Irrational Exuberance: Why we should not believe 95% of papers on Time Series Anomaly Detection

Eamonn Keogh

Distinguished Professor
University of California Riverside

Disclaimer

- I will make some unflattering claims about academic research in anomaly detection.
- Some of my students did help in running experiments. However, these are *my* viewpoints, and are not necessarily endorsed by my students or UCR.
- I am not claiming that *my* research is free of these flaws, or flaws in general.
- My title is click-bait, sorry. However, it is also true.
- My slides are a too wordy, sorry. But I do hope people will also read this offline.

Overarching Claim

- About 95% of papers on Time Series Anomaly Detection (TSAD) have one or more flaws. These flaws include:
  - Testing on deeply flawed datasets
    - Trivial
    - Mislabeled
    - Unrealistic Anomaly Density
    - Run-to failure
  - Use of inappropriate measures of success
  - Non-reproducible experiments
  - Assuming Deep Learning is the answer and ignoring competitive decade-year-old methods
  - Stabbing William of Ockham in the Heart unjustified complexity
  - Not doing anomaly detection, but then calling it anomaly detection.
- Because of these flaws, I argue that their contributions are nil.

Testing on deeply flawed datasets

- Some papers only test on private datasets or synthetic data[2], I will ignore these, as we all should!

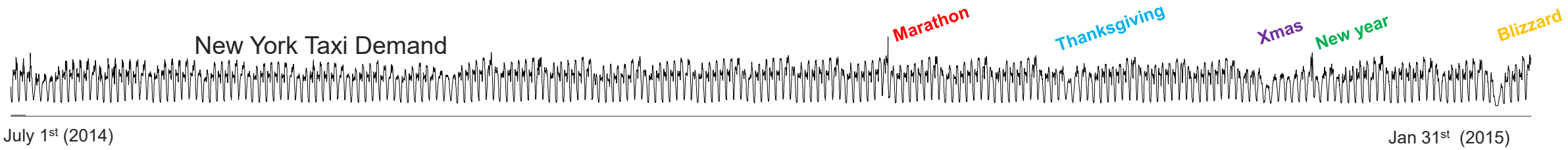
- The vast majority of TSAD papers[1] use one or more of datasets created by **Yahoo**, **Numenta**, **SMAP** (NASA), **MSL** (NASA), **SDM** ("OMNI" Pei's Lab), **MBA-ECG** (Boniol) or **SWAT**.

- Let us take the time to look at these benchmark datasets.

[1]Wu and Keogh: **Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress.**
[2]I do see a *limited* role for some experiments on synthetic data in *some* cases

The Benchmarks are often Mislabeled: Part I

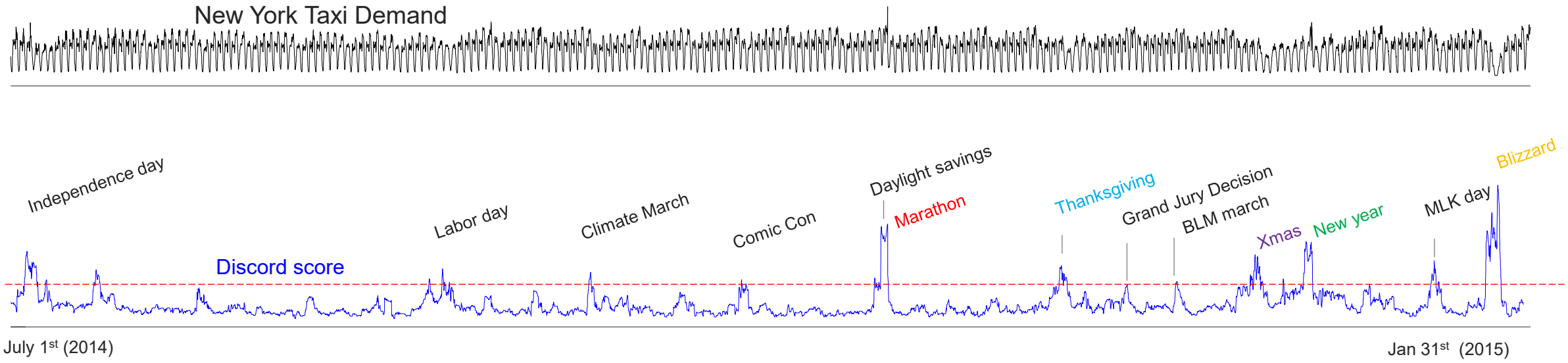
Consider the famous New York Taxi example from Numenta. This is one of the most common benchmarks.
It is claimed that there are *five* anomalies: NYC marathon, Thanksgiving, Christmas, New Year's day, and a Blizzard.



The Benchmarks are often Mislabeled: Part I

Consider the famous New York Taxi example from Numenta. This is one of the most common benchmarks
It is claimed that there are *five* anomalies: NYC marathon, Thanksgiving, Christmas, New Year's day, and a Blizzard.

However, I would argue that there are *at least* five or six additional anomalies, including additional holidays and protests.
Moreover, the anomaly called *Marathon* is really the daylight savings clock change from the night before.



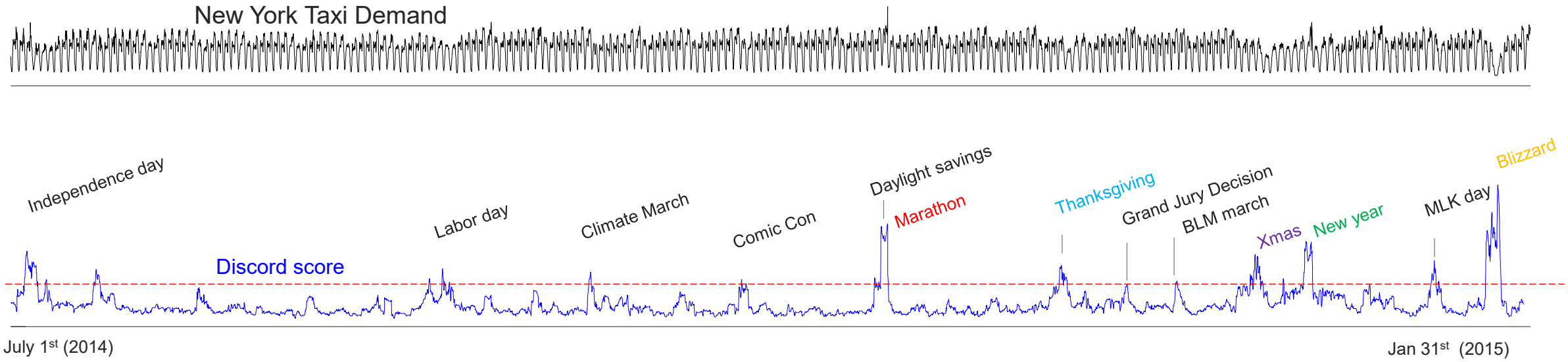
[1]Len Feremans, et al. Pattern-Based Anomaly Detection in Mixed-Type Time Series. ECML/PKDD (1) 2019: 240-256

The Benchmarks are often Mislabeled: Part I

Consider the famous New York Taxi example from Numenta. This is one of the most common benchmarks
It is claimed that there are *five* anomalies: NYC marathon, Thanksgiving, Christmas, New Year's day, and a Blizzard.

However, I would argue that there are at least five or six additional anomalies, including additional holidays and protests. Moreover, the anomaly called *Marathon* is really the daylight savings clock change from the night before.

Knowing this, what do you think of a claim such as this, from a recent paper[1]: "*On the NY Taxi dataset FPOF got 0.877, but we got 0.879, showing our method is better*"?



[1]Len Feremans, et al. Pattern-Based Anomaly Detection in Mixed-Type Time Series. ECML/PKDD (1) 2019: 240-256

Once you realize that the claim of five anomalies in NY Taxi is nonsense. You begin to see many published claims as strange…

> "*The performance of Tri-CAD is compared with those of related methods, such as STL, SARIMA, LSTM, LSTM with STL, and ADSaS. The comparison results show that Tri-CAD outperforms the others in terms of the precision, recall, and F1 –score*"

The perfect precision, recall and F1-scores claimed here, just happens to agree with significant mislabeling.

This strongly suggests *overfitting*


applied sciences

Article

MDPI

Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning

Jehn-Ruey Jiang *, Jian-Bin Kao and Yu-Lin Li

Department of Computer Science and Information Engineering, National Central University, Taoyuan City 32001, Taiwan; 106522115@cc.ncu.edu.tw (J.-B.K.); 107522120@cc.ncu.edu.tw (Y.-L.L.)

* Correspondence: jrjiang@csie.ncu.edu.tw

Abstract: Thanks to the advance of novel technologies, such as sensors and Internet of Things (IoT) technologies, big amounts of data are continuously gathered over time, resulting in a variety of time series. A semi-supervised anomaly detection framework, called Tri-CAD, for univariate time series is proposed in this paper. Based on the Pearson product-moment correlation coefficient and Dickey–Fuller test, time series are first categorized into three classes: (i) periodic, (ii) stationary, and (iii) non-periodic and non-stationary time series. Afterwards, different mechanisms using statistics, wavelet transform, and deep learning autoencoder concepts are applied to different classes of time series for detecting anomalies. The performance of the proposed Tri-CAD framework is evaluated by experiments using three Numenta anomaly benchmark (NAB) datasets. The performance of Tri-CAD is compared with those of related methods, such as STL, SARIMA, LSTM, LSTM with STL, and ADSaS. The comparison results show that Tri-CAD outperforms the others in terms of the precision, recall, and F_1-score.

Keywords: anomaly detection; autoencoder; deep learning; Internet of Things; sensors; time series; wavelet transform

1. Introduction

Anomaly detection involves identifying data that do not conform to the pattern as expected [1,2]. It is also known as outlier detection [3,4] or novelty detection [5]. The research of anomaly detection can be traced back to the 18th century when Bernoulli commented on the widespread practice of discarding discordant data in the absence of a priori information in 1777 [6,7]. Undoubtedly, anomaly detection has attracted much research attention, bring about plenty of research results that can be utilized in various applications. Typical applications of anomaly detection research include outlier detection for big data [8], Internet of Things (IoT) [9], wireless sensor networks (WSNs) [10], network anomaly/intrusion/attack detection [11–16], e-anomaly detection for manufacturing [17–21], video surveillance anomaly detection [22–24], road network/highway traffic anomaly detection [24,25], credit card fraud identification [22,23], insurance fraud detection [26], ECG (electrocardiogram) monitoring [27], suspicious financial transaction detection [28], social media rumor detection [29], and so on.

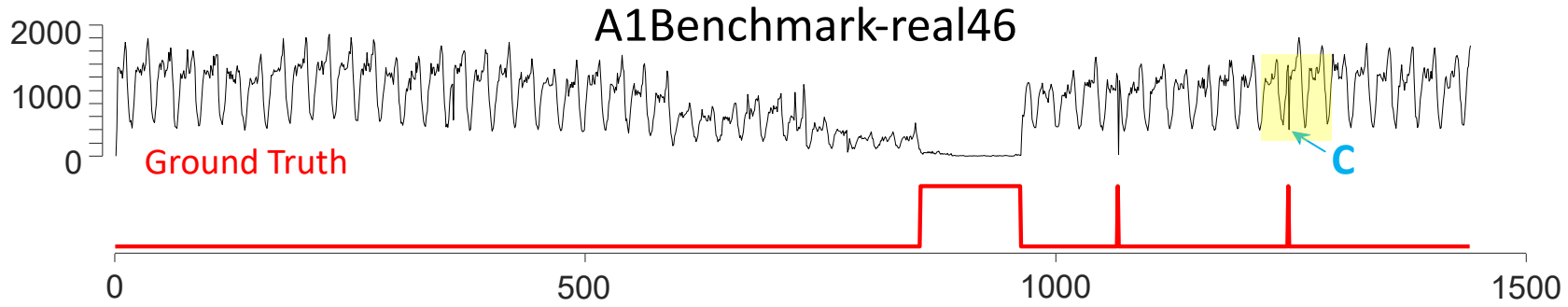
Thanks to the advances of novel technologies, such as sensor technologies and information and communication technologies, big amounts of data are continuously gathered over time. Data that are recorded chronologically constitute a time series. Typical examples of time series include the following gathered data: hourly temperatures of a machine, daily stock prices of a company, weekly interest rates, monthly temperatures of a machine, daily (gross domestic) product of a nation. Time series anomaly detection is very important; it is one of fundamental time series analysis tasks. Many studies have focused on such a research field in the last decades. The readers are referred to the survey papers [32–35] for details.

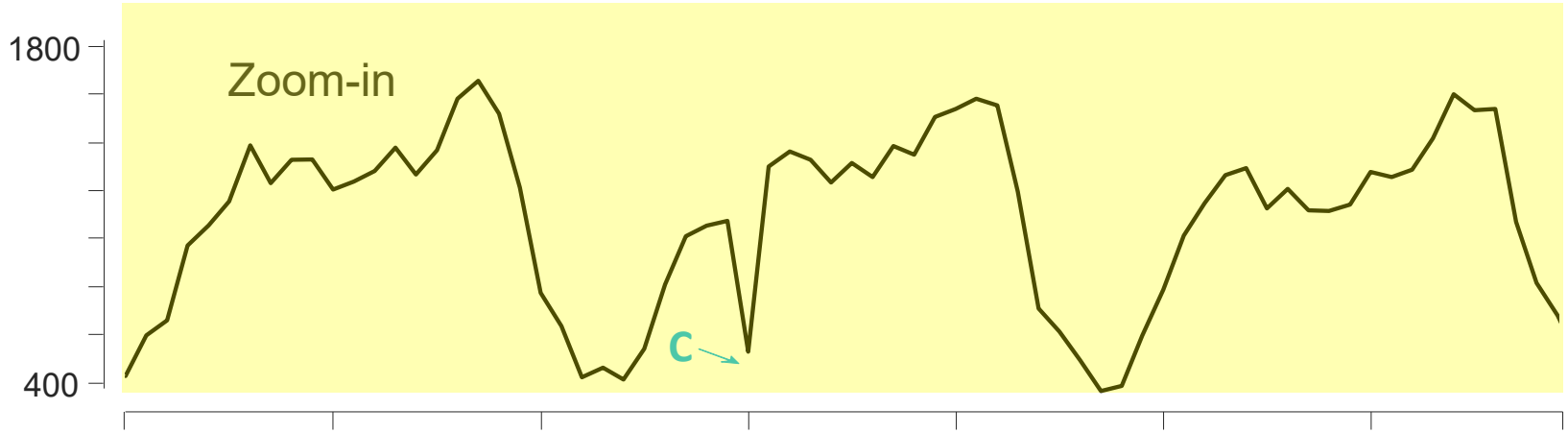
Table 1. Comparisons of the proposed framework Tri-CAD and related met

Time Series	Class	Fixed Window Size (*fws*)	Metrics	STL only	SARIMA only	LSTM only	LSTM with STL	Proposed Framework Tri-CAD
NAB			Precision	0.533	0.000	0.176	0.161	**1.000**
NYC	Class 1	206	Recall	0.889	0.000	0.333	1.000	**1.000**
Taxi			F_1-score	0.667	0.000	0.231	0.277	**1.000**

The Benchmarks are often Mislabeled: Part IIa

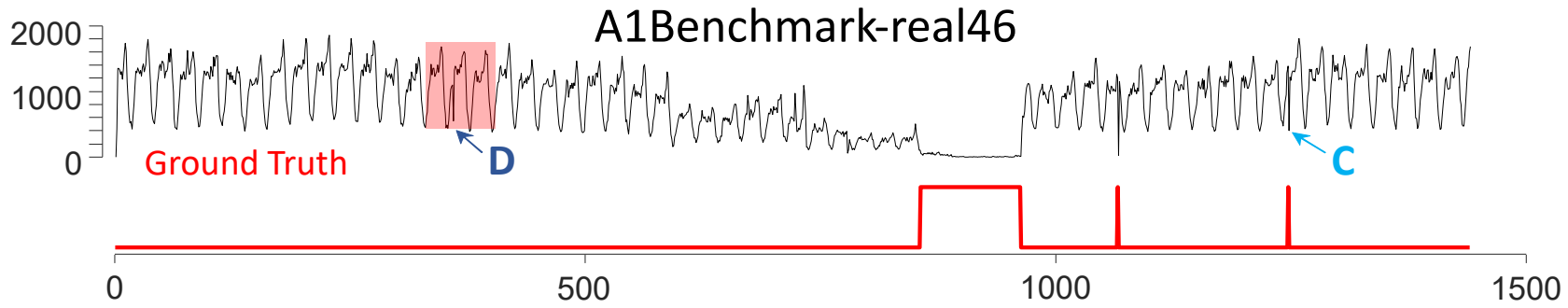
In this Yahoo dataset, **C** is an anomaly...

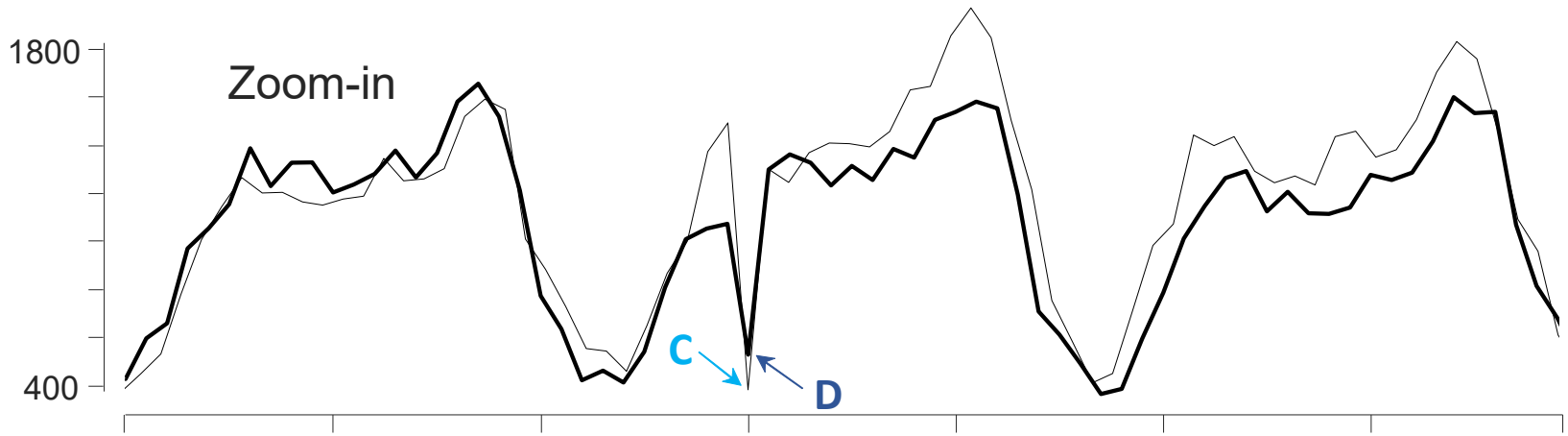




The Benchmarks are often Mislabeled: Part IIb

In this Yahoo dataset, **C** is an anomaly, but **D** is not, yet they are *virtually identical* dropouts.

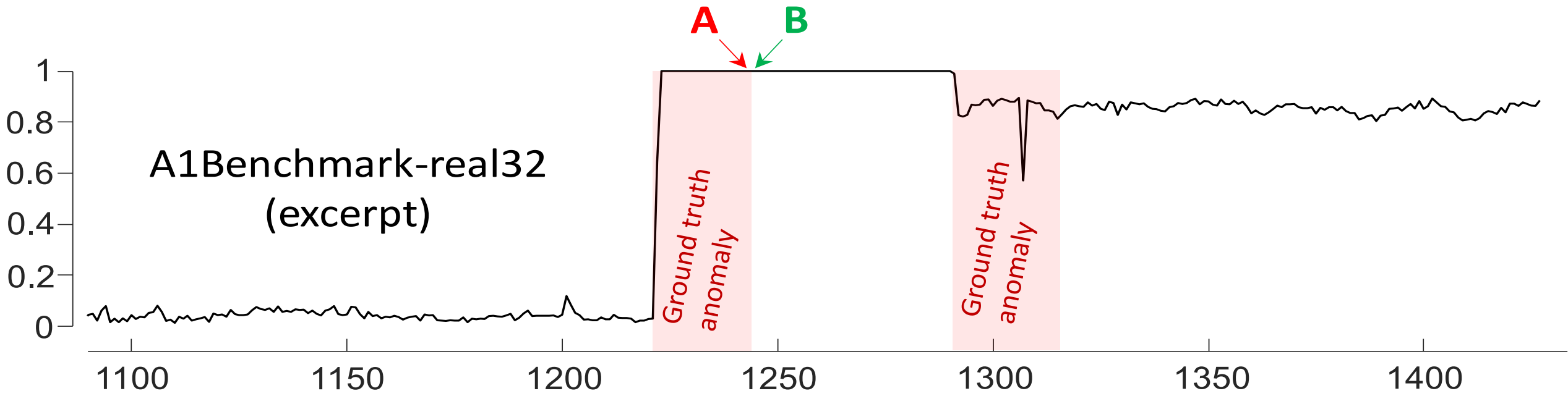




The Benchmarks are often Mislabeled: Part III

In this Yahoo dataset it is claimed **A** is an anomaly, but **B** is not

But literally *nothing* has changed between the two points



Mislabeling really matters!

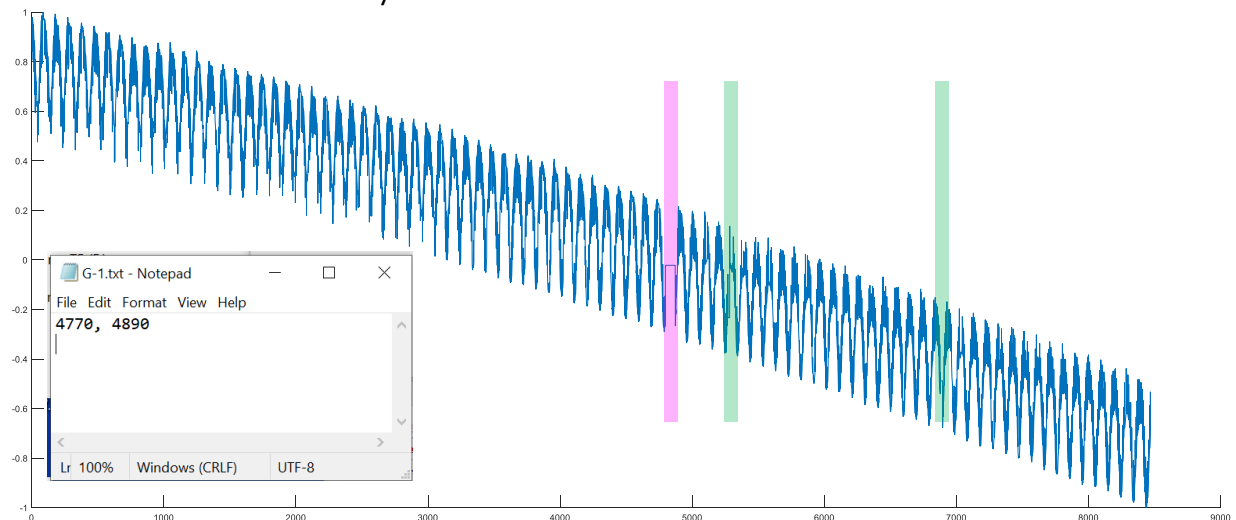
This paper claims it outperforms all rivals on MSL dataset[a]. But the margin of victory over three of its rivals is less than 3%.

However, the amount of mislabeling in this dataset an order of magnitude greater than that!

"USAD outperforms all methods on MSL.." [a]

Methods	MSL			
	P	R	F1	F1*
AE	0.8535	0.9748	0.8792	0.9101
IF	0.5681	0.6740	0.5984	0.6166
LSTM-VAE	0.8599	0.9756	0.8537	0.9141
DAGMM	0.7562	0.9803	0.8112	0.8537
OmniAnomaly	0.9140	0.8891	0.8952	0.9014
USAD	0.8810	0.9786	**0.9109**	**0.9272**

Here is an example from MSR: G-1. The only anomaly labeled in 4770 to 4890. However surely 4270 to 4285 and 6880 to 6894 are anomalies too.



I have tried and tried to tell folks that if the underlying uncertainty in your labels is larger than any change in relative performance, the change is meaningless

Vijayant K. VP of Product: ML & AI at Optum

[a] USAD : UnSupervised Anomaly Detection on Multivariate Time Series

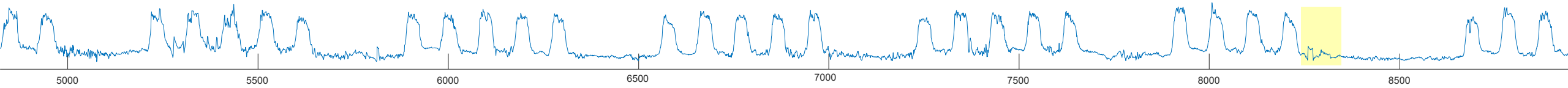
My claim of mislabeled data is almost tautological

In fact, perfect ground truth labels are **impossible** for anomaly detection!

- (however, it *really* is the case that most of the benchmark datasets have mislabelings. In some cases, I was able to confirm with the datasets creators that they had made errors)

Ground Truth Labels are Impossible for Anomaly Detection!

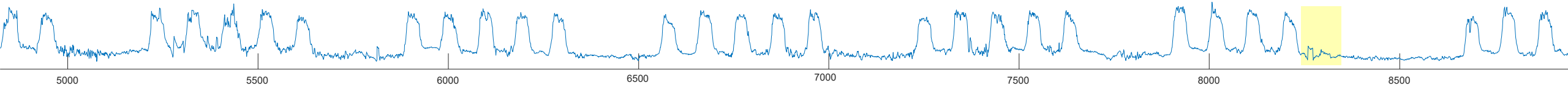
- For some ML problems, we *can* get perfect ground truth, i.e., cats vs dogs
- However, for anomaly detection, we can *never* have perfect ground truth.
- Consider the example below, the electrical power demand for a factory…



- Many people have labeled a Friday holiday as an anomaly, that seems reasonable, right?

Ground Truth Labels are Impossible for Anomaly Detection!

- For some ML problems, we *can* get perfect ground truth, i.e., cats vs dogs
- However, for anomaly detection, we can *never* have perfect ground truth.
- Consider the example below, the electrical power demand for a factory…



- Many people have labeled a Friday holiday as an anomaly, that seems reasonable, right?
  - However, Joe says "*No! The anomaly is at 5817, when the flood forced us to turn on the emergency pump*"
  - And Sue says "*No! The anomaly is the noise at 4900 to 5100, when we switch from gas to TIG welding*"
  - But Tim says "*No! The anomaly is at 5890, when daylight saving time made a day look longer*"
  - And Bic says "*No! The anomaly is at 7420 when we turned off the night lights for an hour as part of IDA*"
- My point is, we *can* never know all the out-of-band possible causes for anomalies. We can never be sure that the anomaly we see, based on a priori or post-hoc information, is the only or "best" anomaly.

Implication: It is nonsense for anomaly detection papers to publish experimental results with four or five significant digits, when there is always large subjectively and uncertainty as to the ground truth.

The Benchmarks are often Trivial:

- A huge fraction of benchmark datasets are trivial to solve.
- To make that claim more concrete, I will define *trivial*.

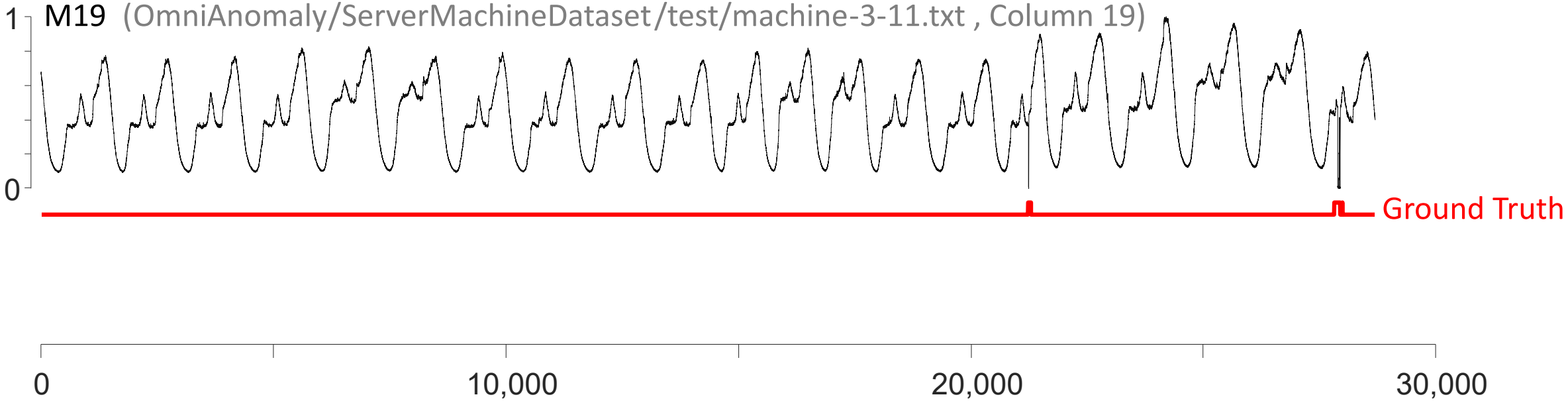
  - A time series anomaly detection problem is *trivial* if it can be solved with a single line of standard library MATLAB code (or Python, R etc.)

    - We cannot "cheat" by calling a high-level built-in function such as *kmeans* or *ClassificationKNN* or calling custom written functions.
    - We must limit ourselves to basic vectorized primitive operations, such as *mean*, *max*, *std*, *diff*, etc.

- We may allow a single magic number in our one-liner. But recall that many anomaly detection algorithms have up to a dozen parameters, and at least a few seem "magic" to me.

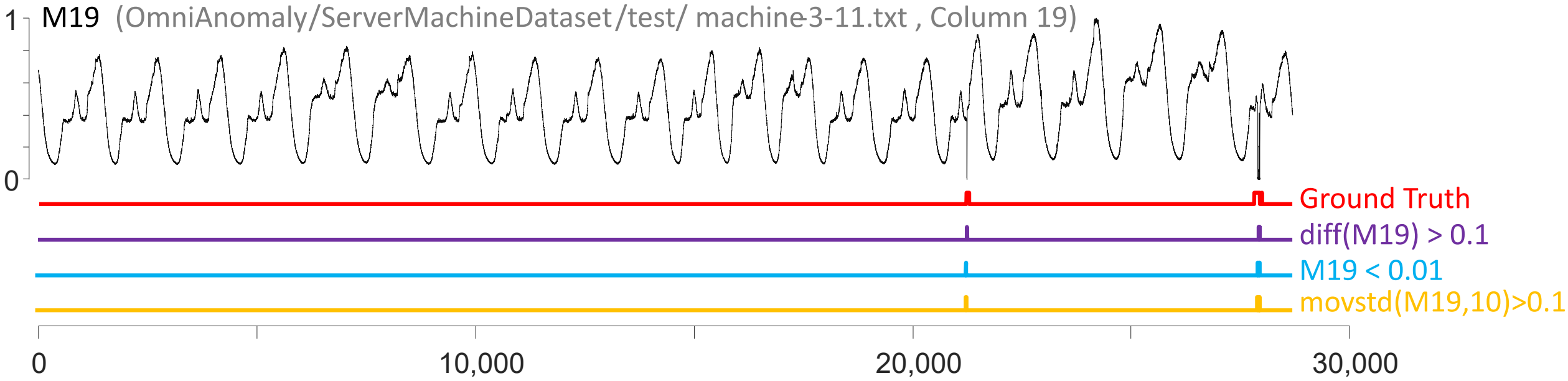
The Benchmarks are often Trivial: Part I

From the OMNI Benchmark, used in dozens of top tier papers.
A test dataset, with ground truth, how hard is this to solve?



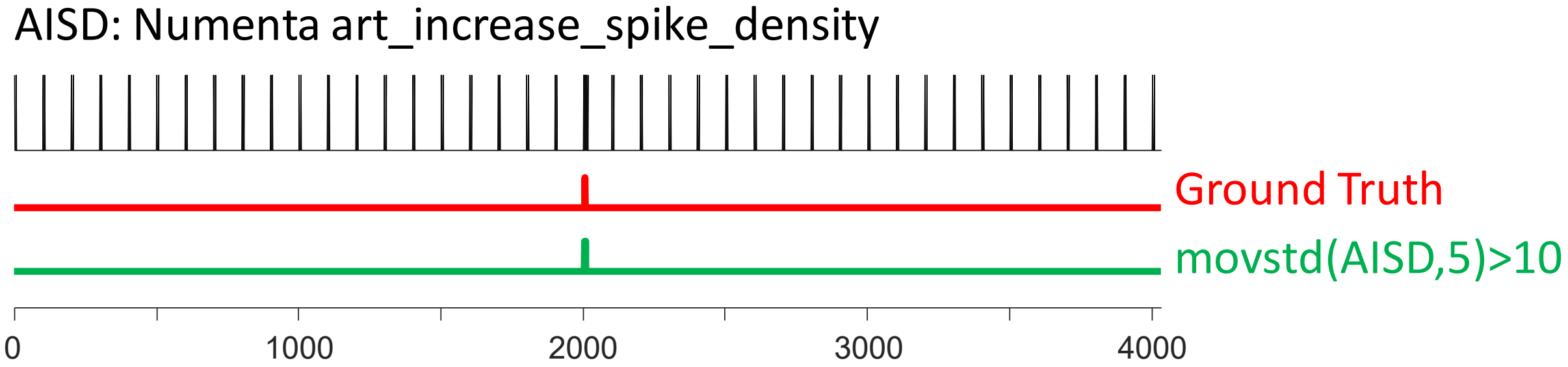
The Benchmarks are often Trivial: Part I

From the OMNI Benchmark.
A test dataset, and three different one-liners that perfectly solve it.



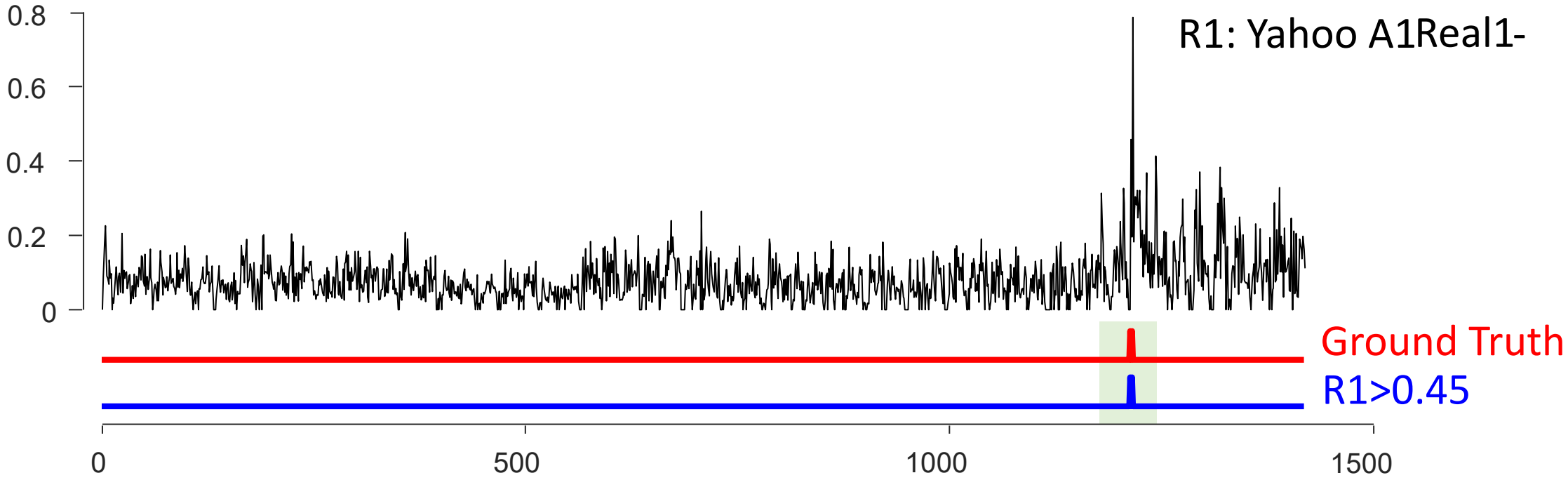
The Benchmarks are often Trivial: Part II

From the Numenta Benchmark.
A test dataset, and a one-liner that perfectly solves it.



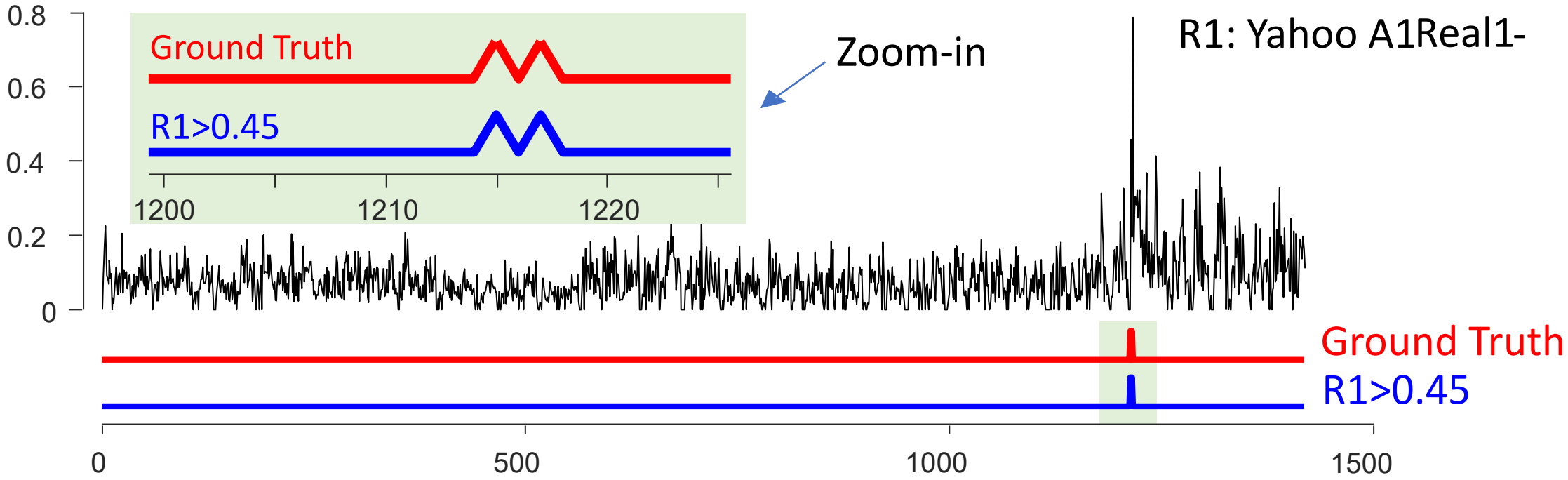
The Benchmarks are often Trivial: Part III

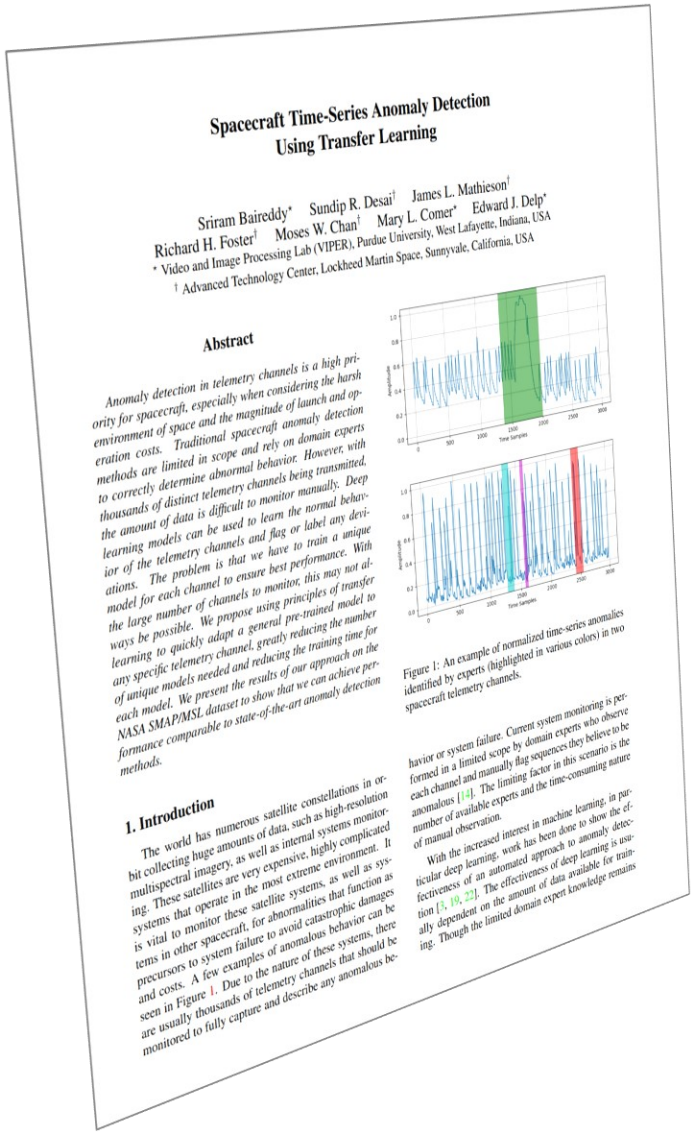
From the Yahoo Benchmark.
A test dataset, and a one-liner that perfectly solves it.



The Benchmarks are often Trivial: Part III

From the Yahoo Benchmark.
A test dataset, and a one-liner that perfectly solves it.
Note how exactly the one-liner predicts the ground truth labels





Spacecraft Time-Series Anomaly Detection
Using Transfer Learning

Sriram Baireddy* Sundip R. Desai† James L. Mathieson†
Richard H. Foster† Moses W. Chan† Mary L. Comer* Edward J. Delp*
* Video and Image Processing Lab (VIPER), Purdue University, West Lafayette, Indiana, USA
† Advanced Technology Center, Lockheed Martin Space, Sunnyvale, California, USA

Abstract

Anomaly detection in telemetry channels is a high priority for spacecraft, especially when considering the harsh environment of space and the magnitude of launch and operation costs. Traditional spacecraft anomaly detection methods are limited in scope and rely on domain experts to correctly determine abnormal behavior. However, with thousands of distinct telemetry channels being transmitted, the amount of data is difficult to monitor manually. Deep learning models can be used to learn the normal behavior of the telemetry channels and flag or label any deviations. The problem is that we have to train a unique model for each channel to ensure best performance. With the large number of channels to monitor, this may not always be possible. We propose using principles of transfer learning to quickly adapt a general pre-trained model to any specific telemetry channel, greatly reducing the number of unique models needed and reducing the training time for each model. We present the results of our approach on the NASA SMAP/MSL dataset to show that we can achieve performance comparable to state-of-the-art anomaly detection methods.

Figure 1: An example of normalized time-series anomalies identified by experts (highlighted in various colors) in two spacecraft telemetry channels.

1. Introduction

The world has numerous satellite constellations in orbit collecting huge amounts of data, such as high-resolution multispectral imagery, as well as internal systems monitoring. These satellites are very expensive, highly complicated systems that operate in the most extreme environment. It is vital to monitor these satellite systems, as well as systems in other spacecraft, for abnormalities that function as precursors to system failure to avoid catastrophic damages and costs. A few examples of anomalous behavior can be seen in Figure 1. Due to the nature of these systems, there are usually thousands of telemetry channels that should be monitored to fully capture and describe any anomalous behavior or system failure. Current system monitoring is performed in a limited scope by domain experts who observe each channel and manually flag sequences they believe to be anomalous [14]. The limiting factor in this scenario is the number of available experts and the time-consuming nature of manual observation.

With the increased interest in machine learning, in particular deep learning, work has been done to show the effectiveness of an automated approach to anomaly detection [3, 19, 22]. The effectiveness of deep learning is usually dependent on the amount of data available for training. Though the limited domain expert knowledge remains

From the NASA MSL dataset.

A test dataset with three anomalies..

A complex method finds the three anomalies, plus one false positive.

However, a one liner can finds just the three anomalies correctly

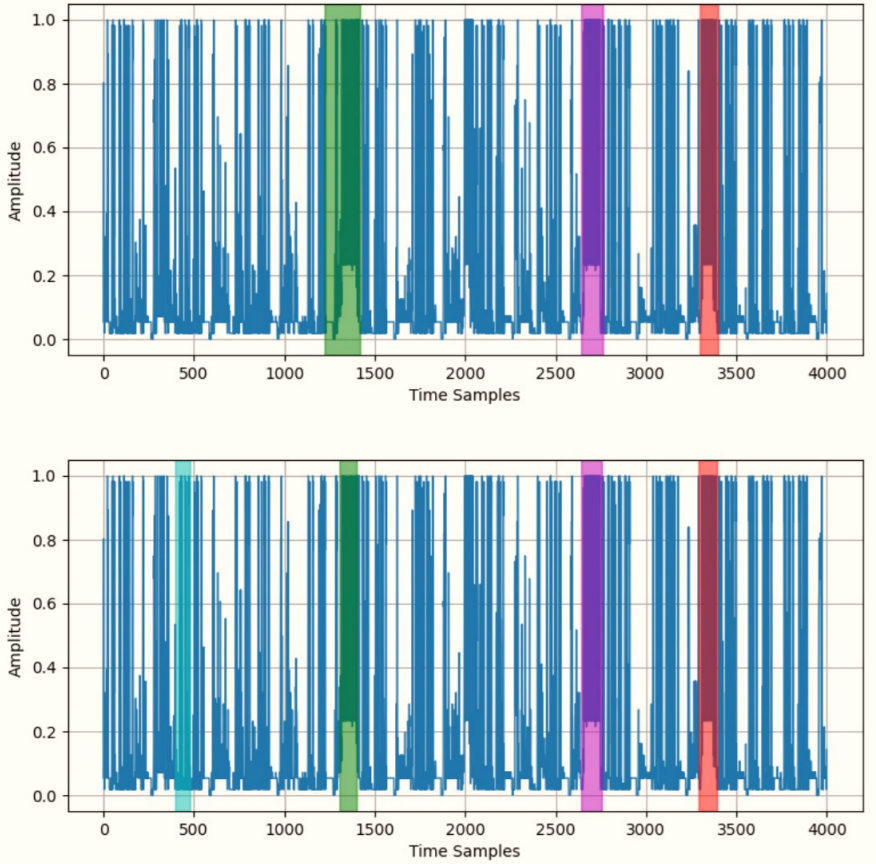
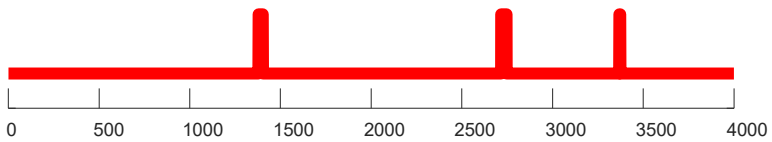


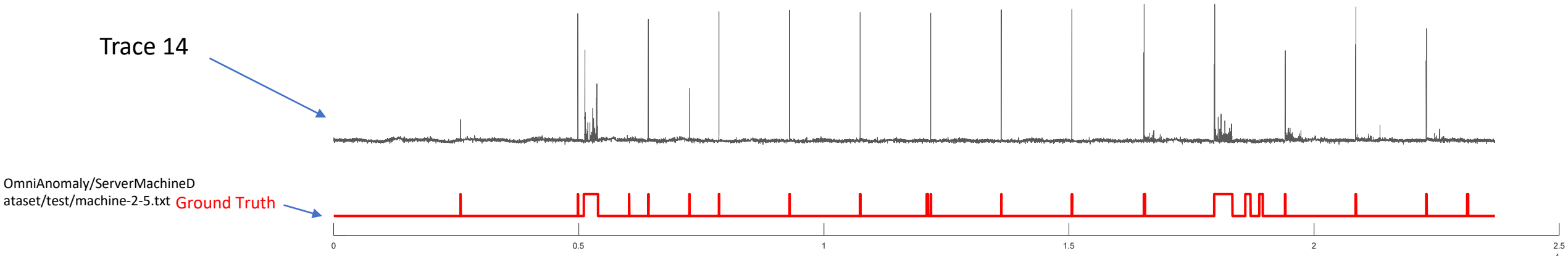
Figure 5: The result of MRONet-FT on channel F-7 from the MSL data. The top row shows the actual anomaly locations and the bottom row shows the detected anomaly sequences, including a false positive.

plot(movmin(F7,64)>-0.75,'r')



The Benchmarks are often Trivial: Part IIII

From the OMNI Benchmark.
This dataset has lots of anomalies, it has 38 dimensions and comes with training data.

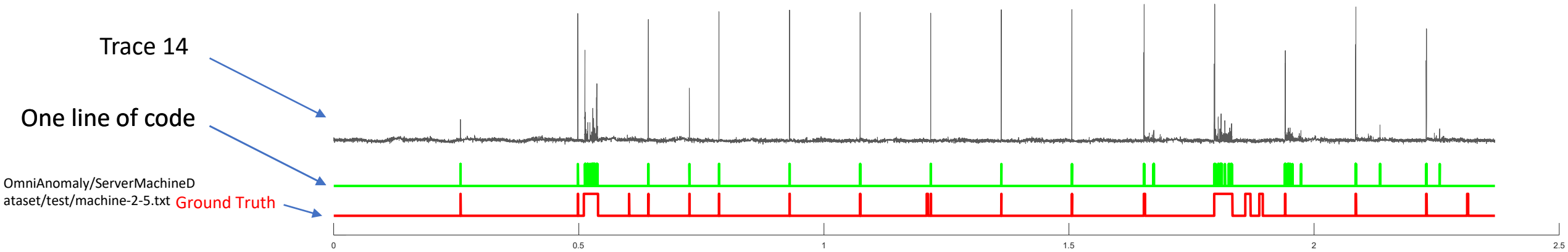


The Benchmarks are often Trivial: Part IIII

From the OMNI Benchmark.
This dataset has lots of anomalies, it has 38 dimensions and comes with training data.

But here, a single line of code, and a single dimension, no training data, no parameters, we can do almost perfectly, and better than any published result.
This single line of code is basic **statistical process control**, about 80 years old.

```
T > mean(T) + (2* std(T));
```



Not convinced by one-liner argument?

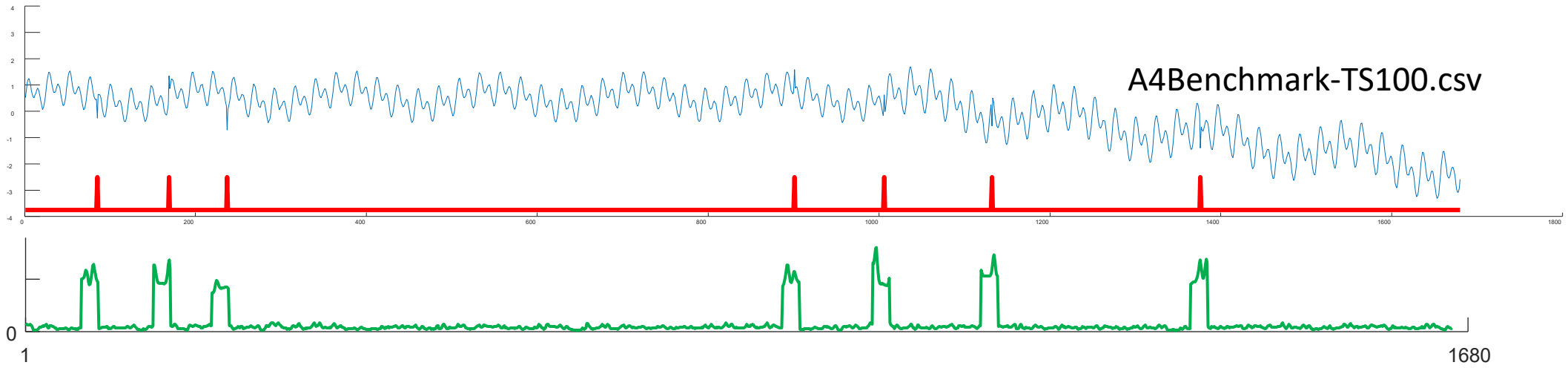
Let's look at one of the Yahoo Benchmarks, and compare it to a 20-year old simple method called *time series discords*.

(Time series discords can be computed by the Matrix Profile, or faster O(n) algorithms)

Time Series Discords are:

- Fast to compute
- Simple to implement: 10 lines of code, a little more to compute fast.
- Require only a single parameter: A recent paper removed even that parameter. See MERLIN
- Do not need training data (but can use it if needed)

Time Series Discords do *extremely well* on all the benchmark datasets.



E. Keogh, J. Lin and A. Fu (2005). HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. (ICDM 2005)

Philosophically: What does the one-liner argument mean?

The one-liner argument could be cast in more rigorous terms, perhaps arguing about linear separability or Kolmogorov complexity. However, that seems to be pretentious and unneeded.

Imagine the following problems

- Determine if an audio signal contains Brazilian or European Portuguese
- Determine if a text review of a product is positive or negative
- Segmentation of an arbitrary song into *intro*, *verse*, *chorus*, *bridge*, and *outro*

All these problems *are* solvable with ML.

Philosophically: What does the one-liner argument mean?

The one-liner argument could be cast in more rigorous terms, perhaps arguing about linear separability or Kolmogorov complexity. However, that seems to be pretentious and unneeded.

Imagine the following problems

- Determine if an audio signal contains Brazilian or European Portuguese
- Determine if a text review of a product is positive or negative
- Segmentation of an arbitrary song into *intro*, *verse*, *chorus*, *bridge*, and *outro*

All these problems *are* solvable with ML.

However, wouldn't you be suspicious if I could solve one of these with a single line of code?

You might investigate and perhaps say: *Ah yes, you could separate positive and negative reviews in that dataset, but **only** because positive reviews are in all uppercase, and negative reviews are in all lower case. Your success here says nothing about the general problem*.

This is what the one liner argument is saying. If I can solve a problem with five seconds of thought and one line of code, then surely the task has some trivial structure that makes it too easy to be interesting.

The Benchmarks are often Trivial: Summary

At least 90% of the benchmark datasets can be solved with very simple methods dating back decades, or with “one-liners”.

At least 90% of the benchmark datasets can be solved without needing to even look at the training data!

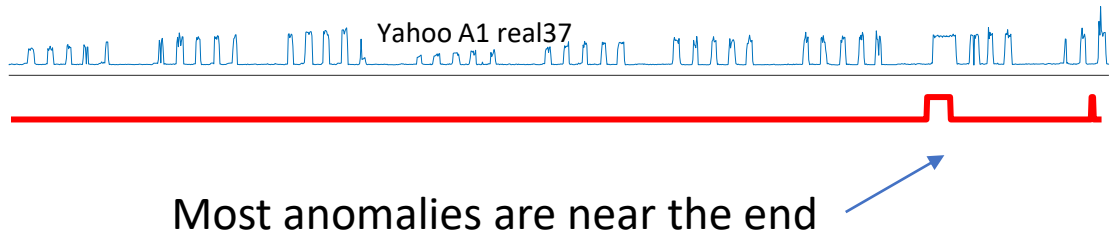
This should be worrisome. In what sense do we need machine *learning*, or deep *learning*, when it is not clear we need to *learn* from the training data in any way?

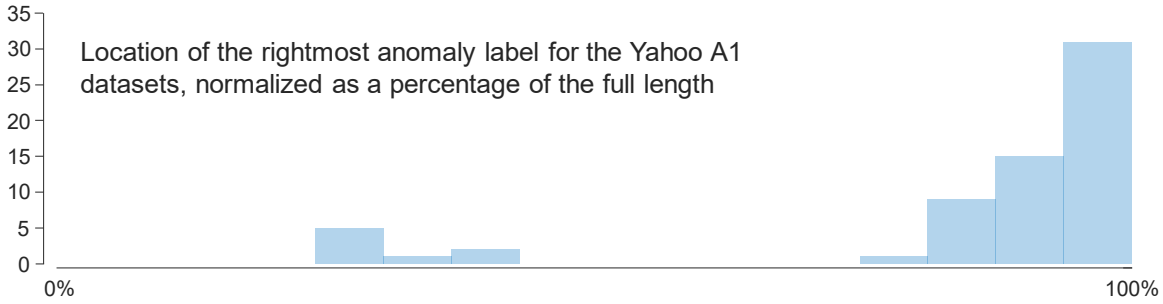
E.S. Page, “On Problems in which a Change in a Parameter Occurs at an Unknown Point,” Biometrika, vol. 44, no. 1-2, 1957, pp. 248-252.

The Benchmarks have other Problems: Run to failure bias

- (esp. Yahoo and NASA) **Run to failure bias**: Most of the anomalies appear at the end of a time series.

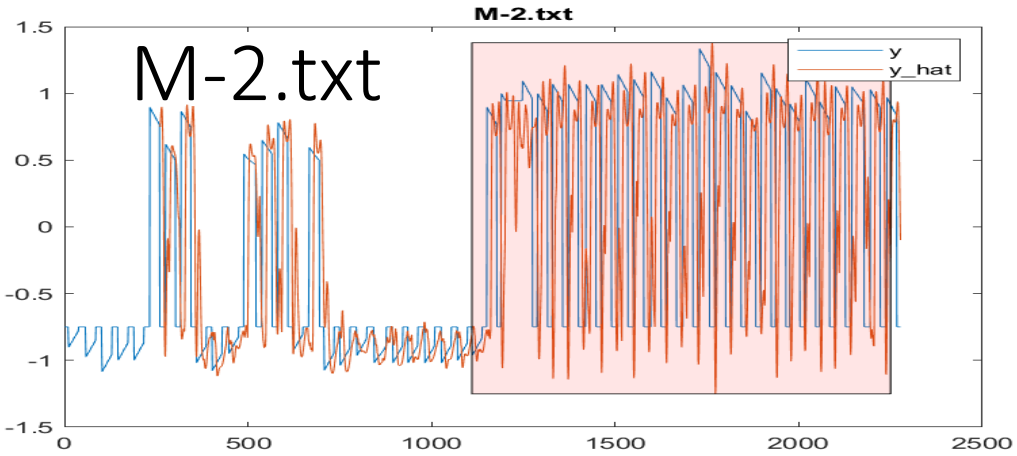
  This means just guessing “near the end” does quite well.

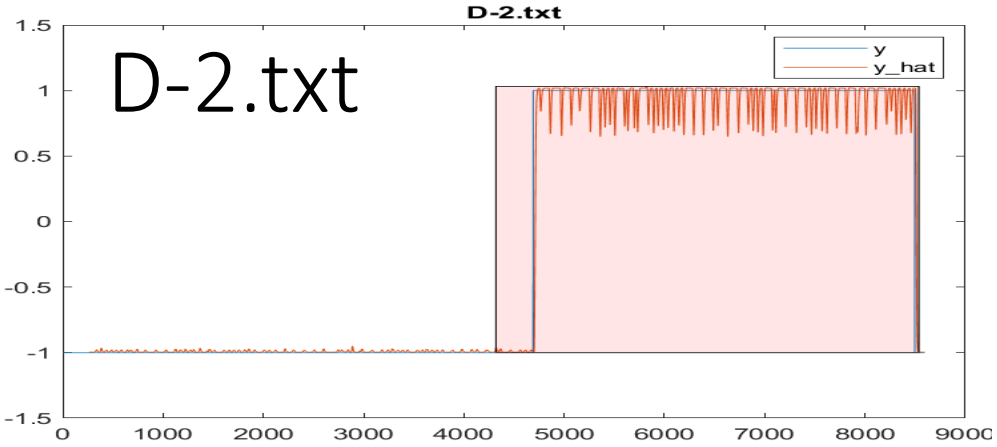




The Benchmarks have other Problems: Unrealistic Anomaly Density I

- Consider these examples from NASA MSL
- More that half the data is labeled as being an anomaly!
- “anomaly” is a synonym for “rarity”, but these anomalies sure aren’t rare.
- A real anomaly detection algorithm must be able to deal with the tiny prior probability of seeing an anomaly.

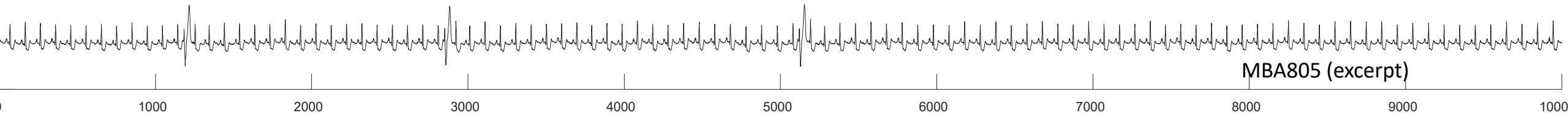




- As an aside, note that we also have both *run-to-failure bias* and *triviality,* and probably partial *mislabeling*

The Benchmarks have other Problems: Unrealistic Anomaly Density II

- Consider this dataset. It has 133 “anomalies”, all of which are basically identical arrythmias. In my view…
  - Isn't this much closer to a classification or clustering problem?
  - If you find one anomaly, you are going to find them all. Reporting *we got 133 out of 133!* implies an unwarranted level of utility and success.
  - Even better, I would *try* to get one wrong, so I could report: *Our accuracy is 0.9924!*
  - Note that it is also trivial in the *one line of code* sense



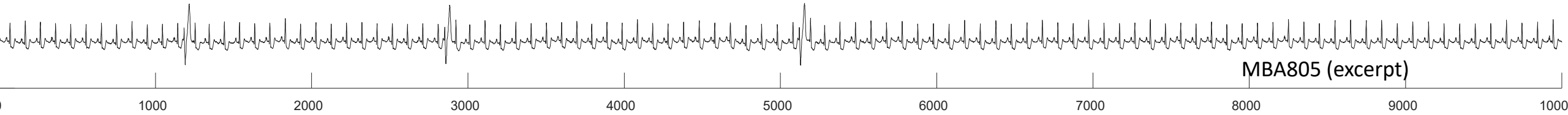
Also suffers from *triviality*

The Benchmarks have other Problems: Unrealistic Anomaly Density III

In the real world, anomalies are rare…

One anomaly a year, not good. Two anomalies in a year, people start talking. Three anomalies in a year, better start looking for a new job

Mike Noskov, Director, Data Science, Aspen Technology



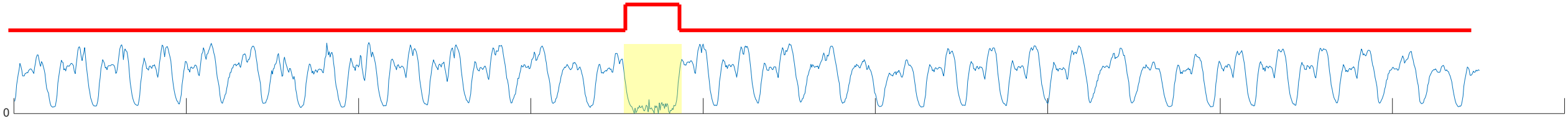
Also suffers from *triviality*

Spurious Precision is Rampant in TSAD I

- Many papers report three, four or five digits of precision for their experimental results.
- Of course, the great uncertainly in ground truth labels suggests that this is meaningless.

Spurious Precision is Rampant in TSAD I

- Many papers report three, four or five digits of precision for their experimental results.
- Of course, the great uncertainly in ground truth labels suggests that this is meaningless.
- However, *even* if we assumed that we had perfect ground truth labels, much of the precision claimed would *still* be wrong.
- Consider the below, this once-a-hour sampled sensor was broken from midnight to midnight on Xmas day.



- Suppose my algorithm finds this, can I claim that I got 24 out of 24?
- No!, these are not independent events, **I got 1 out of 1, not 24 out of 24!**

- Is this obvious to you? Great, but see the next two slides, and see many TSAD papers.

Spurious Precision is Rampant in TSAD I

- Consider this very trivial anomaly detection dataset, *ECG(A)* (it is shown in is entirety!!).
- A recent paper published an experiment on this dataset, giving results with 4 significant digits!
- This implies ludicrously fine distinctions are being made. However, I argue the results should be binary (*detected* | *not-detected*).
- If the precision relates to timing, then it is making a distinction down to 1/250th of a second, something that is medically impossible. (Quote from Cardiologist Dr. Greg Mason "*No one could meaningfully label the onset with a precision greater than 1/10th of a second*")
- Many TSAD papers make similarly meaninglessly overspecified claims, giving the illusion of careful and statistically meaningful distinctions being made. This is just nonsense.

Table 3: Effectiveness of coarse-to-fine fusion in the proposed RAMED model.

ECG(A)	AUROC	AUPRC	F1
w/o coarse-to-fine fusion	0.6479	0.5035	0.4965
full model	**0.7358**	**0.5714**	**0.5427**

[a] Lifeng Shen, Zhongzhong Yu, Qianli Ma, James T. Kwok: Time Series Anomaly Detection with Multiresolution Ensemble Decoding. AAAI 2021: 9567-9575

Non-reproducible Experiments *and* Spurious Precision

- There are many papers that publish on *time series* that are publicly available, but the anomaly *labels* are not available!
- But if the anomaly *labels* are not available, then you cannot reproduce a single number.
- Moreover, this practice is open to an obvious idea for abuse: [1]Run your algorithm, then use *its* predictions as the ground truth!

Neurips 2020

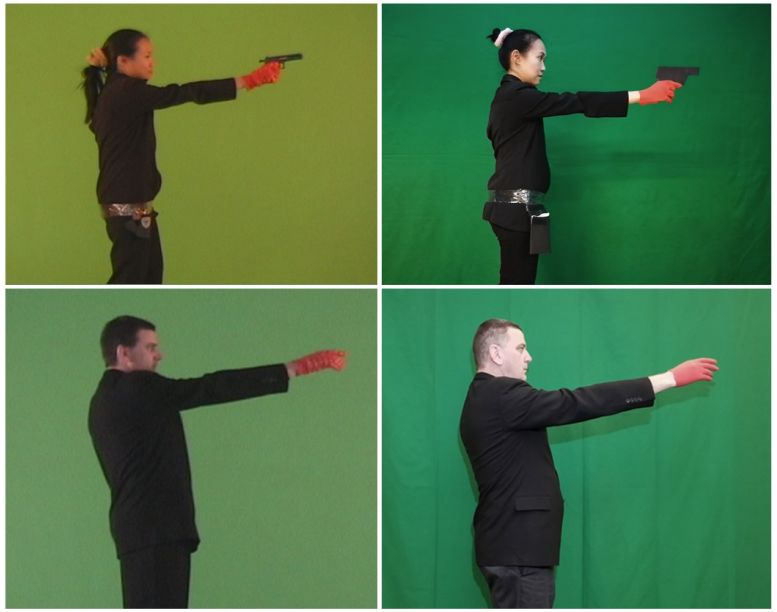
Table 2: Precision (prec), recall (rec) and F1 score results (as %) on various data sets. The number in brackets after the F1 value is the rank of the method. The smaller the better

	2D-gesture			power-demand		
	prec	rec	F1	prec	rec	F1
LOF	27.82	87.21	42.18 (8)	15.29	28.13	19.81 (9)
OC-SVM	65.50	25.57	36.78 (14)	12.40	60.43	20.58 (8)
iso forest	28.54	68.04	40.22 (10)	7.85	89.77	14.44 (13)
deep SVDD	26.26	64.53	37.32 (13)	11.51	64.74	19.54 (10)
AnoGAN	57.85	46.50	51.55 (4)	20.28	44.41	28.85 (5)
DAGMM	25.66	80.47	38.91 (12)	34.37	41.72	37.69 (4)
EncDec-AD	24.88	100.0	39.85 (11)	13.98	54.20	22.22 (6)
LSTM-VAE	36.62	67.76	47.54 (6)	8.00	56.66	14.03 (14)
MadGAN	29.41	76.40	42.47 (7)	13.20	60.57	21.67 (7)
BeatGAN	55.11	45.33	49.74 (5)	8.04	76.58	14.56 (12)
OmniAnomaly	27.70	79.67	41.11 (9)	8.55	78.73	15.42 (11)
MSCRED	61.26	59.11	60.17 (2.5)	55.80	34.32	42.50 (3)
CVDD	56.05	64.95	60.17 (2.5)	49.65	38.36	43.30 (2)
THOC	54.78	75.00	**63.31** (1)	61.50	36.34	**45.68** (1)

Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network

[1]I am not claiming the paper on this page did that, but I do think papers have done this

Non-reproducible Experiments *and* Spurious Precision



- In this case, I happen to have introduced both 2D-guesture and power-demand to the community almost 20 years ago.
- These are reasonable datasets for showing anecdotal examples.
- But the anomalies within them are highly subjective, there is simply no way to pull out four significant digits from these datasets.
- There is no unambiguous way these authors could have obtained ground truth here. These results are nonsense.

Neurips 2020

Table 2: Precision (prec), recall (rec) and F1 score results (as %) on various data sets. The number in brackets after the F1 value is the rank of the method. The smaller the better

	2D-gesture			power-demand		
	prec	rec	F1	prec	rec	F1
LOF	27.82	87.21	42.18 (8)	15.29	28.13	19.81 (9)
OC-SVM	65.50	25.57	36.78 (14)	12.40	60.43	20.58 (8)
iso forest	28.54	68.04	40.22 (10)	7.85	89.77	14.44 (13)
deep SVDD	26.26	64.53	37.32 (13)	11.51	64.74	19.54 (10)
AnoGAN	57.85	46.50	51.55 (4)	20.28	44.41	28.85 (5)
DAGMM	25.66	80.47	38.91 (12)	34.37	41.72	37.69 (4)
EncDec-AD	24.88	100.0	39.85 (11)	13.98	54.20	22.22 (6)
LSTM-VAE	36.62	67.76	47.54 (6)	8.00	56.66	14.03 (14)
MadGAN	29.41	76.40	42.47 (7)	13.20	60.57	21.67 (7)
BeatGAN	55.11	45.33	49.74 (5)	8.04	76.58	14.56 (12)
OmniAnomaly	27.70	79.67	41.11 (9)	8.55	78.73	15.42 (11)
MSCRED	61.26	59.11	60.17 (2.5)	55.80	34.32	42.50 (3)
CVDD	56.05	64.95	60.17 (2.5)	49.65	38.36	43.30 (2)
THOC	54.78	75.00	**63.31** (1)	61.50	36.34	**45.68** (1)

Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network

Assuming that *Deep Learning* is the Answer

- A large fraction of TSAD papers assume that deep learning is the answer, and their paper reduces to: *We will show that this variant of deep learning is better than those seven variants of deep learning*[2].
- However, because of the many reasons discussed above, I think that there is currently *zero* evidence that deep learning is SOTA for TSAD[1].
- Moreover, we should expect deep learning to have difficulties in this setting:
  - Few or no labeled examples
  - Very complex models, with relatively small datasets
  - For time series *classification*, which does have lots of labels, lots of data and lots of constraints, deep learning is only a little bit better than 50-year-old nearest neighbor classification with DTW.
- I am not saying deep learning *could* not work here. But I am willing to say that I do not think *any* current deep learning for TSAD methods can beat simpler decade old approaches. We certainly cannot assume this is the case.

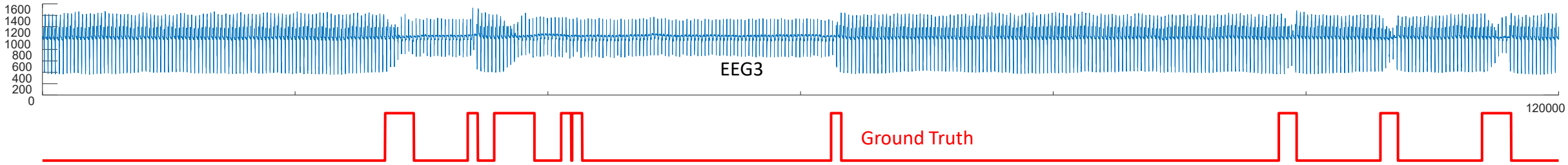
[1] I am not making any claims about deep learning in general

[2] DAEMON: Unsupervised Anomaly Detection and Interpretation for Multivariate Time Series

One of the best papers on deep learning TSAD I have read is the recent: *Temporal convolutional autoencoder for unsupervised anomaly detection in time series*.

Usually well written, strong reproducibly, some real insights.

However, at the end of the day, they have to learn or set a dozen parameters to create predictions for datasets like the below (their first dataset of 48 considered)

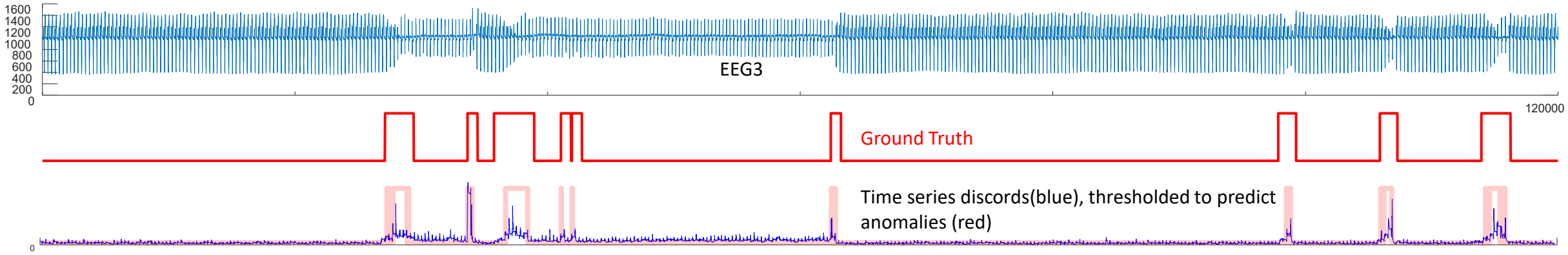



```
[matrixProfile, p, m, d]=interactiveMatrixProfileVer4_website((ecg3(1:120000,2)),300);,    plot(movmin(matrixProfile,64)>2.2,'r')
```

One of the best papers on deep learning TSAD I have read is the recent: *Temporal convolutional autoencoder for unsupervised anomaly detection in time series*.

Usually well written, *strong* reproducibly, some actual insights.

However, at the end of the day, they have to learn or set a dozen parameters to create predictions for datasets like the below (their first dataset of 48)

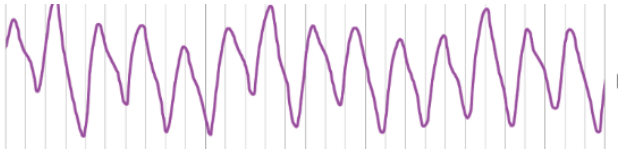


However, a 20-year-old, simple, fast, single-parameter, dozen-lines-of-code method seems to be *at least* competitive. What does the use of deep learning buy for us here?

```
[matrixProfile, p, m, d]=interactiveMatrixProfileVer4_website((ecg3(1:120000,2)),300);,     plot(movmin(matrixProfile,64)>2.2,'r')
```

Example of Deep Learning, but Shallow Thinking I

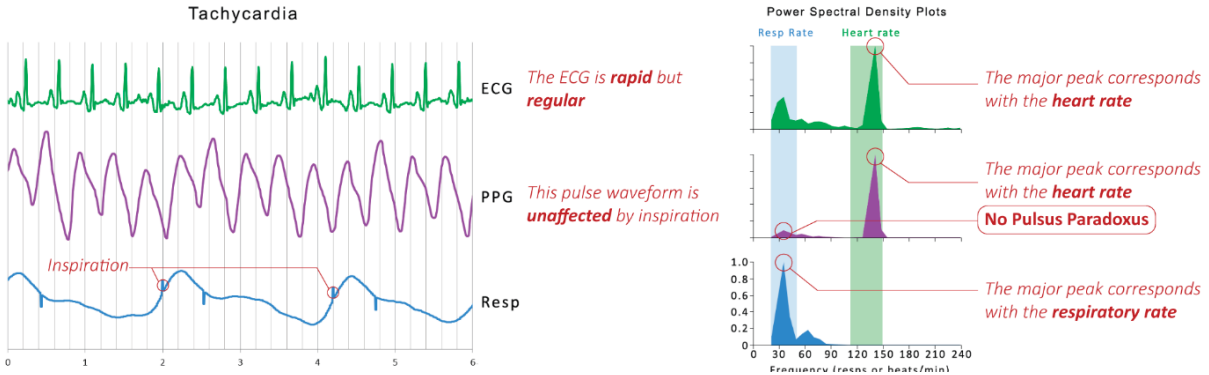
- At least two recent papers did the following
- Use deep learning to look at a PPG signal, and classify the user's behavior into walking/running etc.
- This is a stunning result!
- The result is attributes to various “magical” properties of deep learning.

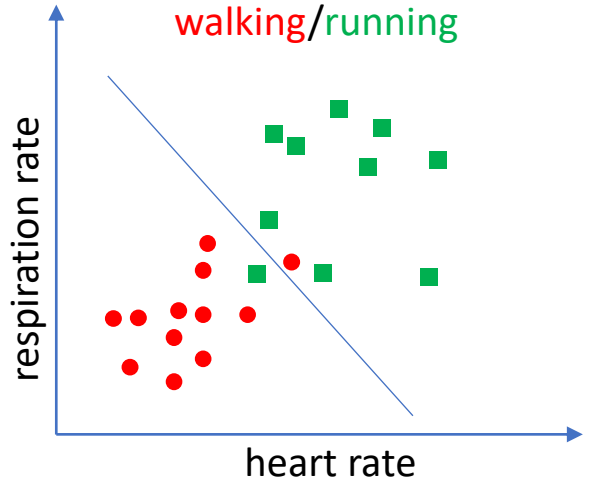


Example of Deep Learning, but Shallow Thinking II

- But wait a minute..
- If you have the PPG, you can trivially get the heart rate, and the respiration rate (as we have done for decades).
- If you treat it as a simple 2D problem feature, a linear classifier is much better!

Here deep learning is doing nothing magical. It is doing indirectly and expensively, what people have been doing *directly* for decades, and attributing it to the "magic" of deep learning.





Example of Deep Learning, but Shallow Thinking III

- I believe that *most* deep learning for TSAD papers are like the example on the last slide
- They are using complex deep learning to solve a problem that was solvable with much simpler methods decades ago.
- They are attributing their “success” to some vague “magic” of deep learning, not to the fact that the datasets they are working on are trivial!

Late Breaking News

- Two papers make similar observations in a slightly different time series context.

Do We Really Need Deep Learning Models for Time Series Forecasting?

Shereen Elsayed *†
elsayed@uni-hildesheim.de

Daniela Thyssens *†
thyssens@uni-hildesheim.de

Ahmed Rashed ‡
ahmedrashed@ismll.uni-hildesheim.de

Lars Schmidt-Thieme ‡
schmidt-thieme@ismll.uni-hildesheim.de

Deep Learning for Road Traffic Forecasting: Does it Make a Difference?

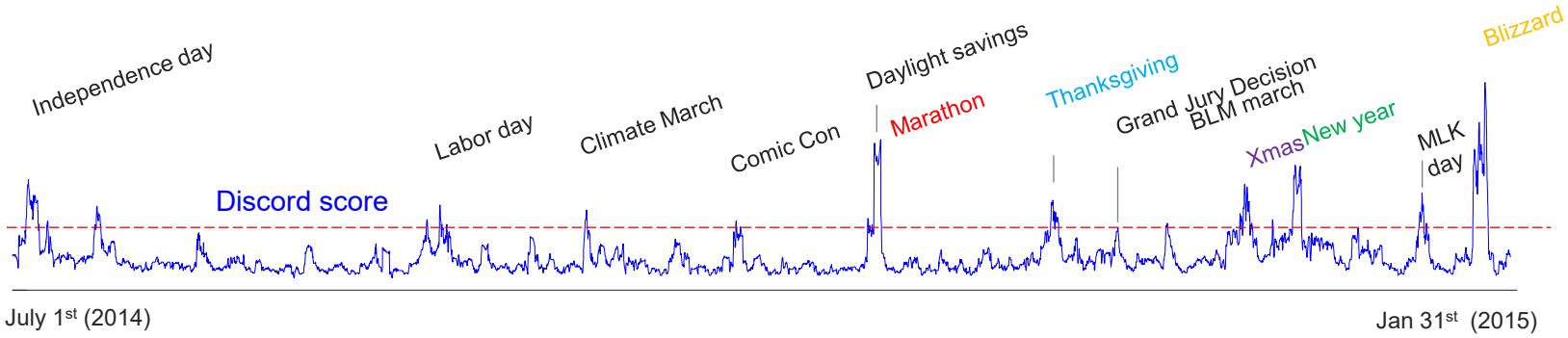
Eric L. Manibardo, Ibai Laña, and Javier Del Ser, *Senior Member, IEEE*

E. L. Manibardo, I. Laña and J. Del Ser, "Deep Learning for Road Traffic Forecasting: Does It Make a Difference?," in *IEEE Transactions on Intelligent Transportation Systems* (Early access), 2021.

Stabbing William of Ockham in the Heart

Recall the NY Taxi dataset.

Up to the limit of subjectively of labels, it is clear that we can find all the anomalies with simple single-parameter method, for example the Matrix Profile.



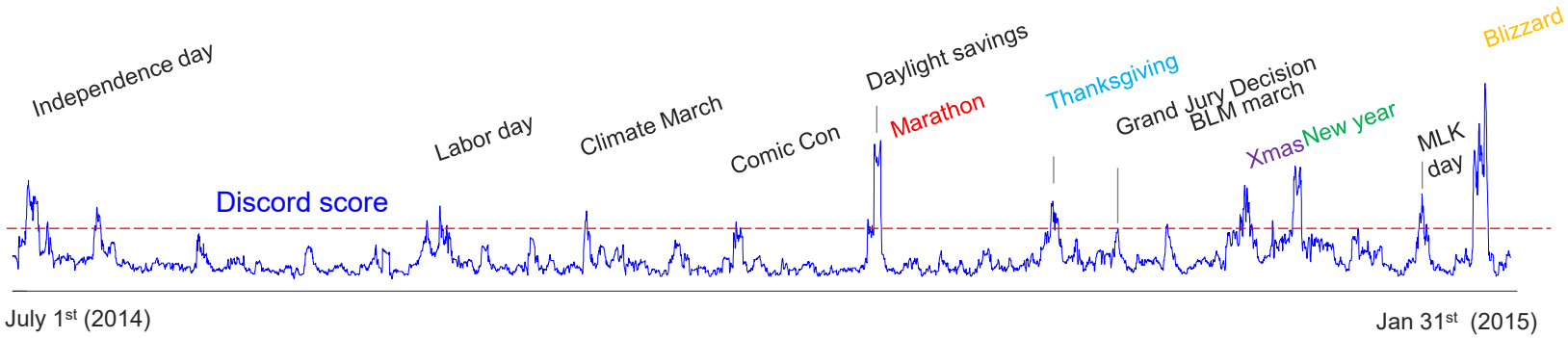
Stabbing William of Ockham in the Heart I

Recall the NY Taxi dataset.

Up to the limit of subjectively of labels, it is clear that we can find all the anomalies with simple single-parameter method like time series discords, that has been around for 20 years.

Yet there are dozens of papers that have to learn or set five or more parameters to do the same (or a worse) job on this dataset.

What does that achieve?



Data Sets	LSTM Architecture
NYC Taxi Demand	2 Recurrent layers: {50, 20}, Dropout: 0.4, 1 Dense layer:{24}, Learning rate: 0.0001

A framework for end-to-end deep learning-based anomaly detection in transportation networks

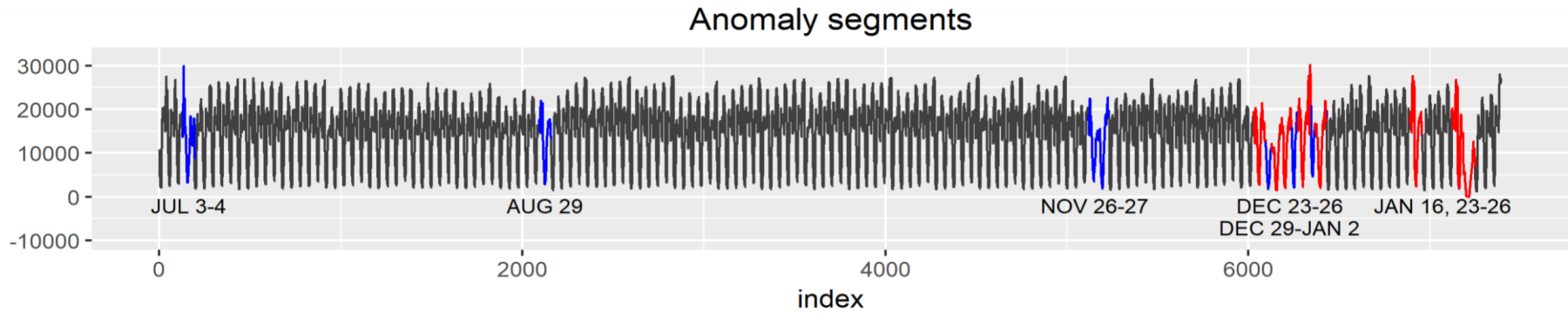
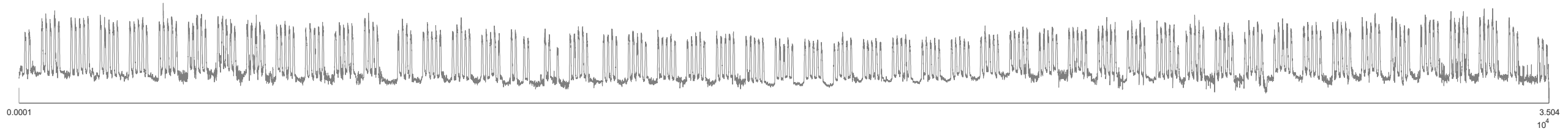


Fig. 6. Experiment for the New York City taxi demand dataset. The top plot shows the expected log-likelihood, and the bottom plot shows anomaly segments in red. The blue segments generally do not firmly represent anomalies and need further analysis. In this case, except for August 29, all blue segments refer to national holidays whose patterns are anomalous. The parameters for this experiment are $w = 30$, $k = 6$, $q = 5$, $h_1 = -3.57$, and $h_2 = -4.28$.

Time series anomaly detection from a Markov chain perspective ICMLA 2019

Stabbing William of Ockham in the Heart II

This paper proposes a *very* complex method "*adaptive anomaly detection .. hierarchical edge computing… multiple anomaly detection DNN models with increasing complexity… adaptive model selection scheme … contextual-bandit problem … reinforcement learning policy network*." [a].
I could not count all the parameters set, but clear more than a dozen.
To evaluate it, they use the dataset below and say…
"*We manually label a day as abnormal if it is a weekday with low power consumption*".

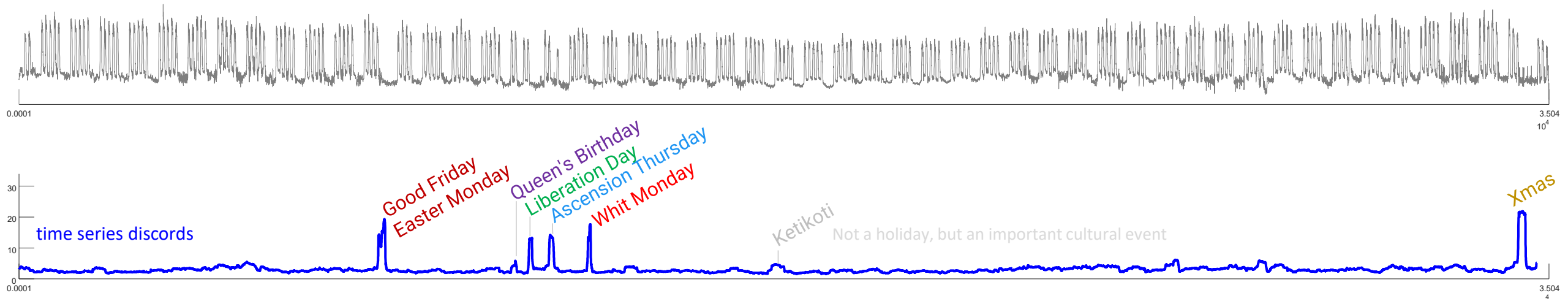


[a] Mao V. Ngo, Tie Luo, Hakima Chaouchi, Tony Q. S. Quek: Contextual-Bandit Anomaly Detection for IoT Data in Distributed Hierarchical Edge Computing. ICDCS 2020: 1227-1230

Stabbing William of Ockham in the Heart II

This paper proposes a very complex method "*adaptive anomaly detection .. hierarchical edge computing… multiple anomaly detection DNN models with increasing complexity… adaptive model selection scheme … contextual-bandit problem … reinforcement learning policy network.*" [a].
I could not count all the parameters set, but clear more than a dozen.
To evaluate it, they use the dataset below and say…
"*We manually label a day as abnormal if it is a weekday with low power consumption*".

It is a very nicely written paper. But at the end of the day, it is clear that we can find all the anomalies with simple single-parameter method like time series discords, that has been around for 20 years.



[a] Mao V. Ngo, Tie Luo, Hakima Chaouchi, Tony Q. S. Quek: Contextual-Bandit Anomaly Detection for IoT Data in Distributed Hierarchical Edge Computing. ICDCS 2020: 1227-1230

Stabbing William of Ockham in the Heart III

Ockham's razor is perhaps the most fundamental principle in all of science.

In essence, *we should prefer the simplest solution to a problem*.

However, it is clear that some TSAD papers are proposing solutions that are orders of magnitude more complex than the need to be, given the data they examine.

The solution?

- Test on non-trivial datasets that warrant all this complexity.
- Stop writing this overcomplicated papers.

Not doing anomaly detection, and calling it anomaly detection

en.wikipedia.org/wiki/Streetlight_effect

Several papers say something like…

- "*We manually label a day as abnormal if it is a weekday with low power consumption*" [a]
- "*A week when any of the first 5 days has low power demands is considered anomalous*" [b]
- "*we considered anomalies as weekdays that have low level on power consumption*" [c]

But think about it. If you can concretely define in a single English sentence what you expect to find in advance, is that really anomaly detection?

Anomaly detection is meant to be "*expect the unexpected*", but given the above, I could just do similarity search, which is a much easier problem.

[a] Mao V. Ngo, Tie Luo, Hakima Chaouchi, Tony Q. S. Quek: Contextual-Bandit Anomaly Detection for IoT Data in Distributed Hierarchical Edge Computing. ICDCS 2020: 1227-1230
[b] LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. Malhotra et al. ICML 2016
[c] LSTM-based Anomaly Detection on Big Data for Smart Factory Monitoring. Van Quan Nguyen 2018

Switching Gears…

- Thus far this talk has been negative, sorry
- Let me switch to a small positive contribution I can make
- Many of the problems I noted derived from poor datasets and poor measures of success (spurious precision etc.)
- I have tried to make some contribution here…

UCR/HEX Anomaly Benchmark Datasets 2021

- I have created a set of 250 time series anomaly detection benchmark datasets.

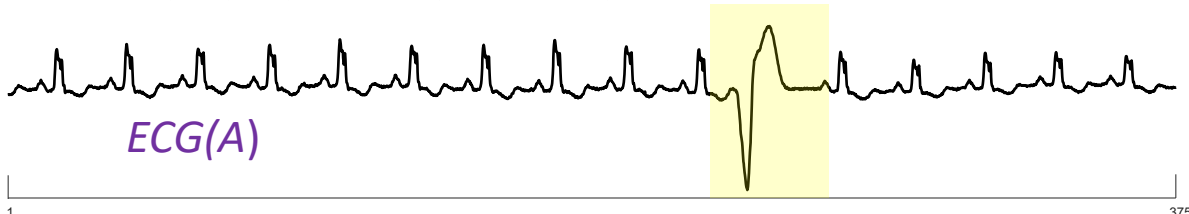
- While not perfect, I hope that they will allow for more meaningful empirical work by the community.

[1]To *reasonable*, but not *perfect* degree of certainty

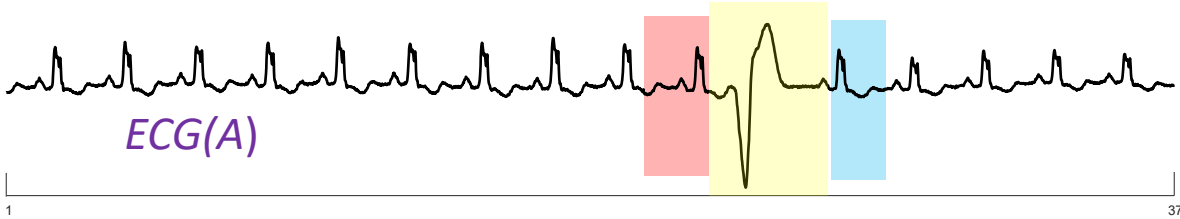
The Rationale for the Datasets

- Given the previously noted issues, we ask if we could we create a better data collection?
  - Avoiding Triviality, Mislabeling, Run-to-failure bias and Unrealistic anomaly density (*some* "trivial" datasets are OK to have a spectrum of difficulty).
  - Avoid issues in scoring by making the discovery of an anomaly be a *binary* event.
  - Have lots of datasets, so the sum of many binary events is a real number (the percentage correct) that discriminates among competing ideas.
  - Some algorithms may be biased to predict the beginning or the end of an anomaly. In any case, the exact beginning or end might not be well defined. Let's bypass the issue, by adding some "slop" before and after the anomaly. This barely effect the default rate and avoids penalizing otherwise very accurate algorithms.

Somehow people get four significant digits out of this…



True anomaly is in yellow region, but prediction in any part of the highlighted region is marked as correct. You score one binary digit.



The Rationale for the Challenge Design

- Anomaly discovery methods typically have two parts
  - (a) finding the region(s) most likely to be an anomaly
  - (b) using some “threshold” to predict if it really *is* an anomaly.
- We may be wrong, but we see ‘a’ as being the hardest part of the challenge. In any case, ‘b’ can sometimes depend on some out-of-band information.
- We can completely remove the ‘b’ question by having *exactly* one anomaly per dataset, and telling people there is *exactly* one anomaly per dataset.
- If we do have binary scores per dataset, we are going to need to have lots of datasets, in order to have discrimination among teams.
- How do we get lots of datasets?? (next slide)

Getting Datasets

- We posted a call for datasets in Reddit/ML and in Dbworld
- We wrote to essentially everyone that published a paper in SIGKDD, ICDM, ICDE, SDM, VLDB, SIGMOD, NeurIPS in the last five years, on the topic of time series anomaly detection[1].
- We wrote to essentially everyone that cited **Yahoo**, **Numenta**, **SMAP**, **MSL**, **SDM**, **MBA-ECG** **SWAT** (using Google scholar).
- These efforts yielded zero datasets.

- Thus, I created the datasets myself (with some help from my students) using datasets I have collected or created over 20 years.

[1]To be fair, we might have missed a paper or two, and we did not aggressively pursue bounce emails etc.

These datasets…

- Largely solve the *mislabeling* problem[1]
- Solve the *triviality*[2]/*run-to-failure*/unrealistic *density problems*
- Completely solve the *spurious precision* problem

- I am not going to go thru all 250 datasets, but let us see an example to get a flavor

[1]To *reasonable*, but not *perfect* degree of certainty. [2]A small subset are trivial, to allow a spectrum of difficulty

012_UCR_Anomaly_**tiltAPB1**_100000_114283_114350.txt

- Dataset number
- Mnemonic name
- From 1 to **X** is training data
- **Begin** anomaly
- **End** anomaly

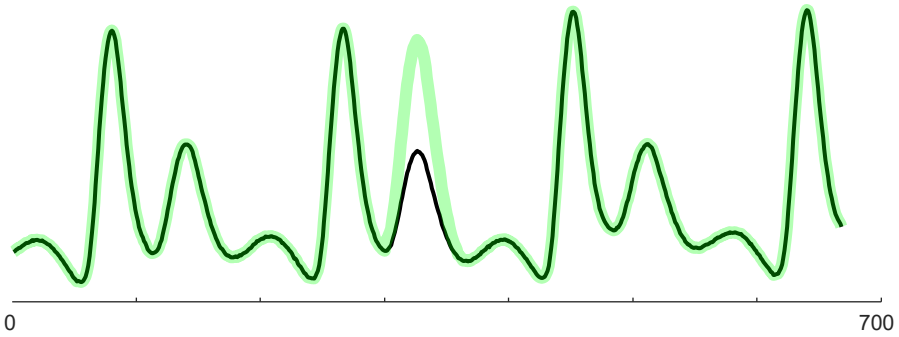
Sample format

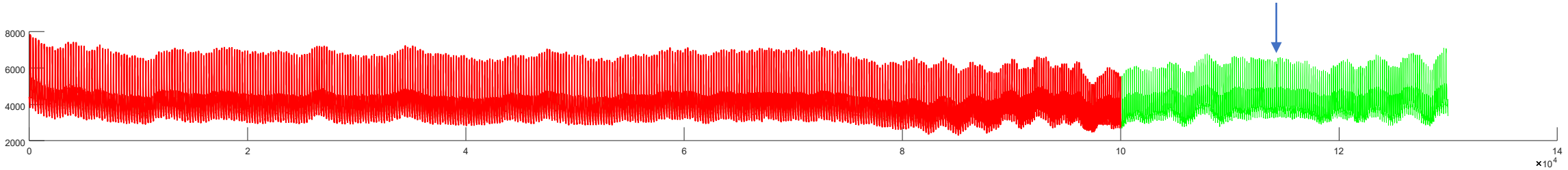
Note the structure of the file names
Files are '-ascii' format.
Note the layout of the plots

The data comes from a healthy male on a tilt table.
At first, he is supine, at around 80000, the table is tilted forward. The trace is his APB.

The anomaly is synthetic. There is a secondary peak after the dicrotic notch. It is normally about half the size of the peak systolic pressure. For one randomly chosen beat, we made it much greater, almost as big as the main peak.

Black is original data; green is data after anomaly was introduced





UCR_Anomaly_gaitHunt1_18500_33070_33180.txt

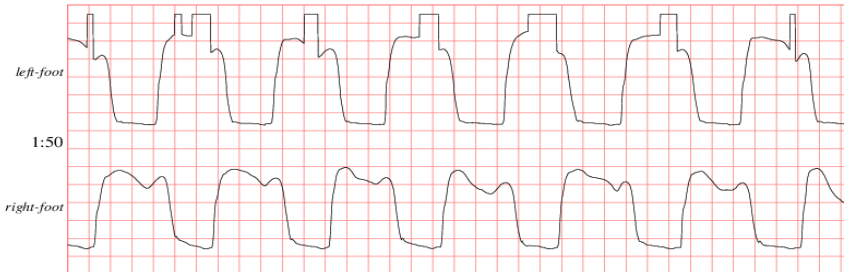
Selected input: record gaitndd/hunt17 , from 0:00.000 to 5:00.000

Gait in Neurodegenerative Disease Database (gaitndd)

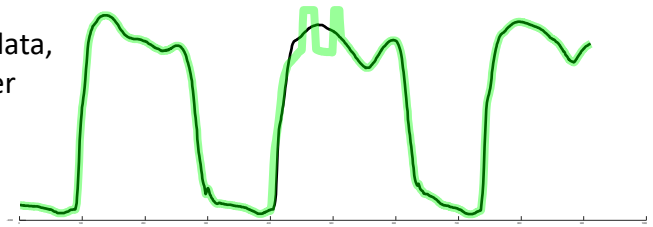
This dataset comes from someone walking on a force plate in a biomechanics lab.
The individual had a mostly symmetric gait, however, after some time, the left foot sensor developed a fault.

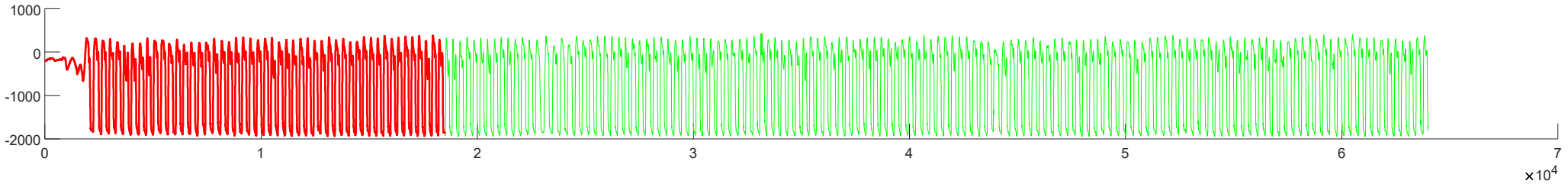
This allows us to create an almost 100% natural dataset. We simply took faulty data from the left foot, and used it to replace some right foot data. We shifted it by a half cycle, slightly smoothing it to remove cut and paste artifacts, and reduced it amplitude so it was not necessarily the tallest peak.

Fault in left sensor



Black is original data, green is data after anomaly was introduced





Conclusions I

- I hope I have convinced you, 95% of papers on TSAD make no contribution
  - These papers are:
    - **In the *best* case**: Solving problems a hard way, that we could have solved an easy way, 20 years ago.
    - **In the *worse* case**: Contriving experiments and cherry-picking to make an intrinsically bad ideas look good.

Conclusions II

- We should see these facts as a wonderful opportunity
- We should have more introspection as researchers.
    - We need to *think* about problems, what are we trying to do?
    - We need to *think* about evaluation, what counts as evidence of success?
- We should be more (*constructively*) critical as reviewers.
    - Demand 100% reproducibility. All code, all (carefully annotated) data, all parameter settings/seeds
    - Insist on common sense baselines
    - The paucity of good datasets is partly to blame, we should reward those who are willing to do the hard work of creating good datasets and making them public.

Questions?

The new datasets will be linked from www.cs.ucr.edu/~eamonn/time_series_data_2018/

Backup slides

Dataset Design Principles I

- **Remove the threshold question**. Anomaly detection typically consists of two parts. A) Find the candidate region(s) that *might be* anomalies. B) Test if, under your model, you should flag these as anomalies. Here we remove 'B', by telling the world that there is *exactly one anomaly* in the test data. We do this because 'B' can depend on external factors (misclassification costs etc.), and we think that in most domains, if you can do 'A' robustly, 'B' will be easy.

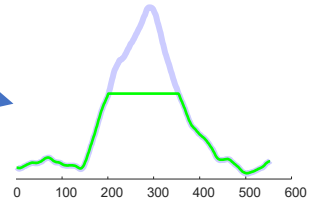
- **Try to have diverse datasets**.

- **Use real data only in the case** you can be sure the anomaly is the *only* (or by a large margin, *most significant*) anomaly.

- **For synthetic data, model something in the real world** (when possible). For example:
  - This anomaly models a nurse placing her hand under the respiration strap.

- **Avoid Goldilocks**. For at least some problems, make multiple versions….
  - One that is obvious, probably *any* algorithm can find them.
  - One that is more subtle
  - One that is very very subtle, probably *no* algorithm can find them.

- **Implication of the above**: It is virtually certain that 100% accuracy is not possible.



This anomaly models a nurse placing her hand under the respiration strap

UCR_Anomaly_resperation1_100000_110260_110412.txt

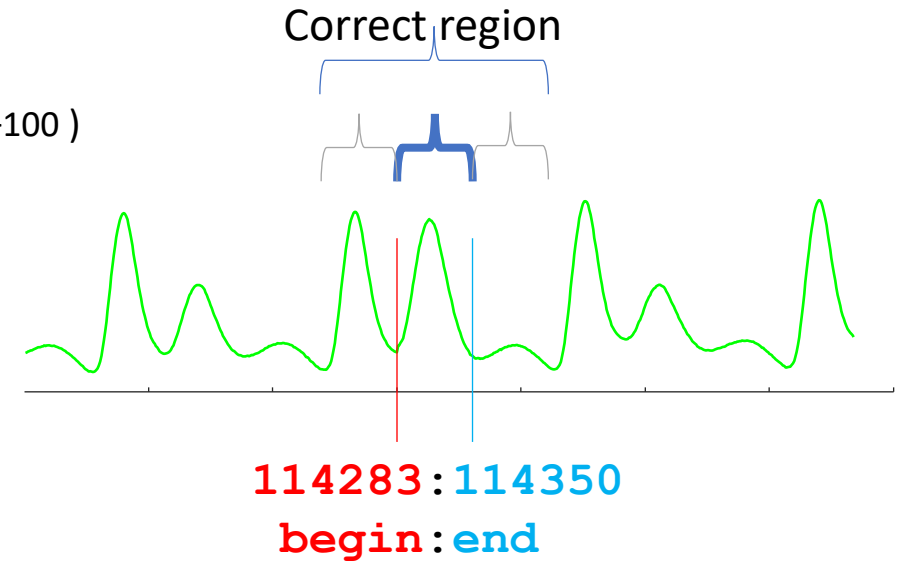
Scoring Function Design Principles I

Avoid complex and opaque scoring functions We want a scoring function that..

- Is a single number, for easy comparisons.
- Does not have spurious location precision. If the ground truth says the anomaly is at say 1250, and an algorithm reports 1247 or 1254, it should be counted as correct. This problem is compound by the fact that different algorithms report the *leading edge*, the *center* or the *trailing edge* of a sliding window.
- Has a binary score for each example, that can be combined to a real number for the full collection.
- Reports a number close to zero for a "random dart" algorithm (i.e. the default rate) and close to one for a perfect algorithm.

My suggestion

- Let length of anomaly be L, $L = \texttt{end} - \texttt{begin}$
- Let the prediction of an algorithm be an integer P
- P is labeled as correct if: $\min(\texttt{begin}-L, \texttt{begin}-100) < P < \max(\texttt{end}+L, \texttt{end}+100)$
- Why the '100' case? Some anomalies can be as short as a single point.



For this collection of datasets, the only meaningful score is something like "207 out of 250"

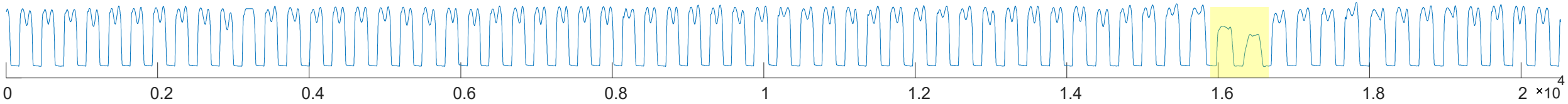
Is Deep Learning really useless for Time Series?

- **Maybe not**: Perhaps no one has figured out how to do properly yet, but one day soon a mind-blowing paper will appear. (And I would be first to champion it)
- Or perhaps that paper is *already* out, but I have foolishly dismissed it, because I find the experiments unconvincing, or because I am just stupid.

- **Maybe yes**: No one expects deep learning to have an impact on say *sorting numbers*. Maybe time series problems are so simple that they cannot benefit from whatever it is that deep learning does.
- **Maybe yes**: Time series is a little unusual in that we have near perfect distance measure in DTW (which includes Euclidean Distance as a special case). Maybe, given a strong distance measure, nothing else really matters.

Ground Truth Labels are Impossible for Anomaly Detection!

- For some ML problems, we *can* get perfect ground truth, i.e., cats vs dogs
- However, for anomaly detection, we can *never* have perfect ground truth.
- Consider the example below, where is the anomaly?



- Surely it is at the highlighted region?
- No, anyone that has worked in a biomechanical lab has seen this many times, it is the patient turning around at the end of the forceplate apparatus.
- The anomaly is at 16,000, the lack of a heel strike, which *is* unusual.

I have tried and tried to tell folks that if the underlying uncertainty in your labels is larger than any change in relative performance, the change is meaningless

Vijayant K. VP of Product: ML & AI at Optum