Watermarking time-series data using DWT

Adapting an existing audio technique to watermark non-medical time series

M. Raave[1]
Supervisor(s): Dr. Z. Erkin[1]**, D. İşler**[2,3]
[1]**EEMCS, Delft University of Technology, The Netherlands**
[2]**IMDEA Networks Institute, Spain**
[3]**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
June 18, 2024

Name of the student: Mike Raave
Final project course: CSE3000 Research Project
Thesis committee: Dr. Zeki Erkin, Devriş İşler, Dr. Asterios Katsifodimos

Abstract

Data security has become more important over the last few years as data sharing over the world has become trivial. Data ownership therefore becomes critical as data can be very valuable and vulnerable to theft. Watermarking is a technique that can help data owners prove ownership over their data. In this paper, an approach is presented to watermark data that is gathered over time, such as weather data. With this research, we propose an adaptation of an existing audio watermarking technique developed in [1]. The adapted algorithm embeds a bit stream into a non-medical time series dataset by calculating the Discrete Wavelet Transform coefficients and modifying their magnitudes. The algorithm shows good robustness against a small range of data modification attacks but lacks capability in larger-scaled attacks. In addition, the proposed algorithm does require additional research to use it in a professional setting.

1 Watermarking

Watermarking is a technique where one can embed pseudo-random noise into a form of data, which should only be able to be extracted by the owner if needed. When the watermark is present, the owner can prove the data belongs to them by verifying their watermark. For any watermarking algorithm, it is important to have imperceptibility, robustness and security [7]. Imperceptibility refers here to the degree that the data is changed because of watermarking, robustness refers to the degree how well a watermark can be extracted after manipulating the watermarked data in various ways and security refers to how easy or difficult the watermarking algorithm is to break.

Over the past decades watermarking has been a highly researched topic as can be seen by the large quantity of publishes throughout the area [10]. It has several applications in multiple fields. Especially in media data such as audio and video, extensive research has been done to find suitable watermarking techniques [8]. However, in non-media data, there is a lack of knowledge of suitable watermarking techniques compared to media instances.

This research aims to contribute to the area of watermarking time series data which is a form of non-media data. Research has been done in watermarking medical time series but not in different applications of time series data. Researching effective ways to watermark non-medical time series is relevant because any form is data can be valuable in these times and therefore it is important that no form of data is neglected in research. This research will introduce a novel algorithm for watermarking non-medical time series data using an audio watermarking algorithm introduced in [1]. The reference algorithm is blind which means that the algorithm does not need the original data to retrieve the watermark. It calculates the 6^{th}-level Discrete Wavelet Transform (DWT) coefficients of the data. For each of these coefficients, it embeds a randomly generated bit. This is done by changing the magnitude of the coefficient to either the closest (n^{th}) or the $(n+1)^{th}$ Fibonacci number in the sequence based on the generated bit and the value of n. More information on the original and adapted algorithm can be found in Sections 2 and 4 respectively.

Watermarking time series data using the DWT has been done already, but to our knowledge, it has only been applied in medical applications, such as in [11]. This algorithm is blind and machine learning (ML) based. The algorithm uses an ML model to retrieve the binary image it uses as a watermark. As the reconstruction of medical signals is the main focus of this algorithm as it is vital in the process of helping diagnose people effectively, this algorithm might not be the best for every other time series application. Therefore, the purpose of this research is to focus on non-medical time series data which is any form of data that is gathered over time that has a non-medical application.

This paper will be structured as follows, in Section 2 relevant papers are discussed that are based on similar research areas. Afterwards, some necessary topics are explained in Section 3 and the developed algorithm is portrayed in Section 4. Section 5 examines the experiments being done to test for robustness and imperceptibility. The ethical implications of this research are considered in Section 6 after which the results are analysed in Section 7. This report discusses future work in Section 8 and concludes with Section 9.

2 Related Work

2.1 Audio Watermarking

Attari et al. proposed an audio watermarking that uses 6^{th}-level DWT coefficients to embed a bit stream as a watermark [1]. They achieve this by dividing the coefficients into frames, assigning a bit per frame and finding the closest Fibonacci number per coefficient to its magnitude. Based on the assigned bit to the frame containing the coefficient and the Fibonacci number, they change the magnitude of the coefficient to either the closest Fibonacci number or one number higher in the Fibonacci sequence. To prove ownership over the data, they extract the watermark by following most of the same steps as with the embedding process. However, per frame, they find the closest Fibonacci number to each coefficient. If more of the found Fibonacci numbers are at even positions in the Fibonacci sequence, the embedded bit in the frame should be a 0, otherwise a 1. As the bit stream is randomly generated, the algorithm is hard to break and

therefore has a high level of security. Attari et al. claimed the algorithm to have high robustness against various attacks. In addition, the algorithm provides a trade-off between imperceptibility and robustness which makes it usable for multiple applications as the frame size of the algorithm can be tuned.

A related algorithm is proposed by Fallahpour et al. in [5]. They propose a similar algorithm which uses the Fast Fourier Transform (FFT) coefficients and gives the possibility to tune parameters, like the frequency band of the signal that is used for embedding the watermark and the frame size, based on the required capacity and robustness. As FFT coefficients do not store information about time, this algorithm is only suited for specific audio applications.

Fallahpour et al. also proposed a related algorithm in [4] in which they propose to use the 2^{nd}-level DWT coefficients and the average value of each frame instead of the Fibonacci sequence to change the magnitude of some of the coefficients. This algorithm is a bit more outdated than the previous two as the parameters to show performance, the bit error rate (BER) and objective differential grade (ODG), are worse compared to the previous two discussed algorithms.

2.2 Medical time series data

Regarding time series data, medical applications have been more popular in research. In this section, a few devised watermarking techniques in this area are discussed. In [11] a technique is developed by Duy et al. that uses an image as a watermark that is scrambled through the Arnold transformation before embedding to improve robustness. The data itself is down-sampled after which the 4^{th}-level DWT approximation coefficients are calculated. Afterwards, the watermark image gets embedded with a bit of the image per data frame. The bit is embedded by modulating the mean relation of the frame based on the value of the bit. For extraction, they use a machine learning model built from part of the watermarked data. The other part is used as testing data to extract the watermark, which makes the algorithm more reliable. This algorithm has high robustness against small forms of cropping, noise addition, filtering and re-sampling attacks and has good imperceptibility such that the watermarked data is still usable. In addition, for watermarks longer than 20 bits, the false positive error rate is about equal to 0.

Furthermore, in [6] a technique is proposed by Gruber et al. to protect personal health data against data leakage. They generate a watermark at the time of requesting data, such that each person gets their personalised watermarked version, which means that it is less prone to data leakage as the watermark contains a link to the person who requested the data. This is done by retrieving the assigned usability constraints to the requested data from a database and computing the probabilities of error sub-ranges based on a Gaussian model. Then by making sure the value of the previous, current and next index are still in the same order as they were before, the watermark is added to the dataset and passed on to the user. This makes sure the structure of the dataset stays intact, such that the watermark is less perceptible. In addition, the method has good robustness against a variety of data modification attacks.

3 Preliminaries

3.1 Discrete Wavelet Transform

The algorithm in this paper uses a technique called the Discrete Wavelet Transform (DWT) to produce a more redundant form of the signal. By applying the DWT to data, the data is split into two parts, approximation and detail. Per the calculated DWT level, the output is stored in the detail coefficients. When the desired level/composition is reached, the resulting coefficients will be the approximation coefficients. By using Formulas 1 and 2 the approximation and detail coefficients can be determined respectively:

$$C^L(i) = \sum_{j=0}^{J_w-1} h(j-2i)C^{L-1}(j) \tag{1}$$

$$D^L(i) = \sum_{j=0}^{J_w-1} g(j-2i)D^{L-1}(j) \tag{2}$$

where L represents the level of the approximation and detail coefficients h(j) and g(j) define the high and low-pass filters of the wavelet respectively and J_w is the length of the filter of the wavelet.

3.2 Fibonacci sequence

The Fibonacci sequence that is being used in the original algorithm for audio watermarking is obtained by Formula (3):

$$F_n = \begin{cases} 0 & \text{if } n \leq 1 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{if } n \geq 1 \end{cases} \tag{3}$$

This sequence has the interesting feature that when the limit is calculated when n approaches infinity as in Formula (4), you get Formula (5) as a result.

$$\lim_{n\to\infty} \frac{F_n}{F_{n-1}} = 1 + \frac{1}{\lim\limits_{n\to\infty} \frac{F_{n-1}}{F_{n-2}}} = \varphi \tag{4}$$

$$\varphi = \frac{1 \pm \sqrt{5}}{2} \tag{5}$$

If φ is positive, then $\varphi \approx 1.618$ which is equal to the Golden Ratio. The Golden Ratio is named after introducing the Golden Rectangle that has sides with the ratio of φ.

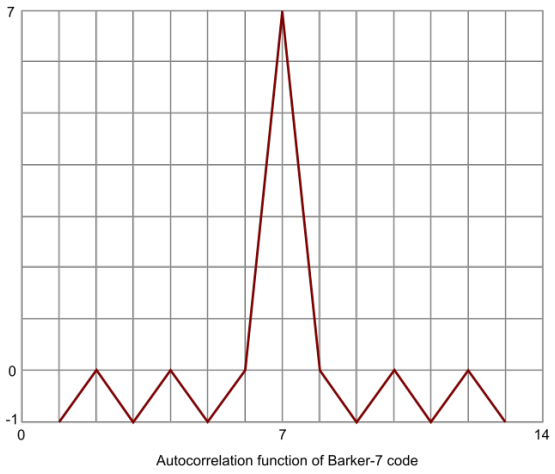

Figure 1: Graph of barker code. By English Wikipedia user Hoemaco, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=5873279.

Since this research is focused on a different form of data compared to the data being used with the original algorithm, experiments have been run with different multiplication rates than the Golden Ratio of the Fibonacci sequence. This will be further elaborated upon in Section 5.

3.3 Barker code

A Barker code is a sequence of bits that are mostly used in telecommunications to synchronize data between the sender and receiver. These sequences can have the length of 2, 3, 4, 5, 7, 11 and 13 and consist of values in -1, 1. The longer sequence one takes, the better the data can be synchronized and errors in data due to distortion can be neglected. Barker codes are generated by having a sliding window of the barker code slide over the barker code itself. The goal is to have one peak in the middle of the resulting graph as shown in Figure 1, when the codes overlap, and the sidelobes of the graphs have a maximum value of 1. It has been proven that there are no other Barker codes with length smaller than 10^{22} [13] [9].

4 Time series watermarking using DWT

This section explains the developed algorithm, which is an adaptation of the algorithm discussed in [1] but suited for time series applications. The input for the developed algorithm is a dataset d, a bit stream as watermark w, a reference set of numbers with a fixed multiplication rate ref and a frame size f.

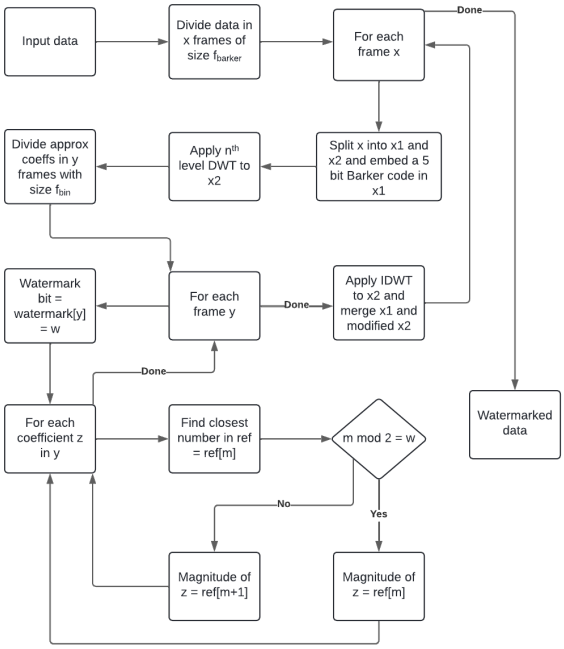

Figure 2: Watermark embedding process.

4.1 Watermark embedding

First of all, the data is split into smaller sections such that the watermark is embedded in multiple places to increase robustness. Each section has a fixed size. Per section, the data is split into 2 lists $n1$ and $n2$ such that $n1 + n2 = n$ as suggested in [12].

In $n1$ 5-bit barker codes are embedded in each sequence to counter cropping attacks. The relative value of a barker code of length 5 has been used as considered in [3] to retrieve the starting position when extracting the watermark. These values are only used for extracting the watermark more effectively.

The watermark itself is embedded in $n2$. First, a 1^{st}-level DWT is applied to the data and the resulting DWT approximation coefficients are divided into frames of size f. Each frame is assigned a bit out of the bit stream by calculating the index of the bit in the bit stream: $l = \lfloor \frac{i}{f} \rfloor + 1$, where i represents the index of the coefficient in the list of approximation coefficients.

For each coefficient c, the n^{th} number in ref is found that is closest to the magnitude of c. To embed the bit, the condition $n \mod 2 = w[l]$ is checked. If this is true the magnitude gets changed to the n^{th} number in ref. If not, it will be changed to the next number in ref. After all coefficients have been considered, the inverse DWT is applied and $n1$ is merged with the modified version of $n2$ to get the watermarked data.

4.2 Watermark extraction

The purpose of the watermark extraction of this algorithm is not to get the original data back, as is the case for many other watermarking algorithms. The purpose here is solely to retrieve the bit stream from the watermarked data and see whether this matches the original bit stream of the owner.

The first steps are similar to the embedding process. However, the first step is to detect the start point of extracting the watermark. The algorithm is most effective where it uses a part of the data that is unchanged for extraction. The detection is the starting point is done by finding the embedded barker codes at the start of each section, which are the added imaginary values to the data as used in [3]. Once those are found, the starting point for extracting the watermark has been set.

Afterwards, the data is split into sections with the same size as with embedding and those sections are each split in $n1$ and $n2$. A 1^{st}-level DWT is applied to $n2$ and the resulting DWT approximation coefficients are divided into frames with size f. Each frame is used for a form of majority voting to find the watermark bit that was embedded. So for each coefficient c in a frame, the n^{th} number is found in ref that is closest to the magnitude of c. Each n connected to each c is stored. If the values of n are more often odd than even per frame, the watermark bit is a 1 and if it is the other way around, it is a 0. Therefore the algorithm always uses odd-sized frames to prevent ambiguity. Once all frames have been considered, the watermark has been extracted from the sequence.

5 Experimental Setup and Results

This section lays out the ways that the algorithm previously described was tested and the results that were obtained.

5.1 Experiment

For experimenting it was decided to focus on imperceptibility and robustness as these metrics have the most relevance in the field of non-medical time series data. The tests have been run separately for imperceptibility and robustness as different metrics have been chosen to define the degree of how well the algorithm was performing. All testing was done with the minimum daily temperatures dataset and the sunspots dataset [2].

For imperceptibility, the following metrics have been chosen: the average of the dataset, the minimum and maximum value of the dataset and the average absolute change of values.

For robustness, the following attacks have been chosen to focus on: cropping, noise addition, scaling and zero-out attacks. These attacks have been chosen since the target area of this research is simple formatted datasets. As a result, these datasets would be most vulnerable to standard data modification attacks as the ones mentioned above.

The program itself has 4 parameters that can be tweaked for personal preference: the frame size of the bins: f_{bin}, the frame size of embedding the barker code: f_{barker}, the DWT level: dwt and the multiplication rate of the values in the reference set: $mult$. In the results, it will be shown how these values will influence the outcome of the algorithm.

Regarding the value of f_{barker}, it was found that the smaller the value is, the more perceptible the watermark will be as a change in DWT values in a small piece of data has a larger effect on the resulting value than with a larger value of f_{barker}. The larger the value is, the less robust the watermark is as a smaller part of the data needs to be modified for the watermark to be irretrievable. All tests have been run with $f_{barker} = 600$, as this provided good imperceptibility and did not have any negative effects on the robustness of the algorithm compared to lower values of f_{barker}.

5.2 Setup

The experiments that have been done for this research have been running on a laptop operating on Windows 11. It contains an Intel(R) Core(TM) i7-9750H CPU at 2.6 GHz with 16 GB RAM. The datasets that have been used for testing are the minimal temperature and the sunspots dataset from [2]. The sunspots dataset contains about 2800 entries and the minimal temperature dataset contains about 3600 entries.

5.3 Results

With the chosen parameters as explained above in mind, the algorithm has been run a hundred times per instance, this took about 2 seconds per instance. Below the measured instances will be discussed.

5.3.1 Imperceptibility

As said previously, the average of the dataset, the minimum and maximum value of the dataset and the average absolute change of values have been chosen to be the metrics to measure imperceptibility in the watermarked data. To measure how imperceptible the watermark could be the program has been run with different parameter values of f_{bin}, dwt and $mult$.
Table 1 shows a few relevant statistics about the datasets used to help understand Table 2, where the results will be shown of the imperceptibility experiment.

Table 1: Statistics of used datasets.

Dataset	Average value	Min value	Max value
Min temp	11.2	0	26.3
Sunspots	51.3	0	253.8

The baseline of the experiment was with the following values: f_{bin} = 11, dwt = 1 and $mult$ = 1.2, as those

Table 2: Imperceptibility results of the test datasets.

Input values			Result values min temperature				Result values sunspots			
f_{bin}	dwt	$mult$	avg diff	min diff	max diff	avg value change	avg diff	min diff	max diff	avg value change
11	1	1.2	0.05	-1.04	2.63	0.91	0.4	-1.49	20.1	4.5
11	3	1.2	0.06	-0.17	1.61	0.90	0.37	-1.87	20.7	4.5
11	5	1.2	0.07	-0.81	0.80	0.87	0.37	-3.71	-3.5	4.6
11	3	1.4	0.11	-0.79	3.38	1.66	0.94	-2.85	18.8	8.4
11	1	1.6	0.38	-1.07	5.55	2.45	2.48	-4.70	57.1	12.8
15	1	1.2	0.05	-1.01	2.37	0.91	0.37	-1.49	17.2	4.55
15	3	1.2	0.06	-0.18	1.62	0.91	0.36	-2.12	20.7	4.56
15	3	1.4	0.09	-0.84	3.45	1.67	1.08	-2.94	14.4	8.38
7	1	1.2	0.05	-1.00	2.15	0.91	0.37	-1.45	22.3	4.57

values gave the most consistently good results regarding watermark extraction and provided a good trade-off between imperceptibility and robustness. The reason the table is not complete is that for some combinations of values, the detection rate of the watermark is not 100%, so this would not be relevant data to include in the results.

From the table, it can be seen that the value of $mult$ creates the largest differences in the dataset. An increase or decrease in the values of f_{bin} and dwt do result in minor changes in the parameter values. However, it is not significant enough to prefer one value over the other.

5.3.2 Robustness

For robustness, the following data modification attacks have been chosen to focus on: cropping, noise addition, scaling and zero-out attacks. In this case, the tests have mainly been run with fixed values of f_{bin}, dwt and $mult$, namely 11, 1 and 1.2 respectively, as those seemed to give the best robustness results with the above-mentioned attacks. These attacks were tested with 10%, 30% and 50% of the dataset being modified. Higher than 50% modification led to a conversion of 50% of the watermark being detected. As this watermarking algorithm only considers zeros and ones as watermarks, it cannot be distinguished from that point whether the algorithm randomly guesses, as the correct guess for a watermark bit is 50% or the algorithm makes more mistakes for a different reason. Below the results per attack will be discussed.

5.3.2.1 Cropping

Regarding cropping attacks, two different types of cropping attacks have been considered. One variant is removing one range of data varying in size and the other is randomly deleting samples. To counter cropping attacks the proposed algorithm uses Barker codes to synchronize the data. The owner of the data can pick a value f_{barker} such that in each frame with a size f_{barker} the watermark will be embedded which makes the likelihood of retrieving the watermark better. For both datasets, which have around 3000 entries, the value of $f_{barker} = 600$ has been picked as a good trade-off between robustness and imperceptibility.

Regarding the attack variant that deletes one range of data, this means that any range up to 65% of the data can be removed and the watermark can still be detected with 100% certainty. In comparison, the original audio watermarking technique used for this research [1] only shows robustness against 200 samples being removed at the start of the signal.

For random sampling, the implementation of the Barker code is less effective. Table 3 shows the results for the datasets that were tested with random sampling compared to the results shown in the medical time series watermarking algorithm by Duy et al. It can be seen from the table below that only 65% of the watermark can be recognized with 10% of the data being randomly removed, which means that it is only slightly better than guessing each bit, which would result in 50% of the watermark being detected on average.

Table 3: Cropping attack results.

Data affected (%)	Detection min temperature (%)	Detection sunspots (%)	Detection medical algorithm (%)[11]
10	65	65	100
30	50	50	-
50	50	50	-

5.3.2.2 Random noise addition

For adding noise to the data, two variants of this attack have been chosen to test with. Either editing the existing values or inserting new entries into the dataset.

First of all, for editing existing values the tests have been run with different percentages of the data being edited. The indexes of these values were randomly sampled and the edited values were restricted to be within the range of 10% of the current value so that it would be less perceptible to others that the data is modified. The results of these tests can be found in Table 4. In the table, it can be seen that for both datasets the watermark can largely be retrieved when 10% of the data is modified. For larger-scaled attacks, the algorithm is more vulnerable.

Table 4: Noise addition attack (Editing) results.

Data affected (%)	Watermark detection min temperature (%)	Watermark detection sunspots (%)
10	90	85
30	55	60
50	50	50

Secondly, for inserting values the tests have been run with different percentages of the data being inserted at locations in the data which were randomly sampled. The inserted values were chosen randomly, but restricted to be within the range of 10% of the values next to it so that it would be less perceptible to others that the data has been changed. The results of these tests can be found in Table 5. In the table, it can be seen that for both datasets the watermark cannot be retrieved reliably anymore. For 10% data modification, the detection rate is higher than when one would guess the watermark bits, but not significantly higher that one could still use the watermarked data in a practical setting.

Table 5: Noise addition attack (Insertion) results.

Data affected (%)	Watermark detection min temperature (%)	Watermark detection sunspots (%)
10	55	60
30	50	50
50	50	50

5.3.2.3 Scaling

A scaling attack entails that the data is being scaled with a certain factor. The results of this attack being successful were fully dependent on the value of $mult$ in the algorithm. The tests have been run with multiple values of $mult$, but 1.6 seemed to give the best results which are given in Table 6 and compared to a medical time series watermark algorithm developed in [11] and the audio algorithm that is used as the main reference for this research, developed in [1].

Table 6: Scaling attack results.

Scaling rate	Detection min temperature (%)	Detection sunspots (%)	Detection medical algorithm (%)[11]	Detection audio algorithm (%)[1]
-30	0	0	-	-
-10	0	0	99	100
10	100	100	99	100
30	100	100	-	-
50	100	100	-	-

In the table, it can be seen that for positive scaling values, the algorithm provides good robustness till 50% positive scaling. For both reference algorithms, there are no results given other than scaling the data values with 10%. When the data is negatively scaled, the algorithm seems to flip all bits which results in 0% of the watermark being recognisable. When testing with lower values of $mult$, the maximal scaling rate which still resulted in 100% detection went down to about 10% positive scaling.

5.3.2.4 Zero-out attack

A zero-out attack entails that a percentage of the data is changed to have a value of zero. Therefore it is similar to a cropping attack, but the dataset does not shrink in size in this case. The indexes of values in the data that were changed were randomly sampled and the experiments have been run with different percentages of the data being modified. Table 7 shows the results of the experiments. In this table, it can be seen that for 10% of the data being modified, the algorithm can still detect about 85% of the watermark. With larger-scaled zero-out attacks, this algorithm proves to be more vulnerable.

Table 7: Zero-out attack results.

Data affected (%)	Watermark detection min temperature (%)	Watermark detection sunspots (%)
10	90	85
30	60	60
50	55	50

6 Responsible Research

In any form of research, it is important to consider the way you handle data and other ethical aspects. In this research, an algorithm has been developed using a Python code base to test and debug the features of the algorithm. This code base can be found online on GitHub at https://github.com/M1ke6/Watermarking for future use. Any researcher can use this code to modify the parameters and find the same and additional results next to what is shown in this paper. Univariate time series datasets have been used to test the algorithm, which were all taken from [2]. Lastly, the use of datasets does bring up privacy and copyright concerns. This has been protected by using open-source datasets that were shared for development purposes.

7 Discussion

There are multiple points to take away from the experiments that were discussed previously. First of all, regarding the imperceptibility tests, it could be concluded that a change in values of f_{bin} and dwt did not give large differences in output. It did seem that for $mult = 1.2$, the results were among the best of the tests. It was expected that $mult = 1.2$ would give the best results as it also gave the most consistently good results regarding watermark extraction overall. However, it was an interesting result that the DWT level would not matter as much for the chosen parameters as most audio watermarking algorithms that use DWT only refer to one specific DWT level that needs to be used. In time series data, a fixed DWT level is according to these results not necessary.

Furthermore, considering the robustness tests, it was found that there was not much data in other papers to compare as there are no results on watermarking non-medical time series previously to our knowledge. In addition, in medical time series or audio watermarking there are many parameters chosen that could not be measured with these datasets. However, the scaling and cropping attacks could be compared to other literature. For scaling attacks, it was interesting to see that the proposed algorithm does flip all bits of the watermark when the scaling rate is below 1, which makes the watermark fully irretrievable. However, from a scaling rate of 1 to 1.5, it achieves a 100% detection rate, which at least matches the reference algorithms with a scaling rate of 1.1 and improves it by having 100% detection up and until a value of 1.5, which the other algorithms show no results for.

Regarding cropping attacks, it could be seen that the embedding process of Barker codes succeeded in countering a form of it, namely where only 1 range of values is removed. A value of $f_{barker} = 600$ has been chosen for good imperceptibility, which means that only 1 block of 600 values in the entire dataset needs to stay intact for the watermark to be 100% retrievable. In comparison, the audio watermarking algorithm in [1] only proves robustness against a block of 200 samples being removed from the start of the audio signal.

Considering the rest of the results of the other attacks, the proposed algorithm had some promising results when only 10% of the data was affected at random indexes, but higher than that, it did not prove to be better than guessing what the watermark sequence is.

In the audio watermarking algorithm by Attari et al., the Fibonacci sequence was being used as a reference set, with $mult = 1.618$ as discussed in Section 3. However, in this research, it has been found that the multiplication rate of the reference set of 1.618 (the Golden Ratio) is too high to use with time series data, even though the algorithm is similar. It did not give much robustness improvements, except against scaling attacks, and it made the watermark more perceptible with a factor of around 3 compared to $mult = 1.2$. A reason for this could be that the data has a different range of size in time series. With audio, there is a large spread among values, but with time series data this could be dependent on the source of the data. The test data that was used did not have a large spread in values as could be seen from Table 1 above.

8 Future work

Even though this research shows some promising results, there are still parts that can be improved in the future or require additional research. First of all, this research has shown robustness against some data modification attacks when 10% of the dataset is affected. We think that the idea of using Barker codes, or something similar can be evaluated further to improve robustness against larger-scale data modification attacks.

In addition, the idea that Attari et al. proposed in [1] to use Fibonacci numbers and the DWT to watermark data could be researched in different areas now that it has been proven to work with time series data. Some other forms of time series can be tested as well, such as multivariate datasets that contain more than 1 variable per row, but this idea could also be researched in the area of numerical datasets for example.

Lastly, one of the main disadvantages of this approach is that it is nearly impossible to retrieve the original dataset after it has been watermarked. Even though it does have its security advantages, it is less practical. In future research, the possibility of developing this feature could be researched to improve the algorithm. A way this can be done is to store additional encrypted values with the watermarked data that let the owner know how much the data has been changed from the original value to the value which is in the reference set.

9 Conclusion

This research proposed a novel technique to watermark non-medical time series. This has been achieved by

adapting an audio watermarking technique developed in [1] to fit the structure of time series data. The algorithm shows robustness against cropping attacks where 1 block of data gets removed and small-scale randomly sampled attacks such as noise addition, scaling and zero-out attacks. The algorithm is more vulnerable to larger-scale attacks, this can be improved in future research. The algorithm shows good imperceptibility depending on what value of $mult$ is chosen for the reference set. The lower the value is, the better the imperceptibility is. This is a small trade-off with the robustness of the algorithm, which makes the algorithm more flexible depending on the application.

References

[1] A. Attari and A. Shirazi. "Robust audio watermarking algorithm based on DWT using Fibonacci numbers". *Multim. Tools Appl.*, 77(19):25607–25627, 2018.

[2] J. Brownlee. "7 time series datasets for Machine Learning", Dec 2020.

[3] C. Campbell. "13 - Coding Techniques Using Linear SAW Transducers". In C. Campbell, editor, *Surface Acoustic Wave Devices and their Signal Processing Applications*, pages 297–327. Academic Press, 1989.

[4] M. Fallahpour and D. Megías. "High capacity audio watermarking using the high frequency band of the wavelet domain". *Multim. Tools Appl.*, 52(2-3):485–498, 2011.

[5] M. Fallahpour and D. Megías. "Robust Audio Watermarking Based on Fibonacci Numbers". In *10th International Conference on Mobile Ad-hoc and Sensor Networks, MSN 2014, Maui, HI, USA, December 19-21, 2014*, pages 343–349. IEEE Computer Society, 2014.

[6] S. Gruber, B. Neumayr, C. Fabianek, E. Gringinger, C. Georg Schütz, and M. Schrefl. "Towards Informed Watermarking of Personal Health Sensor Data for Data Leakage Detection". In X. Zhao, Y. Shi, A. Piva, and H. Kim, editors, *Digital Forensics and Watermarking - 19th International Workshop, IWDW 2020, Melbourne, VIC, Australia, November 25-27, 2020, Revised Selected Papers*, volume 12617 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2020.

[7] P. Kadian, S. M. Arora, and N. Arora. "Robust Digital Watermarking Techniques for Copyright Protection of Digital Data: A Survey". *Wirel. Pers. Commun.*, 118(4):3225–3249, 2021.

[8] S. Kumar, B. Kumar Singh, and M. Yadav. "A Recent Survey on Multimedia and Database Watermarking". *Multim. Tools Appl.*, 79(27-28):20149–20197, 2020.

[9] K. Leung and B. Schmidt. "The Field Descent Method". *Des. Codes Cryptography*, 36(2):171–188, Aug 2005.

[10] A. Soltani 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*, 4:2670–2704, 2016.

[11] T. Duy Pham, D. Tran, and W. Ma. "An intelligent learning-based watermarking scheme for outsourced biomedical time series data". In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 4408–4415. IEEE, 2017.

[12] J. Shen, F. Pan, and Y. Guo. "Digital audio watermark sharing based on the Chinese remainder theorem". In *2012 5th International Congress on Image and Signal Processing*, pages 572–576, 2012.

[13] R. Turyn and J. Storer. "On Binary Sequences". *Proceedings of the American Mathematical Society*, 12(3):394–399, 1961.

A Search keywords

Table 8: Keywords used for literature study.

Search engines	IEEE Explore, ScienceDirect, Google Scholar, Scopus
Keywords	watermarking, audio, fibonacci, time series
Example query	watermarking AND fibonacci AND time series