Applying Wavelet Analysis to S.M.A.R.T. 1 for Hard Drive Classification
by
Logan Childers

A Senior Honors Project Presented to the
Honors College
East Carolina University
In Partial Fulfillment of the
Requirements for
Graduation with Honors
by
Logan Childers
Greenville, NC
December 2018

Approved by:
Dr. Njinasoa Randriampiry - Research Mentor
Mathematics Department - Thomas Harriot College of Arts and Sciences

Abstract

Wavelet analysis is applied to the problem of predicting hard disk failure by examining S.M.A.R.T. stat 1, one of several measures of performance maintained by hard drives. Once the discrete wavelet transform is performed for each hard drive's data, computationally simple measures are taken on the detail coefficients of the transforms. These measures are then examined in aggregate to determine whether S.M.A.R.T. stat 1 possesses any indicator for imminent hard disk failure. The resulting analysis found no factors specific to any metric of the living or dead hard drive populations indicative of hard disk failure.

Contents

1 Background

1.1 Hard Drives, S.M.A.R.T., and Failure Prediction

An open problem in Computer Science is the task of predicting, in general, when a hard disk drive will fail.

Hard disk drives are almost certainly one of the most widespread storage solutions used in modern computing. According to TrendFocus, over 200 million hard drives have shipped per year since the year 2000 [8]. With so much of our storage capacity existing as hard disk storage, and thus so much of our data existing on hard disks, it becomes very important for everyone from the average consumer to large companies and organizations to try and predict when a hard drive will fail, so that important data can be migrated off of the drive before it is lost.

The current built-in monitoring system used on the vast majority of hard disk drives is S.M.A.R.T, which stands for Self-Monitoring, Analysis, and Reporting Technology, entered the public domain in 1995, and has since been implemented in various ways by all major hard drive manufacturers. S.M.A.R.T. works by monitoring several measures of hard drive performance, including frequency of errors, drive hours, and temperature, and then making an assessment of the functionality of the drive, presented as either a failure, meaning the drive will likely not be able to continue functioning in the very near future, or a pass, indicating that the drive has detected no critical errors. Despite the system's stated purpose, however, S.M.A.R.T. has an incredibly low rate of detecting failures before they occur, namely 3-10% detection [6].

Thus, the goal arises of developing a better method of monitoring hard disk health. Many attempts at this have involved trying to use the S.M.A.R.T. metrics, but substituting a more effective algorithm. [9] discusses several of these attempts, with most applying novel statistical methods and machine learning algorithms in order to develop systems with high failure detection rates and low false alarm rates. False alarm rates, the rate at which healthy drives are declared nonfunctional, are an important consideration to manufacturers in particular, as minimization of this measure is the cause of the S.M.A.R.T. algorithm's abysmal detection rate [6]. Detection rates from the aforementioned studies range from 50-80%, with detection rates decreasing in line with false alarm rates. BackBlaze, a cloud storage company, considers five S.M.A.R.T stats in particular (5, 187, 188, 197, and 198) as early indicators of failure, and posts a good success rate of detecting 76% of imminent failures with a 4% false alarm rate [7].

While academic and industrial models have made significant improvements in terms of detection rates, there are still problems which necessitate continued research into drive failure prediction. Using BackBlaze's model in particular, the problem of a high false alarm rate means that their system would likely not see wide implementation from manufacturers themselves, and roughly 24% of drive failures aren't predicted by their S.M.A.R.T. ensemble. For academic models, problems arise from the expense of computing the metrics they employ and in perilous assumptions made by learning schemes that overall endanger the validity of the model [9].

Beyond the models themselves, the S.M.A.R.T. stats they are relying on introduce their own unique problems. The definition and recording of S.M.A.R.T. stats varies significantly by manufacturer, such that many S.M.A.R.T stats are labeled as "Vendor Specific Values".

Additional troubles arise from the fact that most manufacturers do not release information about precisely what their definition entails or appropriate values of their stats, resulting in unclear meanings for many S.M.A.R.T. stats. Finally, another factor that varies by manufacturer is which S.M.A.R.T stats are actually implemented. While these are troubling flaws in any set of S.M.A.R.T. data, the system still represents the most comprehensive and effective monitoring tool available today.

1.2 Wavelets and Wavelet Transforms

Wavelets are, using a basic definition, small wave-like functions. Mathematically, a wavelet must satisfy two properties. The first is that a wavelet must have finite energy:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

Stated differently, the function must be square-integrable, and thus be in the Hilbert space $L^2(\mathbb{R})$. The second requirement is that, given the Fourier transform of the wavelet $\hat{\psi}(f)$, the following admissibility condition must hold in order for inverse transformations to exist:

$$\int_{0}^{\infty} \frac{|\hat{\psi}(f)|^2}{f} df < \infty$$

Additionally, the admissibility condition ensures that the total area under the wavelet is zero, or:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

Any function ψ satisfying these properties is thus a wavelet. Beyond this, other desireable properties can be sought for a wavelet, such as smoothness, vanishing moments, compactness of support, and the satisfaction of an Multiresolution Analysis (MRA). An MRA is a set of properties relating different scales of the wavelet, and thus establishes that a wavelet forms an orthonormal basis over $L^2(\mathbb{R})$ with an accompanying scaling function. A mother wavelet generates a wavelet basis $\psi_{j,k}(t)$ by the parametrization:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{j}} \psi(\frac{t-k}{j})$$

with j representing scale and k representing translation [1].

This study makes use of the Daubechies family of wavelets for the purpose of conducting the discrete wavelet transform on the S.M.A.R.T. data to be analyzed. The specific wavelets used in the analysis of the data were the Haar wavelet (db1), as well as the Daubechies 2 and 4 wavelets (db2 and db4) respectively. The wavelets in this family satisfy a multiresolution analysis and form an orthonormal basis in $L^2(\mathbb{R})$. This orthonormality allows the Daubechies family of wavelets to be utilized in the Fast Wavelet Transform algorithm, which is a particularly efficient algorithm suited to computer-based calculation [5]. Choosing wavelets based on this criteria would allow for the possibility of useful results of this study being implemented in a relatively simple algorithm at a later date.

The discrete and continuous wavelet transforms are methods by which a signal can be converted into another form for the sake of observation. The continuous wavelet transform $T(j,k)$ is defined as:

$$T(j,k) = \int_{-\infty}^{\infty} x(t)\psi^{*}_{j,k}(t)dt$$

with the ψ^{*} representing the complex conjugate of the wavelet utilized [1]. The discrete wavelet transform is then a discretization of the continuous wavelet transform, defined as:

$$x(t) = \sum_{j,k} a_{j,k}\psi_{j,k}(t)$$

where the $x(t)$ now represents a discrete time signal, the wavelet $\psi_{j,k}(t)$ is discrete, and the collection of points $a_{j,k}$ is the result of the discrete wavelet transform, rather than a collection of curves [4]. From a practical standpoint, the discrete wavelet transform is implemented as an algorithm used to decompose a discrete time signal into multiple levels of detail coefficients as well as a set of approximation coefficients. The implementation of this transform utilized for this study is the Wavelet Toolbox found in MATLAB. Figure 1 is an example of the discrete wavelet transform performed on sample S.M.A.R.T. 1 data.

The resulting display features the detail coefficients at each level $(d_1, d_2, ...)$, an approximation a_5, and the original signal for which the discrete wavelet transform was calculated. Each visual, rather than displaying the sequence of coefficients generated at each level, is instead displaying the inverse discrete wavelet transform with every level of coefficients except the level of interest set to zero. Thus, the graph displayed for d_5 is the result of the inverse transform conducted on the whole decomposition, but with d_1 through d_4 and a_5 set to zero. In this way, the detail coefficients can be associated with particular intervals of time, and the fantastic time-resolution of the wavelet transform utilized.

Wavelet Transforms, in particular discrete wavelet transforms, have seen wide ranging applications across a variety of subjects for a variety of purposes. In industrial settings, the discrete wavelet transform has found significant usage in classification of machine states. In [3], Berger et. al. used the biorthogonal 6,8 wavelet to calculate the discrete wavelet transform of cutting state data for an orthogonal cutting tool. Using computationally simple measurements based on the detail coefficients of the transforms, two metrics were located that indicated the transition between chatter and non-chatter states. In [10], Zanardelli developed an algorithm based on the discrete wavelet transform that was able to successfully detect and classify engine failures of particular types.

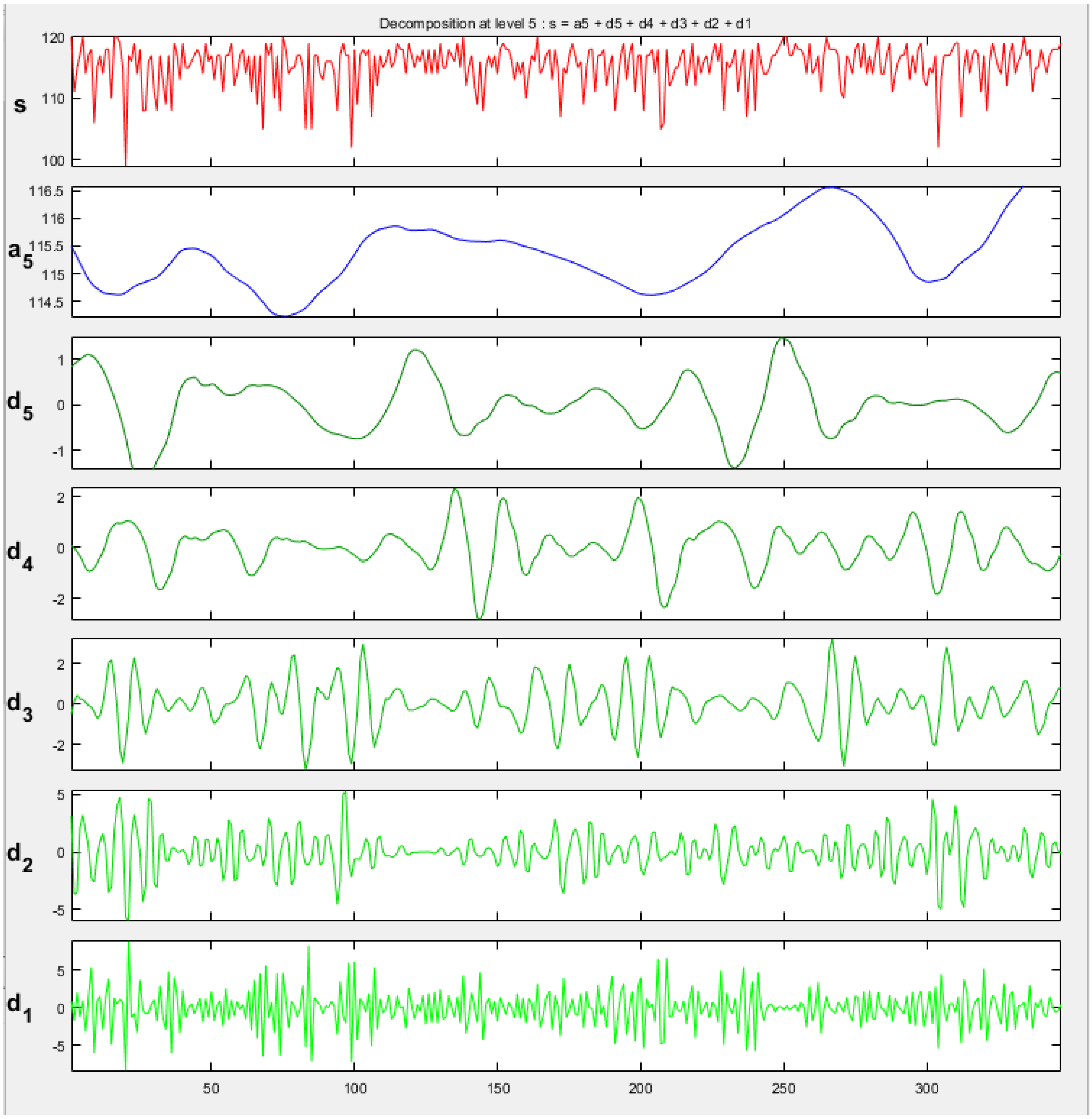

Figure 1: Discrete Wavelet Transform performed on S.M.A.R.T. 1 (normalized): “Seek Error Rate” for serial no. Z300ZV0V

2 Purpose and Research Questions

While wavelets have found a significant expanse of applications across a wide variety of subjects, wavelet analysis has not yet been applied as a possible analytic tool in predicting hard drive failure. Further, the existence of the Fast Wavelet Transform and its recursive implementation provides an interesting opportunity to develop a predictive method that requires relatively little computing power. Given the apparent novelty of this approach, this study asks the following question:

With an eye towards computational simplicity, is it possible to locate a metric based on the Discrete Wavelet Transform of S.M.A.R.T. 1 that can classify a hard drive as functional or failed.

3 Experimental Design

3.1 Data Set and S.M.A.R.T.

The hard disk data examined comes from a subset of BackBlaze's S.M.A.R.T. records for their large drive population. The data set (found at [2]) captures daily records of S.M.A.R.T. readings using Smartmontools on their collection of over 100,000 running drives as of 2018. The data set is available to the public for reuse given credit is provided to BackBlaze as the origin of the data. The specific data used for the study is the collection of all data from 2017. From this data, a sampling of failed and live drives was taken. The drives analyzed were composed of 111 failed drives and 34 living drives, most of which were manufactured by Seagate, due to a significant proportion of BackBlaze's drive population representing Seagate drives.

Originally, S.M.A.R.T. stats 1, 2, 5, 7, 187, 188, 192, 193, 197, 198, 241, and 242 were being considered for analysis, due to those stats having both comprehensive reporting across most drives and interesting possibilities for prediction. Upon examination of the combined data for each drive, it was found that all stats except S.M.A.R.T. 1 were not amenable to analysis by wavelet transform. In BackBlaze's predictors (5, 187, 188, 197, 198), most of the data prior to failure, and indeed most all of the data recorded for all drives, had zero values. Since BackBlaze uses nonzero values of these stats as indicators of imminent failure, this makes sense. Figure 2 illustrates a typical signal representing one of the aforementioned BackBlaze predictors.

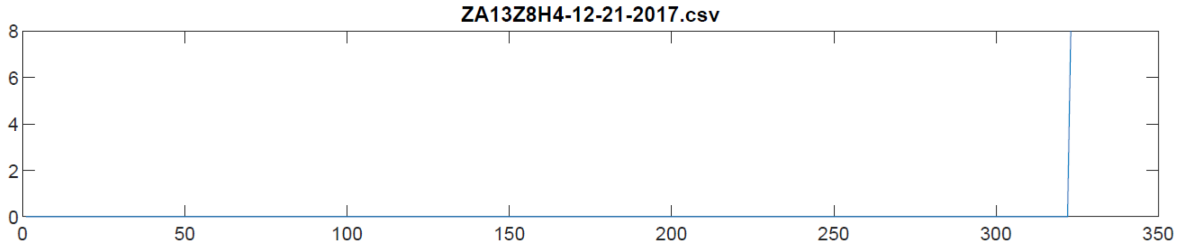

Figure 2: S.M.A.R.T. 5 (raw): "Reallocated Sectors Count" for serial no. ZA13Z8H4

It was determined that analysis of this type of data would not be reasonable, as no reasonable wavelet-based indicator can be derived from an exceedingly long string of zero values, followed solely by ascending behavior. The remaining S.M.A.R.T. stats besides S.M.A.R.T 1 largely displayed ascending behavior, such that any useful trends would likely be derived by a method other than the wavelet transform. A typical example is given in Figure 3.

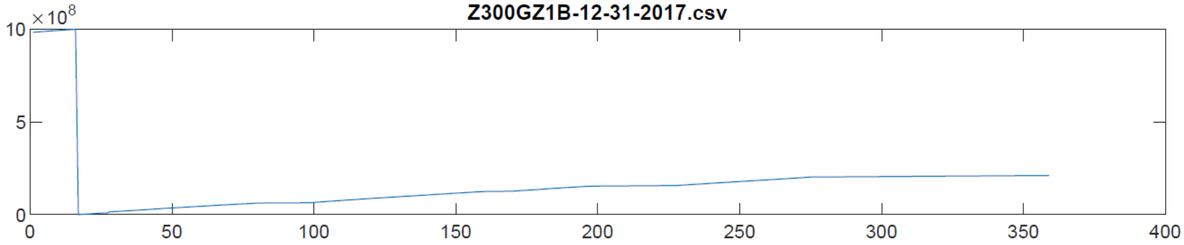

Figure 3: S.M.A.R.T. 7 (raw): "Seek Error Rate" for serial no. Z300GZ1B

Thus, S.M.A.R.T. 1 was chosen as a good experimental stat for its prevalent representation in the data set, its nontrivial behavior, and additionally based on the assertion that the what S.M.A.R.T. 1 measures (Seek Error Rate) could hypothetically be a predictor of impending failure.

3.2 Data Processing/Measurement

S.M.A.R.T. data for initial observation was collected by using csvkit within the Anaconda python environment for Windows. The BackBlaze data is collected in compressed zip files by day of measurement, each file containing one spreadsheet for each drive measured. Retrieving a sample of the data involved merging all of the documents together, then employing the csvgrep command to retrieve a list of all failed drives, followed by the all data related to the serial numbers of specific failed drives, and then all data related to the serial numbers of a random choice of living drives.

Once a sampling of the data was excised into a form that could be processed, the data was loaded into MATLAB where initial plots (similar to Figure 1) were created for the aforementioned candidate S.M.A.R.T. stats. Upon analyzing these plots, S.M.A.R.T. 1 was chosen to conduct further analysis on. Further, all drives with less than 200 days of measurements were removed, leaving 111 living drives and 34 failed drives in the sample to examine.

The discrete wavelet transform was then performed on the data using commands available in the Wavelet Toolbox, and the resulting detail and approximation coefficients stored with the correspond data they were derived from.

From here, 20880 outputs were combined into 288 aggregate data plots, which were created using various grouping parameters. These parameters were:

- Length of Time: Lengths of time transformed were either 30 or 60 days.
- Time Before Failure: All ranges of data were transformed at either 2 days before failure or 5 days before failure.

- Wavelet: All ranges considered were transformed using the Haar Wavelet (db1), the db2 wavelet, and the db4 wavelet in separate instances.

- Type of Value: Hard drives report S.M.A.R.T. values as both raw values, which are the absolute measurements taken, and normalized values, which are adjusted to a vendor specific scale that is often undisclosed.

- Failure State: Drives were split into groupings of living and failed drives for each category.

Within all combinations of these parameters, the specific metrics used for analysis were calculated. The metrics calculated based on the detail coefficients of the transform are the average, standard deviation, and mean absolute deviation (m.a), which is defined as:

$$m.a. = \frac{1}{n} \sum_{i} |x_i - \mu|$$

as well as the ratios of these measures between detail levels of the same drive. As in [3], these metrics were calculated over intervals of the time series data, the interval used here being 5 days of data or 5 data points. Measures were taken using the MATLAB functions mean, std, and mad for the average, standard deviation, and mean absolute deviation respectively.

Once all metrics were taken and stored with the drives they were derived from, they were printed individually using MATLAB's native saveas function and appended into the appropriate categories. Code snippets are available in the Appendix for several of the operations described above. An example page from a result plot is given in Figure 4 below.

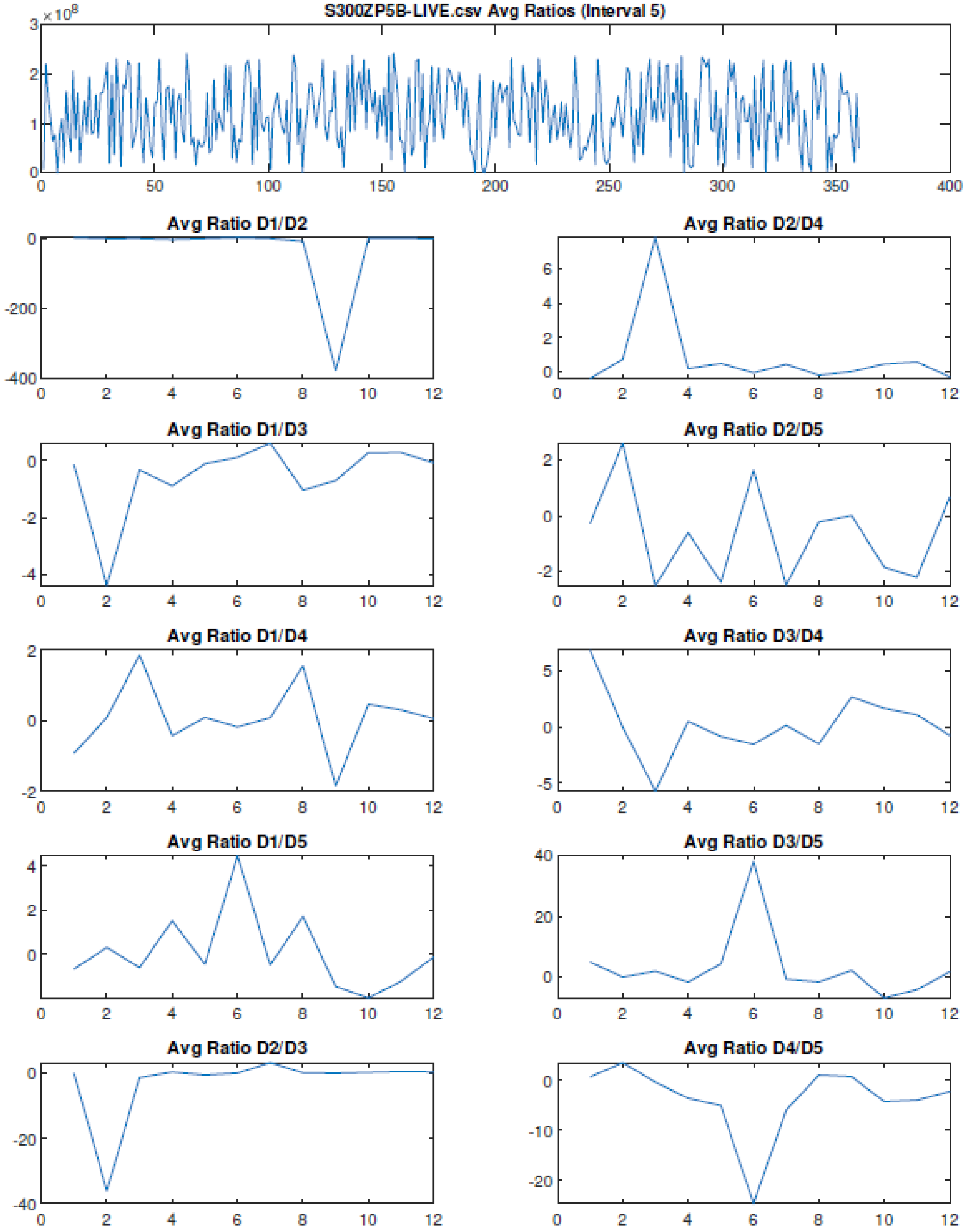

Figure 4: Average Ratios between detail coefficient levels for serial no. S300ZP5B

4 Analysis and Results

Once the data plots were generated, categories of measurement were compared by examining the living and failed plot collections of the same metric, searching for similarities in the range of values, the slope of values, and in the progression of the values over time that was common to only the living or dead drives, and that appeared consistently. A range of values consistently displayed in the dead drives would have allowed for the possibility of a threshold to be set upon one of the metrics beyond which a drive could likely be considered dead. Instead, the range of values occupied by the living and dead drive populations was both varied by drive and consistent in displaying this variance in both populations. A particular slope of the values, primarily at the end of the life of a drive, would have also been indicative of a possible predictive metric. Instead, behavior towards end of life varied widely in both populations in all metrics, having a large range of positive and negative slopes. The values approached by all metrics fell into two cases, with both populations approaching uniformly the same measure, as with the plots relating to standard deviation, or neither population approaching a consistent value for a given metric, as with measures like the averages. A particular progression of data, and it's respective consistent presence in the data plots, would indicate a pathological characteristic in the data that the wavelet transform was able to successfully detect, and which could possibly be used in prediction. Instead, no uniform progression of the data for any metric in the wavelet decomposition, nor based on the decomposition, was able to produce any such anomaly that was unique to either living or dead drives or consistent across on of those populations. The data resulting from the wavelet transforms conducted on S.M.A.R.T. 1, in the end, did not reveal any measure in which the two specifications of confinement within a population and consistency across the population were met.

5 Reflection and Conclusion

While no positive results were found with respect to a wavelet-based predictive factor on S.M.A.R.T. 1, this does not necessarily indicate that there is indeed no such predictive factor available. It is possible that another wavelet might be more suited to this data and would yield a more revealing wavelet transformation, or that more complex measures of the amplitude of the transform coefficients may be necessary in order to measure any useful behaviors hidden in S.M.A.R.T. 1.

Further, when considering the data utilized in the study, it is concerning that the data set is exceedingly sparse in collecting measurements when compared to [3], which conducted these measurements on a considerably denser set of data. While this study could have conducted the wavelet transform over much longer spans of the data, it was determined that this would negate the usefulness of the results, even if something were found, due to the massive amount of time (at least 200 days), that would need to be collected for a given hard disk drive before an effective prediction could be performed. A decision was also made to not take aggregate measures on a shorter interval than 5 days because that would likely do very little to craft a novel view of the data relative to the basic plots of the detail coefficients themselves. A remedy to this possible confound would then be a significantly denser collection of measurements over of similar period of time. For example, hourly measurements taken over 15 days would yield a data set with twelve times the density of data, and would likely be significantly more useful in analyzing data near failure.

While most of the drives analyzed in this study were manufactured by Seagate, the variation in models of drive could also have been a possible confound in comparing living and failed drives in aggregate. A study could be done which only considers the same model of drive when examining S.M.A.R.T. and S.M.A.R.T.-based metrics, but this would have significant consequences for the generalizability of results, as this procedure could locate results which are an artifact of that particular drive model, rather than commonalities that are a legitimate consequence of the drive having failed recently.

It may also be possible that no predictive metric was found because S.M.A.R.T. 1 does not possess any predictive power with regards to determining if a drive will imminently fail. Due to this, it is best to consider, for future studies, a wider range of S.M.A.R.T. values, as well as less stringent requirements on prevalence in a heterogeneous group of drives. With the latter requirement relaxed, there are likely several S.M.A.R.T. stats common within drives of the same manufacturer that should be duly investigated, both from the standpoint of wavelet analysis and using other analytic tools.

6 References

[1] Addison, P. S. *The illustrated wavelet transform handbook: Introductory theory and applications in science, engineering, medicine and finance.*, (2002), Bristol: Institute of Physics Publishing

[2] BackBlaze *BackBlaze Hard Drive Stats*, Retrieved December 2, 2018, from https://www.backblaze.com/b2/hard-drive-test-data.html

[3] Berger, B. S., Minis, I., Harley, J., Rokni, M., and Papadopoulos, M., *Wavelet based cutting state identification*, Journal of Sound and Vibration (1998), 213(5), 813-827. doi:10.1006/jsvi.1997.1495

[4] Burrus, C. S., Gopinath, R. A., and Guo, H. *Introduction to wavelets and wavelet transforms: A primer.* (1998), Upper Saddle River, N.J: Prentice Hall.

[5] Debnath, L. *Wavelet transforms and their applications*, (2002) Boston: Birkhauser.

[6] Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado., *Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application*, J. Mach. Learn. Res. 6 (December 2005), 783-816

[7] Klein, A. *What SMART Hard Disk Errors Actually Tell Us*, Retrieved December 1, 2018, from https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/

[8] TrendFocus, and Statista *Global shipments hard disk drives (HDD) 1976-2020 — Statistic*, (2018), Retrieved December 1, 2018, from https://www.statista.com/statistics/398951/global-shipment-figures-for-hard-disk-drives/

[9] Wang, Y., Ma, E. W. M., Chow, T. W. S., and Tsui, K. *A two-step parametric method for failure prediction in hard disk drives*, (2014), IEEE Transactions on Industrial Informatics, 10(1), 419-430. doi:10.1109/TII.2013.2264060

[10] Zanardelli, W. G. *The use of wavelet analysis for the prognosis of failures in electric motors*, (2000)

7 Appendix: Code Listings

This code is used to read compiled csv files into MATLAB

```
% Specify the folder where the files live.
myFolder = 'C:\Users\Logward123\Desktop\Heartbeat\The␣Repair\2017␣Combined\';
if ~isdir(myFolder)
  errorMessage = sprintf('Error:␣Bad␣directory:\n%s', myFolder);
  uiwait(warndlg(errorMessage));
  return;
end
% Get a list of all csv files in the folder.
filePattern = fullfile(myFolder, '*.csv');
theFiles = dir(filePattern);
for k = 1 : length(theFiles)
  baseFileName = theFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName);
  theFiles(k).data = readtable(fullFileName);
end
```

This code takes the structure generated by reading in the data and appends the detail and approximation coefficients of the discrete wavelet transform at 5 levels to the row the transform was conducted on, then renames the structure. The wavelet used in this transform is db1: the Haar Wavelet.

```
for k = 1 : length(theFiles)

        s = table2array(theFiles(k).data(:,[7]));
        l_s = length(s);

        if(~isnan(s))

        [C,L] = wavedec(s,1,'db1');
        theFiles(k).cA1 = appcoef(C,L,'db1',1);
        theFiles(k).A1 = wrcoef('a',C,L,'db1',1);
        [C,L] = wavedec(s,2,'db1');
        theFiles(k).cA2 = appcoef(C,L,'db1',2);
        theFiles(k).A2 = wrcoef('a',C,L,'db1',2);
        [C,L] = wavedec(s,3,'db1');
        theFiles(k).cA3 = appcoef(C,L,'db1',3);
        theFiles(k).A3 = wrcoef('a',C,L,'db1',3);
        [C,L] = wavedec(s,4,'db1');
        theFiles(k).cA4 = appcoef(C,L,'db1',4);
        theFiles(k).A4 = wrcoef('a',C,L,'db1',4);
        [C,L] = wavedec(s,5,'db1');
        theFiles(k).cA5 = appcoef(C,L,'db1',5);
        theFiles(k).A5 = wrcoef('a',C,L,'db1',5);
        theFiles(k).Details = C;
        theFiles(k).D1 = wrcoef('d',C,L,'db1',1);
        theFiles(k).D2 = wrcoef('d',C,L,'db1',2);
        theFiles(k).D3 = wrcoef('d',C,L,'db1',3);
        theFiles(k).D4 = wrcoef('d',C,L,'db1',4);
        theFiles(k).D5 = wrcoef('d',C,L,'db1',5);

end
theFiles1 = theFiles;
```

This code works on a structure containing several structures of the same type as “theFiles”. Here, sstat 6 and 7 represent the column of theFiles holding S.M.A.R.T. 1’s normalized and raw values respectively. A version of this code was written for each category to be generated for observation, as described above. This code generates the db2 discrete wavelet transform coefficients for a timespan of 60 days which ends 2 days before the the failure of the dead drives, and does so for the normalized and raw S.M.A.R.T. 1 stats.

```
c = 1;
ModdedFiles(1).transforms = theFiles;
ModdedFiles(1).stat = 0;
ModdedFiles(1).ver = 'ver';

for sstat = [7,6]


  for k = 1 : length(theFiles)

    s0 = table2array(theFiles(k).data(:,[sstat]));
    cut1 = length(s0)-61;
    cut2 = cut1 + 59;
    s1 = s0(cut1:cut2);
    s = s1;
    l_s = length(s);


    if(~isnan(s))

      [C,L] = wavedec(s,1,'db2');
      theFiles(k).cA1 = appcoef(C,L,'db2',1);
      theFiles(k).A1 = wrcoef('a',C,L,'db2',1);
      [C,L] = wavedec(s,2,'db2');
      theFiles(k).cA2 = appcoef(C,L,'db2',2);
      theFiles(k).A2 = wrcoef('a',C,L,'db2',2);
      [C,L] = wavedec(s,3,'db2');
      theFiles(k).cA3 = appcoef(C,L,'db2',3);
      theFiles(k).A3 = wrcoef('a',C,L,'db2',3);
      [C,L] = wavedec(s,4,'db2');
      theFiles(k).cA4 = appcoef(C,L,'db2',4);
      theFiles(k).A4 = wrcoef('a',C,L,'db2',4);
      [C,L] = wavedec(s,5,'db2');
      theFiles(k).cA5 = appcoef(C,L,'db2',5);
      theFiles(k).A5 = wrcoef('a',C,L,'db2',5);
      theFiles(k).Details = C;
      theFiles(k).DetLengths = L;
      theFiles(k).D1 = wrcoef('d',C,L,'db2',1);
      theFiles(k).D2 = wrcoef('d',C,L,'db2',2);
      theFiles(k).D3 = wrcoef('d',C,L,'db2',3);
      theFiles(k).D4 = wrcoef('d',C,L,'db2',4);
      theFiles(k).D5 = wrcoef('d',C,L,'db2',5);

    end

  end

  ModdedFiles(c).transforms = theFiles;
  ModdedFiles(c).stat = sstat;
  ModdedFiles(c).ver = 'db2-60days-2beforefail';
  c = c + 1;
  load('REPless200.mat', 'theFiles');

end
```

This code performs the aggregate measures on the structure created in the previous code block. Division into measurement periods of 5 days is done using the index in each of the aggregate functions (i.e., mean, std, mad).

```
load('MEASURE.mat', 'ModdedFiles');
MeasuredFiles(1).transforms = ModdedFiles(1).transforms;
MeasuredFiles(1).stat = 0;
MeasuredFiles(1).ver = 'ver';

for k = 1 : length(ModdedFiles)

  MeasuredFiles(k) = ModdedFiles(k);
  tempTheFiles = MeasuredFiles(k).transforms;

  for j = 1 : length(tempTheFiles)

    sec = ["D1","D2","D3","D4"];
    lensec = 4;

    if((length(tempTheFiles(j).A1)) > 30)
      sec = ["D1","D2","D3","D4","D5"];
      lensec = 5;
    end

    secsofdata = ((length(tempTheFiles(j).A1)) / 5);

    for m = 1 : lensec;

      curdata = tempTheFiles(j).(sec{m});
      means = [0]; standdev = [0]; meanstand = [0]; medstand = [0];

      for n = 1 : secsofdata


        means(n) = mean(curdata(((n*5)-4):n*5));
        standdev(n) = std(curdata(((n*5)-4):n*5));
        meandev(n) = mad(curdata(((n*5)-4):n*5));


      end

      tempTheFiles(j).("Means").(sec{m}) = means;
      tempTheFiles(j).("Stds").(sec{m}) = standdev;
      tempTheFiles(j).("MeanDevs").(sec{m}) = meandev;



    end

    for p = 1 : lensec
      for o = 2 : lensec

        if(p < o)

          tempTheFiles(j).("MeanRatios").(sec{p}+""+sec{o}) = (tempTheFiles(j).("Means").(sec{p})) ./ (tempTheFiles(j).("Means").(sec{o}));
          tempTheFiles(j).("StdRatios").(sec{p}+""+sec{o}) = (tempTheFiles(j).("Stds").(sec{p})) ./ (tempTheFiles(j).("Stds").(sec{o}));
          tempTheFiles(j).("MeanDevRatios").(sec{p}+""+sec{o}) = (tempTheFiles(j).("MeanDevs").(sec{p})) ./ (tempTheFiles(j).("MeanDevs").(sec{o}));


        end

      end
    end

  end

  MeasuredFiles(k).transforms = tempTheFiles;

end
```