Detecting system failure: Signal Processing + ML on LEM dataset

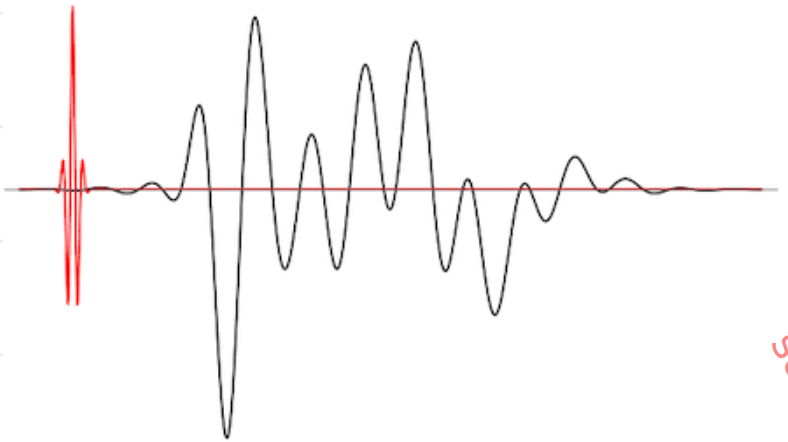
Group 7:

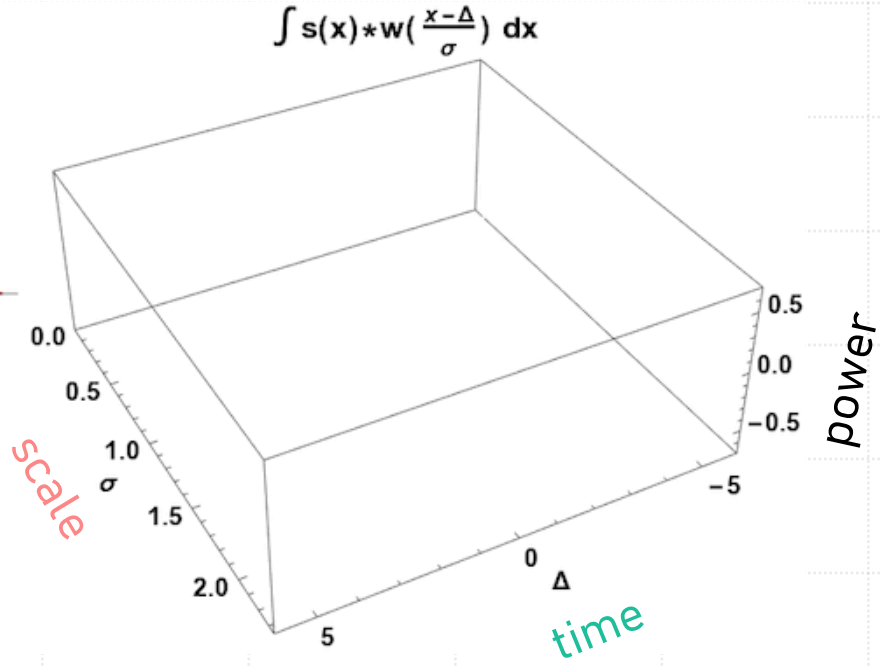
Sadiya Bhawania,

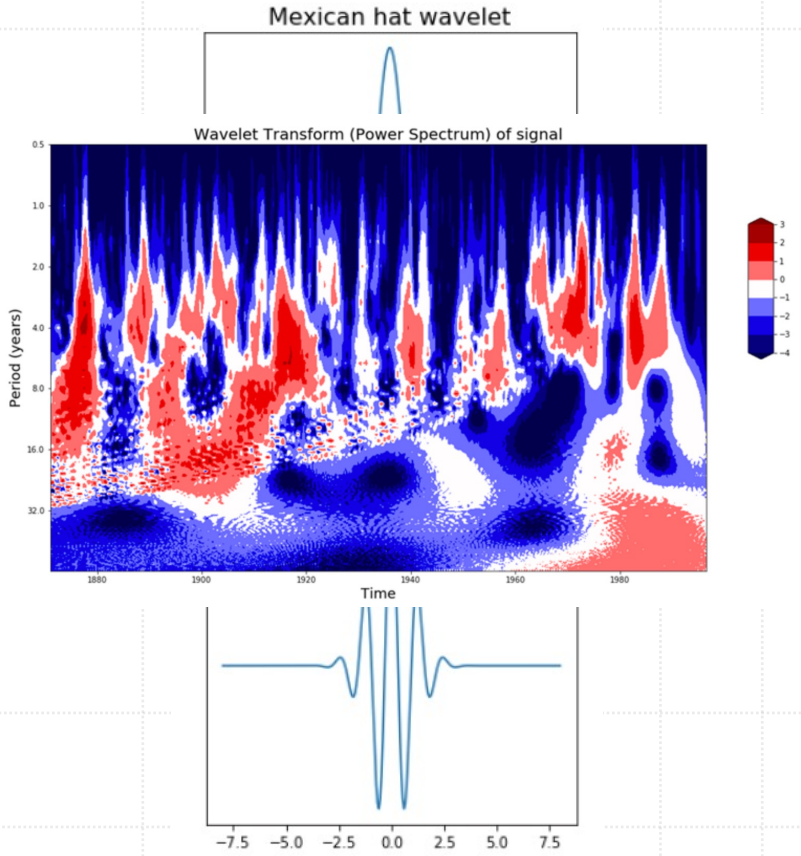
Katerina Bosko,

Misha Mody

Signal Processing with Continuous Wavelet Transformation

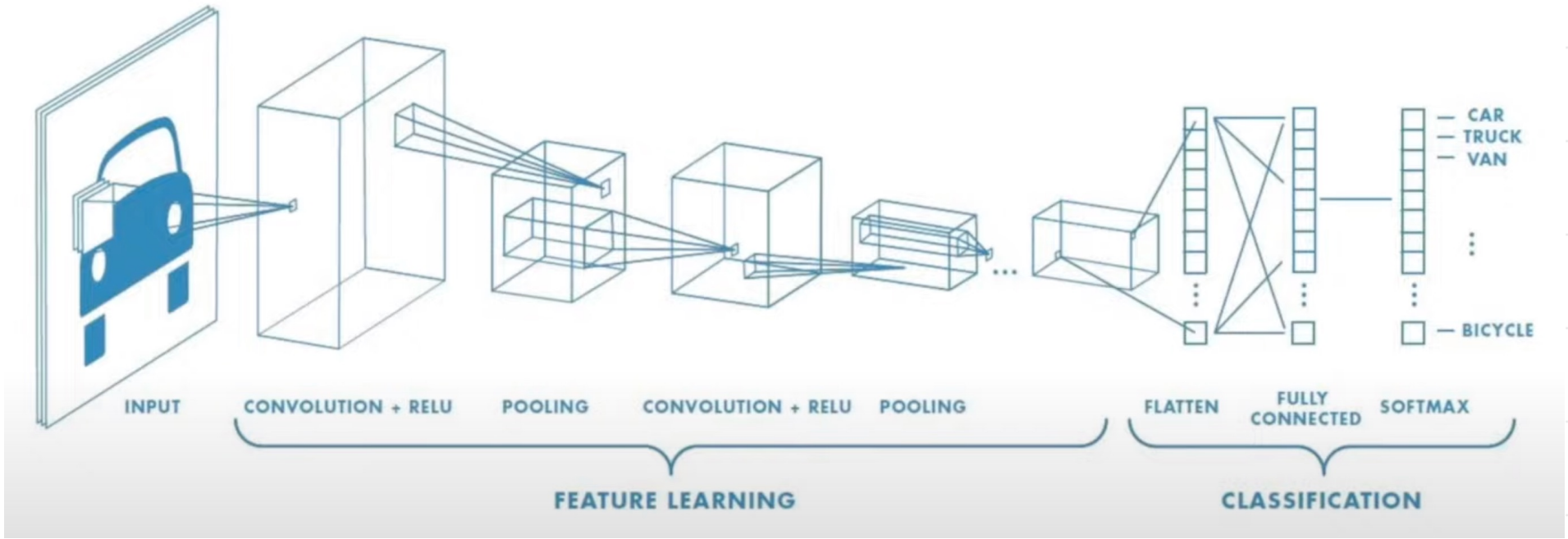






Source: https://handwiki.org/wiki/Continuous_wavelet_transform (left)
https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/ (right)

Convolutional Neural Networks

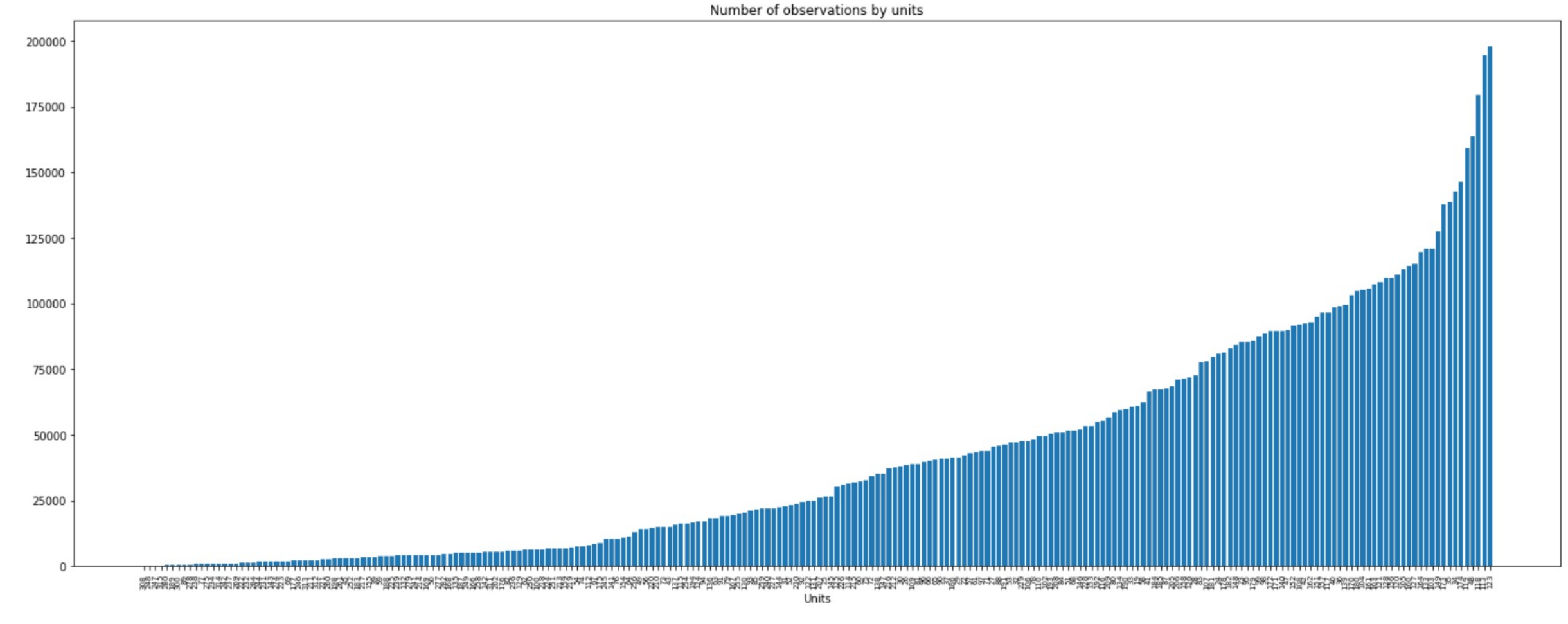


Source: https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e (first), Dataquest course on CNN, https://www.youtube.com/watch?v=9xcky8jXQAM&t=3s (second)

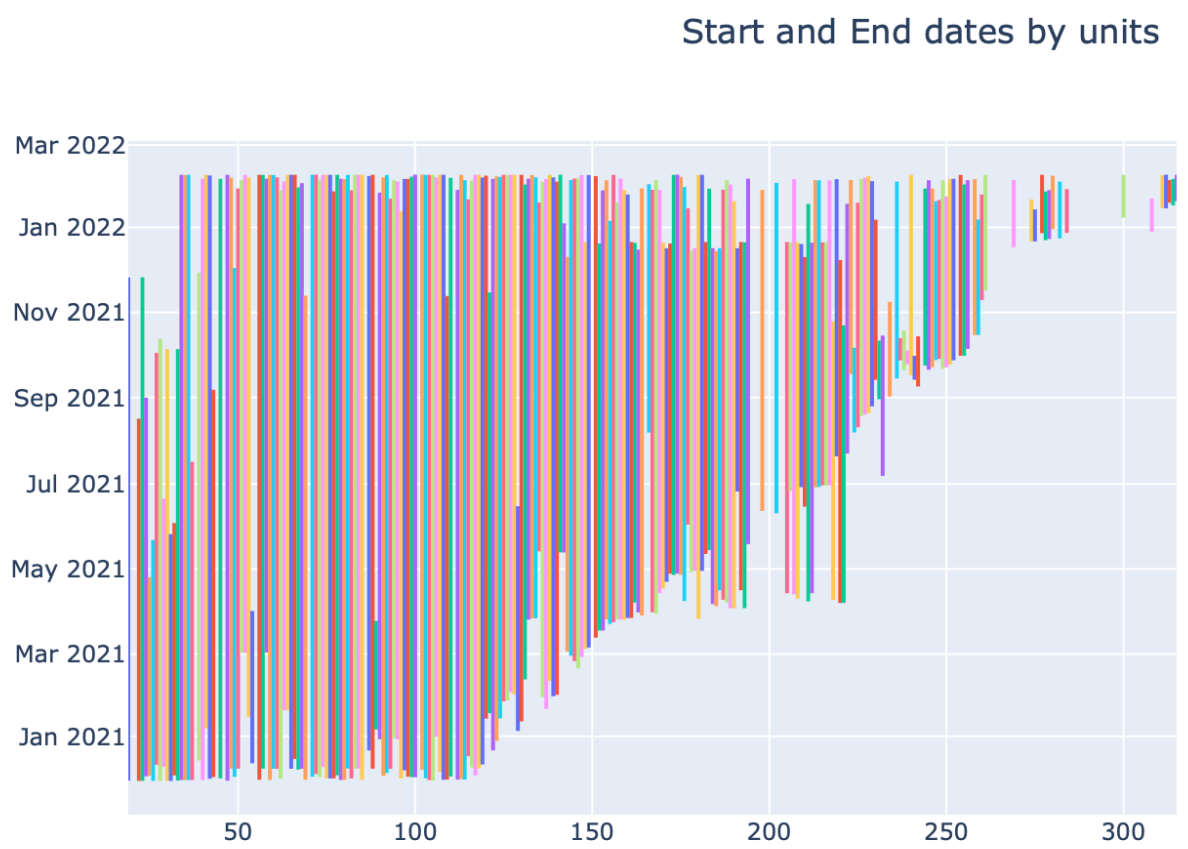
Methodology

1. Data Preprocessing
2. Operationalization of “system failure”
3. Signal Processing with Continuous Wavelet Transform(CWT)
4. Convolutional Neural Network on scaleograms

Data Preprocessing Challenge

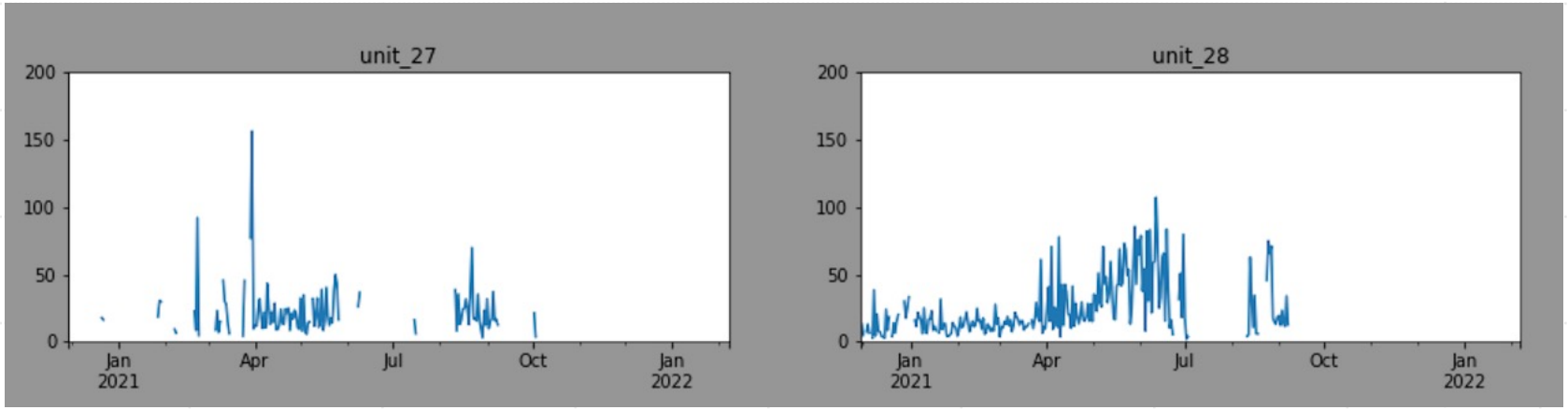


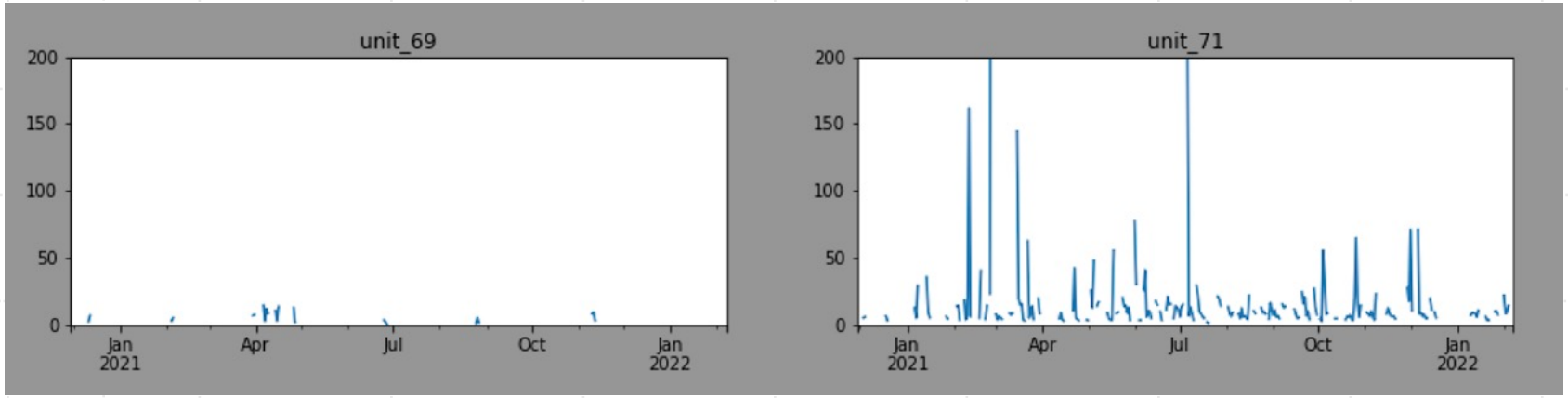
Data Preprocessing Challenge



Data Preprocessing Challenge

- Daily mean m_vs by units:





Challenge:

How do you create continuous time series ready for ML algorithms?

5 step solution → continuous segments of given size

Nov 29, 2020

1. Timeframe

Feb 07, 2022

	unit_19	unit_22	unit_23	unit_24	unit_25	unit_26	unit_27	unit_28	unit_29	unit_30	unit_31
2020-11-29 22:00:00											
2020-11-29 23:00:00		8.00	10.50							6.33	
2020-11-30 00:00:00		8.96	11.75			6.76		6.00		6.37	17.61
2020-11-30 01:00:00		5.56				15.00				9.05	
2020-11-30 02:00:00		8.72				5.43				5.18	
2020-11-30 03:00:00		5.40				7.82				5.00	
2020-11-30 04:00:00		5.10				4.98				6.75	
2020-11-30 05:00:00		10.51				5.13				4.62	
2020-11-30 06:00:00	2.50					6.15		8.73		3.70	
2020-11-30 07:00:00						4.87		5.43		3.23	
2020-11-30 08:00:00						3.58		4.52		2.60	
2020-11-30 09:00:00						3.30		5.05			
2020-11-30 10:00:00						4.68		5.68			
2020-11-30 11:00:00								9.53			
2020-11-30 12:00:00								4.97			
2020-11-30 13:00:00								4.72			
2020-11-30 14:00:00								6.09		22.75	
2020-11-30 15:00:00			32.95							33.67	
2020-11-30 16:00:00			20.58					3.33		37.57	
2020-11-30 17:00:00			22.84			14.69		2.86		5.20	
2020-11-30 18:00:00	3.40		6.29			18.98				28.20	
2020-11-30 19:00:00			4.68			14.65		9.07		27.05	
2020-11-30 20:00:00			21.67			7.55		13.93		9.86	
2020-11-30 21:00:00						2.82		332.25		14.63	
2020-11-30 22:00:00			18.00			3.00				18.43	
2020-11-30 23:00:00	2.00					2.58		4.00		5.00	
2020-12-01 00:00:00						1.67		5.09		13.60	
2020-12-01 01:00:00								5.27		6.12	
2020-12-01 02:00:00								7.07		8.58	

2. Hourly mean m_vs

4. Chop into segment

	unit_19	unit_22	unit_23	unit_24	unit_25	unit_26	unit_27	unit_28	unit_29	unit_30	unit_31
2020-11-29 22:00:00	0	0	0	0	0	0	0	0	0	0	0
2020-11-29 23:00:00	0	1	1	0	0	0	0	0	0	1	0
2020-11-30 00:00:00	0	2	2	0	0	1	0	1	0	2	1
2020-11-30 01:00:00	0	3	0	0	0	2	0	0	0	3	0
2020-11-30 02:00:00	0	4	0	0	0	3	0	0	0	4	0
2020-11-30 03:00:00	0	5	0	0	0	4	0	0	0	5	0
2020-11-30 04:00:00	0	6	0	0	0	5	0	0	0	6	0
2020-11-30 05:00:00	0	7	0	0	0	6	0	0	0	7	0
2020-11-30 06:00:00	1	0	0	0	0	7	0	1	0	8	0
2020-11-30 07:00:00	0	0	0	0	0	8	0	2	0	9	0
2020-11-30 08:00:00	0	0	0	0	0	9	0	3	0	10	0
2020-11-30 09:00:00	0	0	0	0	0	10	0	4	0	0	0
2020-11-30 10:00:00	0	0	0	0	0	11	0	5	0	0	0
2020-11-30 11:00:00	0	0	0	0	0	0	0	6	0	0	0
2020-11-30 12:00:00	0	0	0	0	0	0	0	7	0	0	0
2020-11-30 13:00:00	0	0	0	0	0	0	0	8	0	0	0
2020-11-30 14:00:00	0	0	0	0	0	0	0	9	0	1	0
2020-11-30 15:00:00	0	0	1	0	0	0	0	0	0	2	0
2020-11-30 16:00:00	0	0	2	0	0	0	0	1	0	3	0
2020-11-30 17:00:00	0	0	3	0	0	1	0	2	0	4	0
2020-11-30 18:00:00	1	0	4	0	0	2	0	0	0	5	0
2020-11-30 19:00:00	0	0	5	0	0	3	0	1	0	6	0
2020-11-30 20:00:00	0	0	6	0	0	4	0	2	0	7	0
2020-11-30 21:00:00	0	0	0	0	0	5	0	3	0	8	0
2020-11-30 22:00:00	0	0	1	0	0	6	0	0	0	9	0
2020-11-30 23:00:00	1	0	0	0	0	7	0	1	0	10	0
2020-12-01 00:00:00	0	0	0	0	0	8	0	2	0	11	0
2020-12-01 01:00:00	0	0	0	0	0	0	0	3	0	12	0
2020-12-01 02:00:00	0	0	0	0	0	0	0	4	0	13	0

3. Count continuous points

	('unit_19', 0)	('unit_19', 1)	('unit_23', 0)	('unit_23', 1)	('unit_23', 2)	('unit_23', 3)
0	2021-04-30 19:00:00	2021-06-25 16:00:00	2020-12-02 21:00:00	2021-01-08 02:00:00	2021-01-11 14:00:00	2021-01-16 02:00:00
1	2021-04-30 20:00:00	2021-06-25 17:00:00	2020-12-02 22:00:00	2021-01-08 03:00:00	2021-01-11 15:00:00	2021-01-16 03:00:00
2	2021-04-30 21:00:00	2021-06-25 18:00:00	2020-12-02 23:00:00	2021-01-08 04:00:00	2021-01-11 16:00:00	2021-01-16 04:00:00
3	2021-04-30 22:00:00	2021-06-25 19:00:00	2020-12-03 00:00:00	2021-01-08 05:00:00	2021-01-11 17:00:00	2021-01-16 05:00:00
4	2021-04-30 23:00:00	2021-06-25 20:00:00	2020-12-03 01:00:00	2021-01-08 06:00:00	2021-01-11 18:00:00	2021-01-16 06:00:00
5	2021-05-01 00:00:00	2021-06-25 21:00:00	2020-12-03 02:00:00	2021-01-08 07:00:00	2021-01-11 19:00:00	2021-01-16 07:00:00
6	2021-05-01 01:00:00	2021-06-25 22:00:00	2020-12-03 03:00:00	2021-01-08 08:00:00	2021-01-11 20:00:00	2021-01-16 08:00:00
7	2021-05-01 02:00:00	2021-06-25 23:00:00	2020-12-03 04:00:00	2021-01-08 09:00:00	2021-01-11 21:00:00	2021-01-16 09:00:00
8	2021-05-01 03:00:00	2021-06-26 00:00:00	2020-12-03 05:00:00	2021-01-08 10:00:00	2021-01-11 22:00:00	2021-01-16 10:00:00
9	2021-05-01 04:00:00	2021-06-26 01:00:00	2020-12-03 06:00:00	2021-01-08 11:00:00	2021-01-11 23:00:00	2021-01-16 11:00:00
10	2021-05-01 05:00:00	2021-06-26 02:00:00	2020-12-03 07:00:00	2021-01-08 12:00:00	2021-01-12 00:00:00	2021-01-16 12:00:00
11	2021-05-01 06:00:00	2021-06-26 03:00:00	2020-12-03 08:00:00	2021-01-08 13:00:00	2021-01-12 01:00:00	2021-01-16 13:00:00
12	2021-05-01 07:00:00	2021-06-26 04:00:00	2020-12-03 09:00:00	2021-01-08 14:00:00	2021-01-12 02:00:00	2021-01-16 14:00:00
13	2021-05-01 08:00:00	2021-06-26 05:00:00	2020-12-03 10:00:00	2021-01-08 15:00:00	2021-01-12 03:00:00	2021-01-16 15:00:00
14	2021-05-01 09:00:00	2021-06-26 06:00:00	2020-12-03 11:00:00	2021-01-08 16:00:00	2021-01-12 04:00:00	2021-01-16 16:00:00
15	2021-05-01 10:00:00	2021-06-26 07:00:00	2020-12-03 12:00:00	2021-01-08 17:00:00	2021-01-12 05:00:00	2021-01-16 17:00:00
16	2021-05-01 11:00:00	2021-06-26 08:00:00	2020-12-03 13:00:00	2021-01-08 18:00:00	2021-01-12 06:00:00	2021-01-16 18:00:00
17	2021-05-01 12:00:00	2021-06-26 09:00:00	2020-12-03 14:00:00	2021-01-08 19:00:00	2021-01-12 07:00:00	2021-01-16 19:00:00
18	2021-05-01 13:00:00	2021-06-26 10:00:00	2020-12-03 15:00:00	2021-01-08 20:00:00	2021-01-12 08:00:00	2021-01-16 20:00:00
19	2021-05-01 14:00:00	2021-06-26 11:00:00	2020-12-03 16:00:00	2021-01-08 21:00:00	2021-01-12 09:00:00	2021-01-16 21:00:00

Segments' mask for independent variables

Hourly mean m_vs segments:

	(19, '2021/04/30', '20	(19, '2021/06/25', '20	(23, '2020/12/02', '20	(23, '2021/01/08', '20	(23, '2021/01/11', '20	(23, '2021/01/16', '20
0	33.75	68.00	15.17	11.54	12.86	13.56
1	29.67	29.00	17.52	11.52	29.53	11.16
2	33.80	39.20	14.15	9.07	18.47	10.83
3	18.56	19.50	6.36	6.48	8.50	7.88
4	11.36	17.72	6.90	7.65	5.05	6.49
5	17.17	19.67	5.70	4.72	11.95	6.15
6	15.50	60.75	20.22	4.28	13.33	4.87
7	41.08	38.29	20.38	4.97	16.94	5.38
8	6.96	34.67	4.33	10.23	14.93	5.00
9	6.02	26.00	4.53	2.87	18.06	4.50
10	5.65	7.56	4.63	4.75	15.25	3.75
11	9.45	8.72	3.20	5.00	19.90	2.58
12	5.28	11.14	3.25	14.89	11.93	5.24
13	4.88	9.16	4.00	45.33	9.55	35.14
14	5.22	8.63	4.20	18.84	6.97	16.25
15	3.83	12.31	5.20	28.13	7.23	8.06
16	1.23	6.56	6.00	11.14	5.20	27.15
17	1.43	2.28	20.27	33.20	15.17	12.62
18	1.57	2.48	37.00	20.00	8.98	18.00
19	1.42	5.62	34.17	11.75	3.27	12.00

5. Apply segments mask onto variables

2472 segments

Classification

1. For each segment find percentage of [hourly mean] m_vs values that are larger than certain threshold – 50 mV
2. Classify segments with >= 10% of values as “bad” segments

“bad” segment

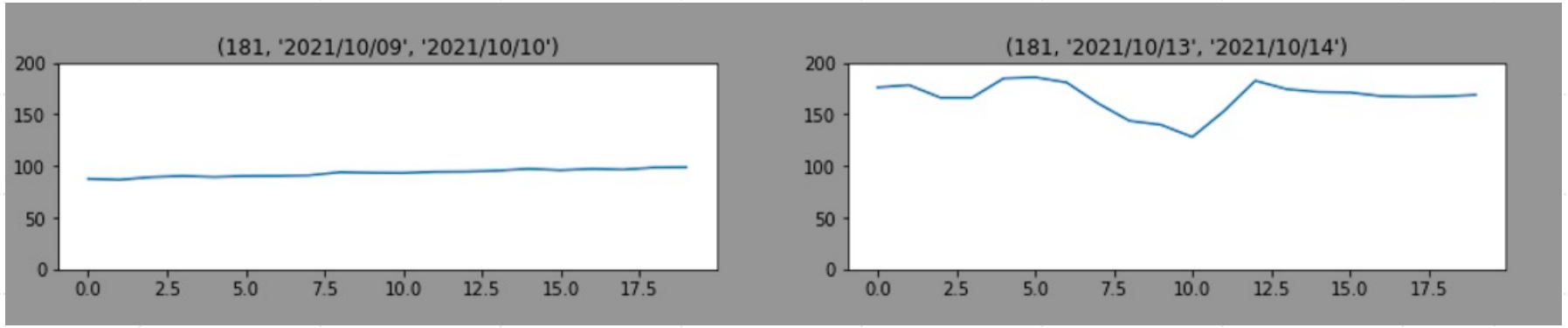
⇩

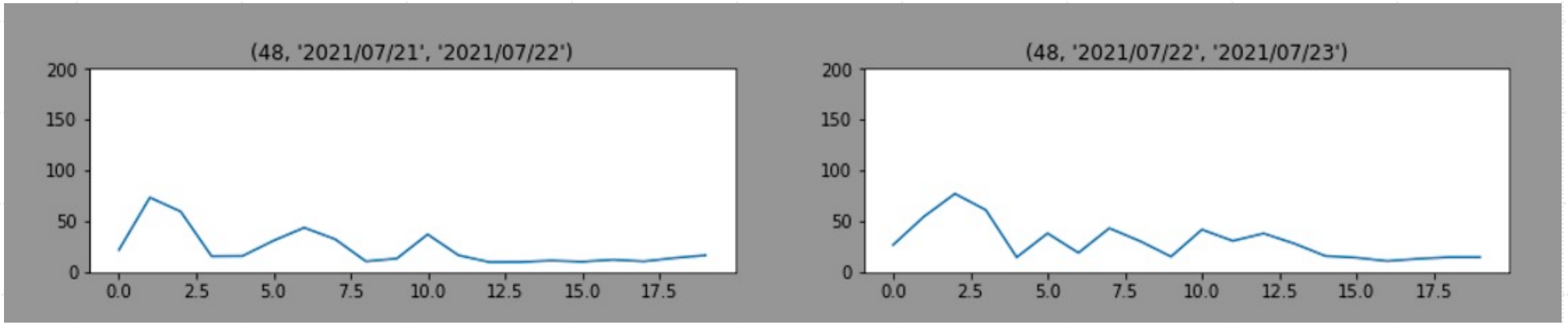
	(19, '2021/04/30', '20	(19, '2021/06/25', '20	(23, '2020/12/02', '20	(23, '2021/01/08', '20	(23, '2021/01/11', '20	(23, '2021/01/16', '20
0	33.75	68.00	15.17	11.54	12.86	13.56
1	29.67	29.00	17.52	11.52	29.53	11.16
2	33.80	39.20	14.15	9.07	18.47	10.83
3	18.56	19.50	6.36	6.48	8.50	7.88
4	11.36	17.72	6.90	7.65	5.05	6.49
5	17.17	19.67	5.70	4.72	11.95	6.15
6	15.50	60.75	20.22	4.28	13.33	4.87
7	41.08	38.29	20.38	4.97	16.94	5.38
8	6.96	34.67	4.33	10.23	14.93	5.00
9	6.02	26.00	4.53	2.87	18.06	4.50
10	5.65	7.56	4.63	4.75	15.25	3.75
11	9.45	8.72	3.20	5.00	19.90	2.58
12	5.28	11.14	3.25	14.89	11.93	5.24
13	4.88	9.16	4.00	45.33	9.55	35.14
14	5.22	8.63	4.20	18.84	6.97	16.25
15	3.83	12.31	5.20	28.13	7.23	8.06
16	1.23	6.56	6.00	11.14	5.20	27.15
17	1.43	2.28	20.27	33.20	15.17	12.62
18	1.57	2.48	37.00	20.00	8.98	18.00
19	1.42	5.62	34.17	11.75	3.27	12.00

Classification: “bad” segments

122 out of 2472 segments classified as “bad”

→ 5%

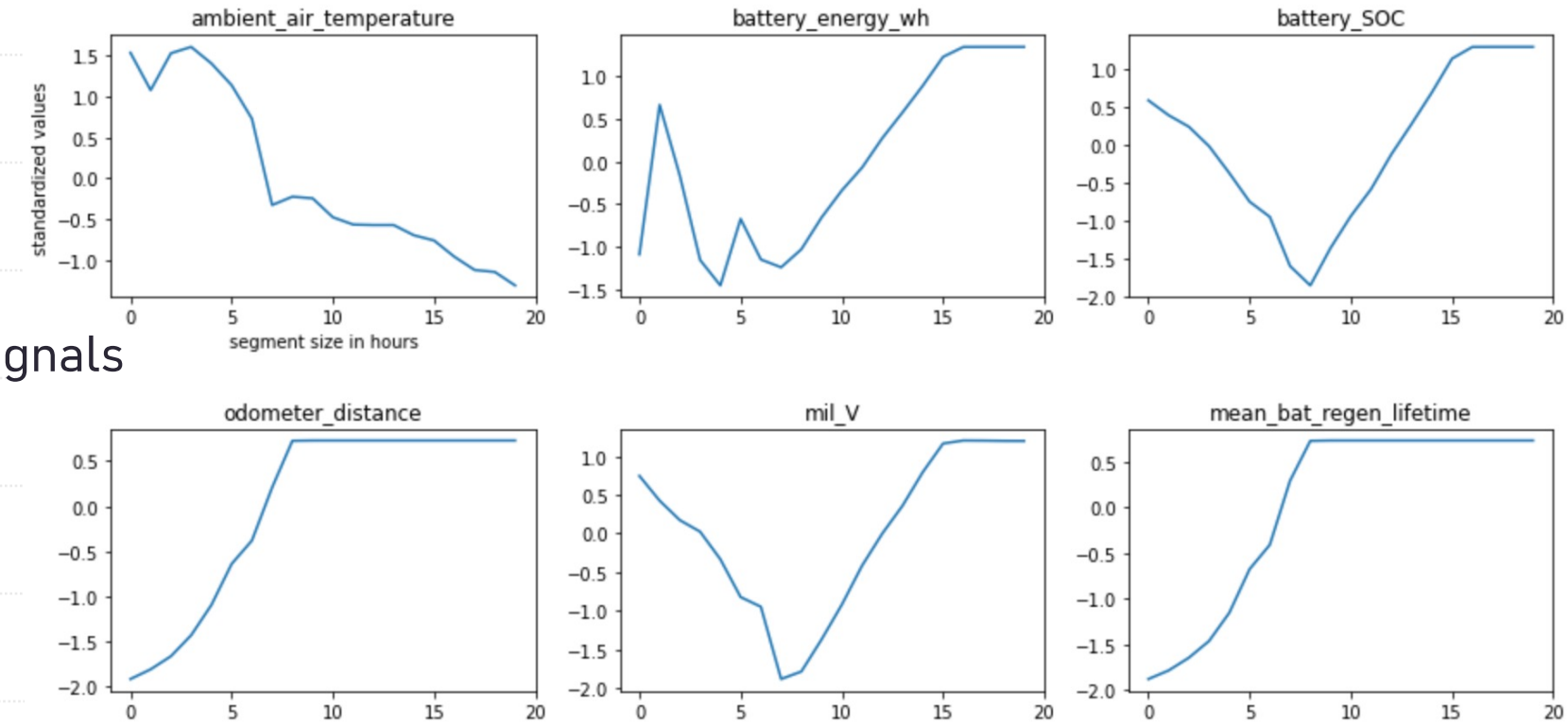




Input Data for Signal Processing

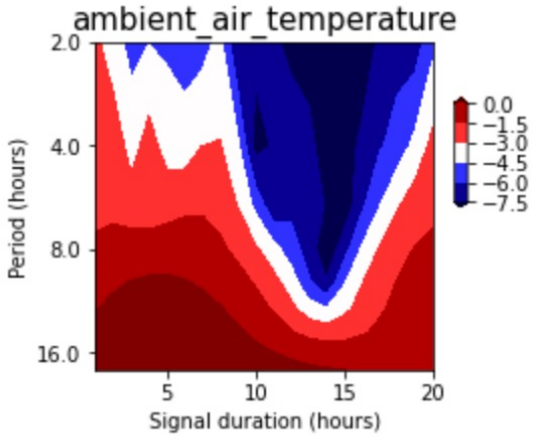
- Standardize
- Oversample
- Split into train/test

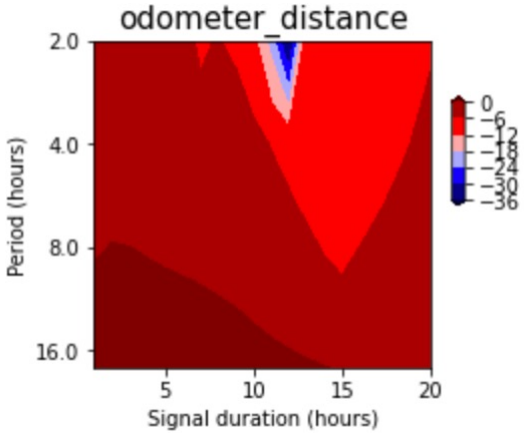
	dimensions	# ”bad” signals
Train	3310 x 20 x 6	1668
Test	1390 x 20 x 6	682

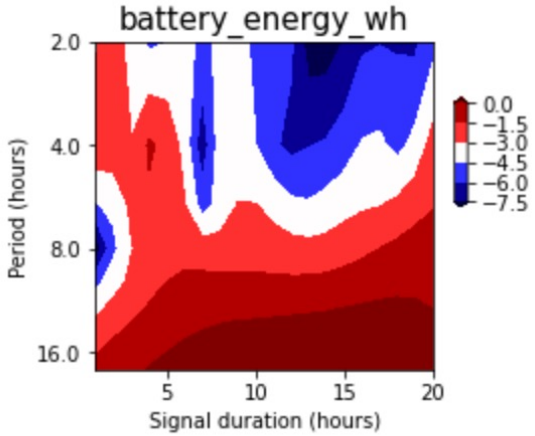


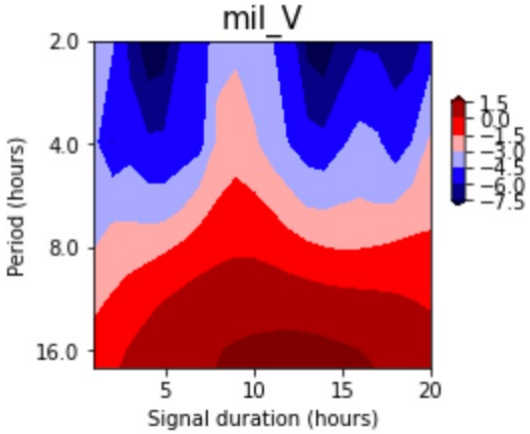
Input images for CNN

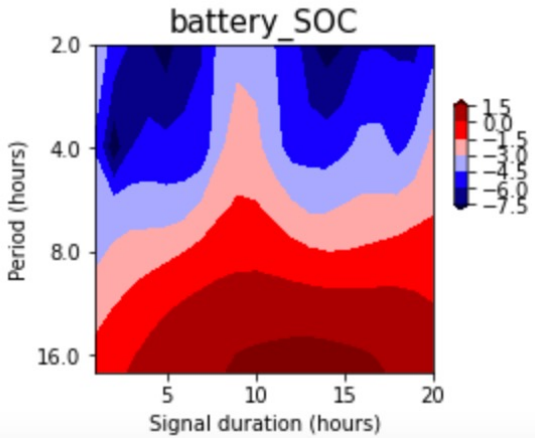
- Scaleograms
  = output of signal processing with Continuous Wavelet Transform (CWT)

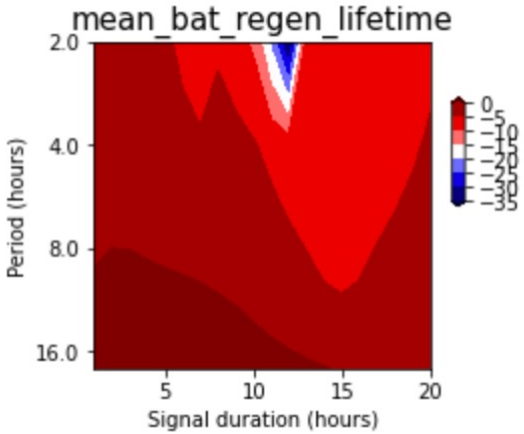






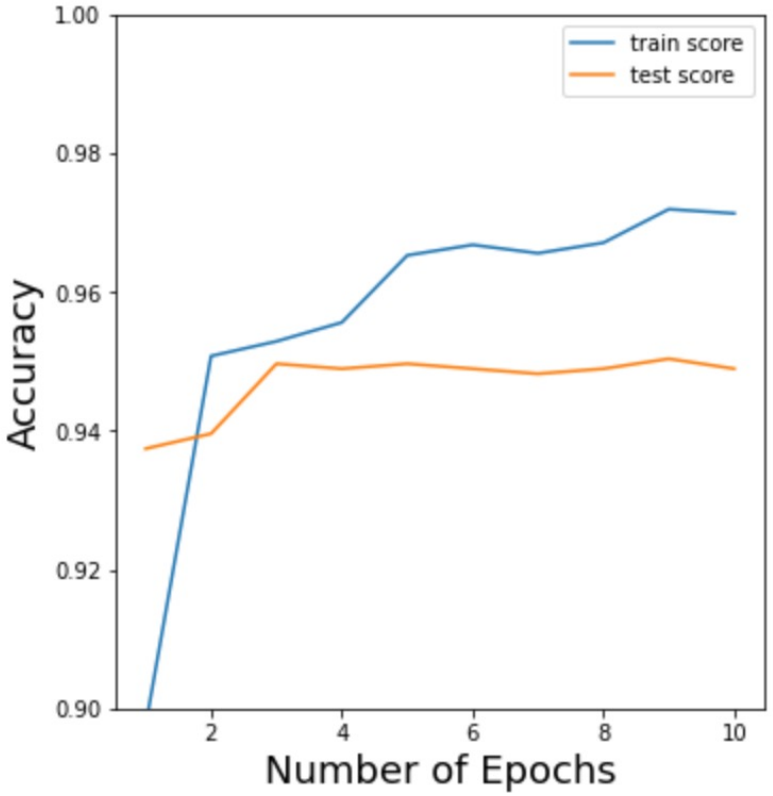






Convolution Neural Network

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 15, 15, 32)        4832

 max_pooling2d (MaxPooling2D  (None, 7, 7, 32)         0
 )

 conv2d_1 (Conv2D)           (None, 3, 3, 64)          51264

 max_pooling2d_1 (MaxPooling  (None, 1, 1, 64)         0
 2D)

 flatten (Flatten)           (None, 64)                0

 dense (Dense)               (None, 1000)              65000

 dense_1 (Dense)             (None, 2)                 2002

=================================================================
Total params: 123,098
Trainable params: 123,098
Non-trainable params: 0
```



Thank you!