 IEEE International Conference on Image Processing, ICIP-98, Chicago, Illinois, USA, October 4-7, 1998*, pp. 404–408, IEEE Computer Society, 1998.

[6] J. A. Bloom, I. J. Cox, T. Kalker, J. M. G. Linnartz, M. L. Miller, and C. B. S. Traw, "Copy protection for DVD video," *Proc. IEEE*, vol. 87, no. 7, pp. 1267–1276, 1999.

[7] W. S. Kim and K. 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*, pp. 2842–2846, IEEE, 2020.

[8] S. Ray, "A quick review of machine learning algorithms," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 35–39, 2019.

[9] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[10] S. Kumar, B. K. Singh, and M. Yadav, "A recent survey on multimedia and database watermarking," *Multim. Tools Appl.*, vol. 79, no. 27-28, pp. 20149–20197, 2020.

[11] A. Bamatraf, R. Ibrahim, and M. N. M. Salleh, "A new digital watermarking algorithm using combination of least significant bit (LSB) and inverse bit," *CoRR*, vol. abs/1111.6727, 2011.

[12] A. Zeki and A. Manaf, "A novel digital watermarking technique based on isb (intermediate significant bit)," *World Academy of Science, Engineering and Technology*, vol. 38, 02 2009.

[13] I. Yeo and H. J. Kim, "Generalized patchwork algorithm for image watermarking," *Multim. Syst.*, vol. 9, no. 3, pp. 261–265, 2003.

[14] Q. Su, G. Wang, S. Jia, X. Zhang, Q. Liu, and X. Liu, "Embedding color image watermark in color image based on two-level DCT," *Signal Image Video Process.*, vol. 9, no. 5, pp. 991–1007, 2015.

[15] H. Agarwal, B. Raman, and I. Venkat, "Blind reliable invisible watermarking method in wavelet domain for face image watermark," *Multim. Tools Appl.*, vol. 74, no. 17, pp. 6897–6935, 2015.

[16] Q. Su, Y. Niu, G. Wang, S. Jia, and J. Yue, "Color image blind watermarking scheme based on QR decomposition," *Signal Process.*, vol. 94, pp. 219–235, 2014.

[17] F. Regazzoni, P. Palmieri, F. Smailbegovic, R. Cammarota, and I. Polian, "Protecting artificial intelligence ips: a survey of watermarking and fingerprinting for machine learning," *CAAI Trans. Intell. Technol.*, vol. 6, no. 2, pp. 180–191, 2021.

[18] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.

[19] M. Begum and M. S. Uddin, "Digital image watermarking techniques: A review," *Inf.*, vol. 11, no. 2, p. 110, 2020.

[20] S. Rani and R. Halder, "Comparative analysis of relational database watermarking techniques: An empirical study," *IEEE Access*, vol. 10, pp. 27970–27989, 2022.

[21] I. J. Cox, M. L. Miller, J.-P. M. Linnartz, and T. Kalker, “A review of watermarking principles and practices 1,” *Digital signal processing for multimedia systems*, pp. 461–485, 2018.

[22] Q. Su and B. Chen, “Robust color image watermarking technique in the spatial domain,” *Soft Comput.*, vol. 22, no. 1, pp. 91–106, 2018.

[23] D. Dua and C. Graff, “UCI machine learning repository,” 2017.

[24] M. KOKLU and I. OZKAN, “Multiclass classification of dry beans using computer vision and machine learning techniques.,” 2020.

[25] T. Y. L. S. R. on Data and C. Sharing, “Reproducible research,” *Computing in Science amp; Engineering*, vol. 12, pp. 8–13, sep 2010.