Experience with Machine Learning in Accelerator Controls

K.A. Brown, S. Binello, T. D’Ottavio,
P.S. Dyer, S. Nemesure, D.J. Thomas
Collider Accelerator Department (C-AD), BNL

BROOKHAVEN NATIONAL LABORATORY
ICALEPCS2017
Barcelona · Spain, October 8-13 · Palau de Congressos de Catalunya

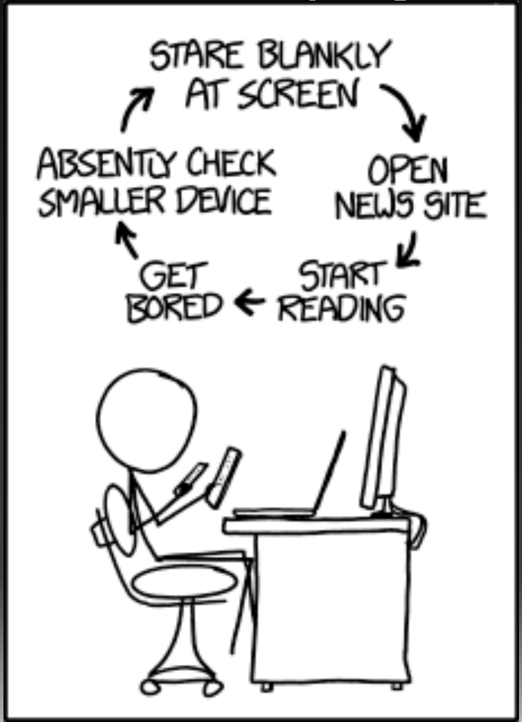
Outline

- Why add this complexity to the controls?
- What others are doing or have done.
- Some definitions
- TensorFlow™ Deep Learning approach
- NuPIC neurocomputing approach
- Future plans

https://xkcd.com/1838/

Why?

- Alarms can come too late
- Already have real-time data processing on comfort displays in MCR & on the web.



Computers don't:
Stare blankly
Absently check phone messages
Get bored
Browse the web

https://xkcd.com/1411/

Why? Possible Applications

Power Supply starting to regulate poorly.

Beam Loss slowly increasing or change in pattern.

Beam orbit change or change in pattern.

Vacuum behavior change.

Orbit Correction (AI system)

In general: use ML to detect changes, then can make correlations

Can we can use machine learning to recognize (or train, if you like) in software what a person is able to detect visually?

Machine Learning literature tells us, yes . . .

But, it may not be easy.

https://xkcd.com/1425/

Related Work

- Arthur Samual, IBM, 1959, where the idea was to give *"computers the ability to learn without being explicitly programmed"*

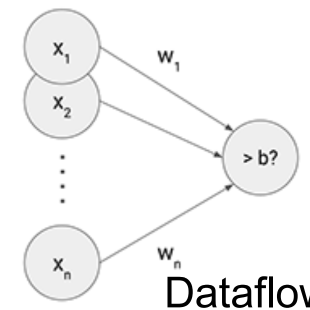
- 1987, T. Higo, H. Shoaee, and J. E. Spencer, SLAC, applications to AI
- 1989, J. E. Spencer, SLAC, using Neural Network techniques in accelerator controls
- 1991, D. Nguyen, M. Lee, R. Sass, and H. Shoaee, SLAC, used Neural Network techniques for beam line control
- 1994 E. Bozoki and A. Friedman, BNL, neural networks for orbit control in the NSLS I
- 2012, E. Meier, Australian Synchrotron, neural networks for orbit correction
- Iterative learning for LLRF, etc. at various facilities
- 2015, A. L. Edelen, et al., have been experimenting with the use of machine learning techniques for RF gun temperature control

Definitions

- Learning = improving with experience at some task
- Machine Learning (ML) focuses on algorithms that can ‘learn’ (as in above def.) and make predictions on data
- Statistical machine learning uses automated techniques for predictive modeling
- In ML the focus is on developing efficient algorithms to optimize a predictive model
- Anomaly detection = learn what is normal and flag deviations as anomalies
- Outlier identification = statistically significant deviations from some mean

TensorFlow™ Deep Learning approach

- Open-source library developed by Google
- A framework for creating deep learning models
  - Deep learning models are basically multi-layered neural networks.
- In a TensorFlow™ model
  - A neural network model is constructed
    - Autoencoder : feedforward model (in ~ out)
    - Long Short-term memory: recurrent model (in ~ out)
  - This is compiled into a dataflow graph
    (separates definition of computations from their execution)



$$\sum_{i}^{n} w_i x_i > b$$

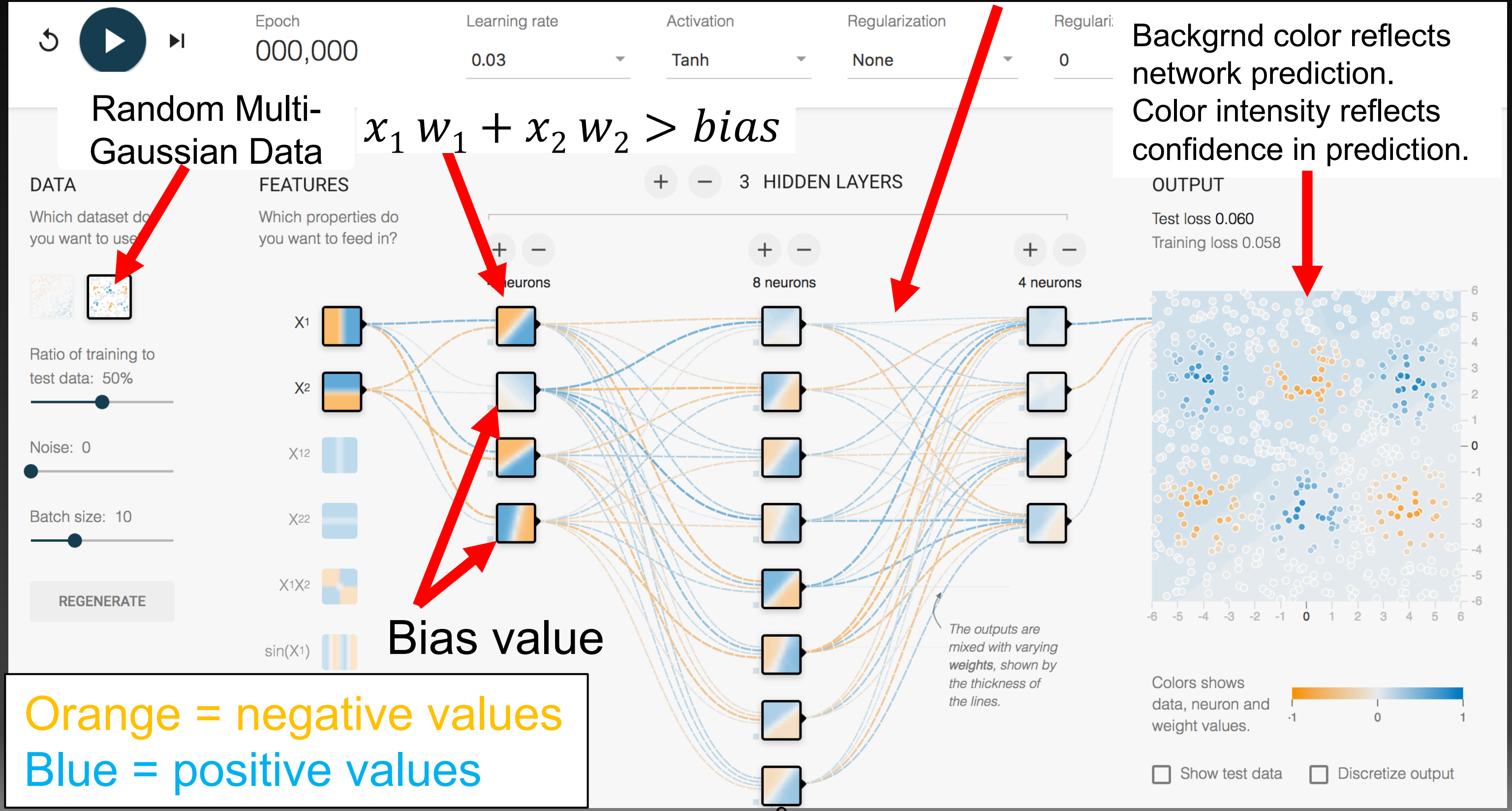
Dataflow graph example

3 layer neural network: 4 - 8 - 4 neurons

Input values colored from -1 to 1

$x_1, x_2 \in \sim(-6, 6)$

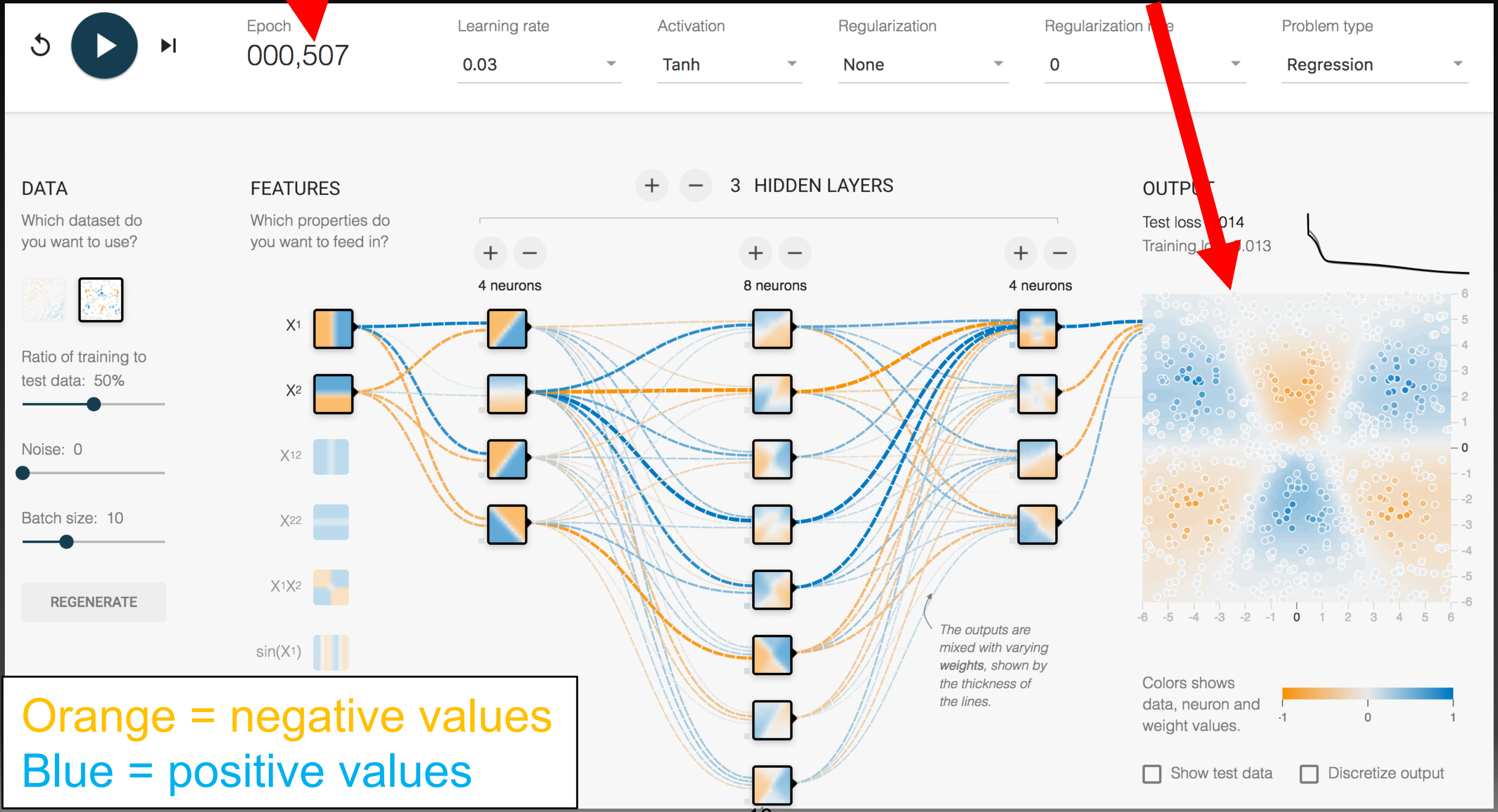
Boldness of lines reflects the weight



http://playground.tensorflow.org

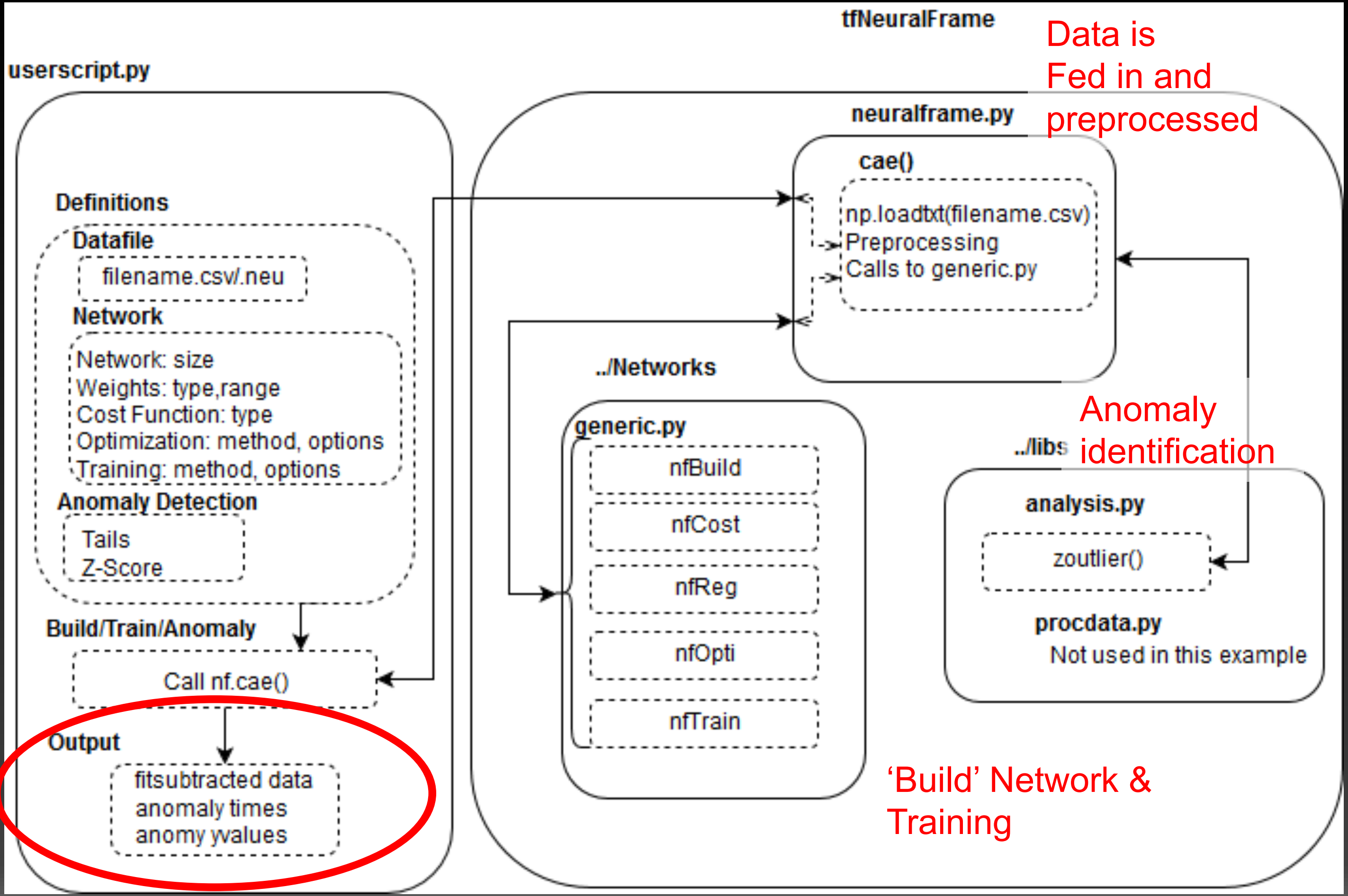
After >500 ‘iterations’.

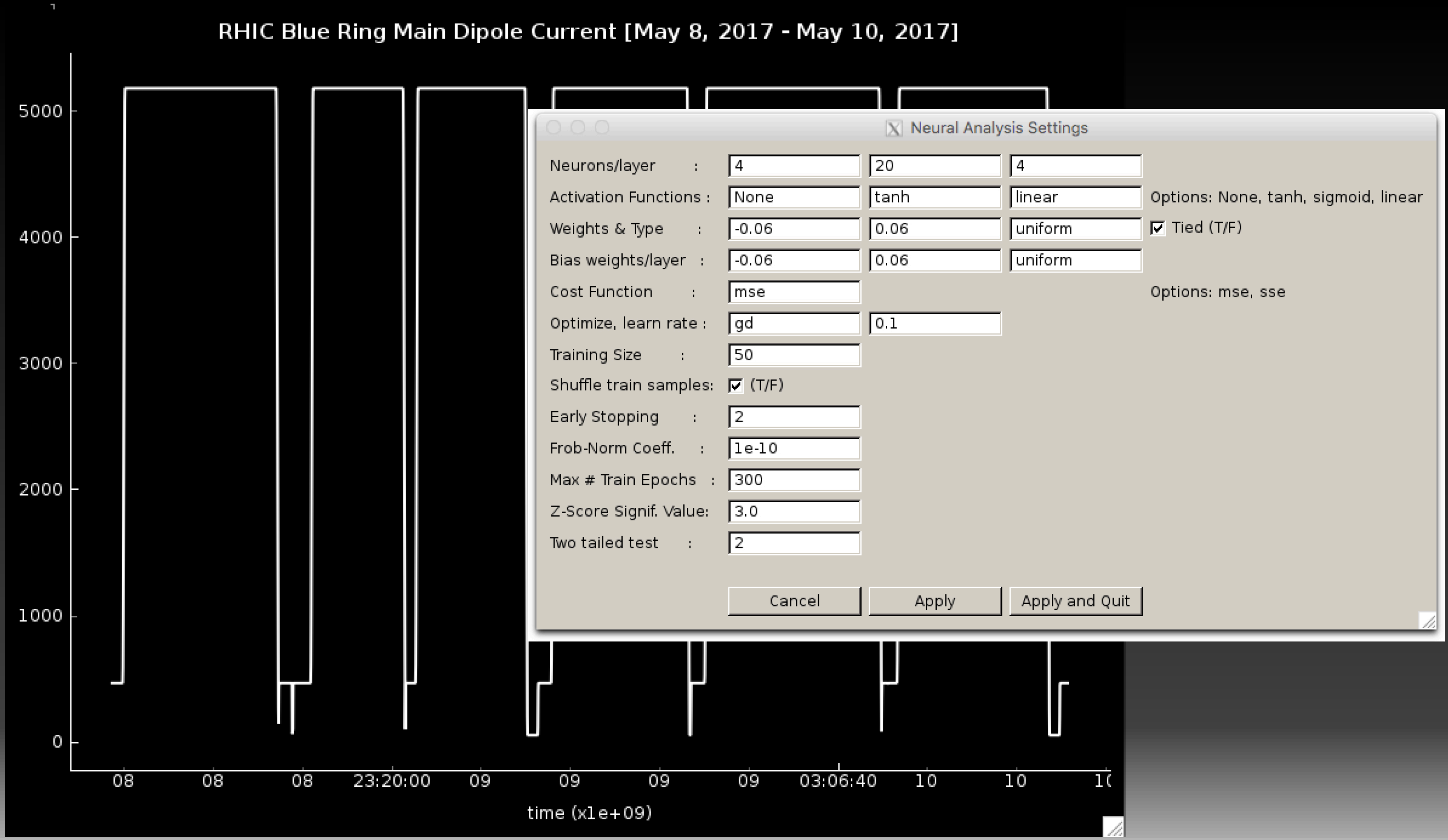
Learned pattern now matches original pattern.



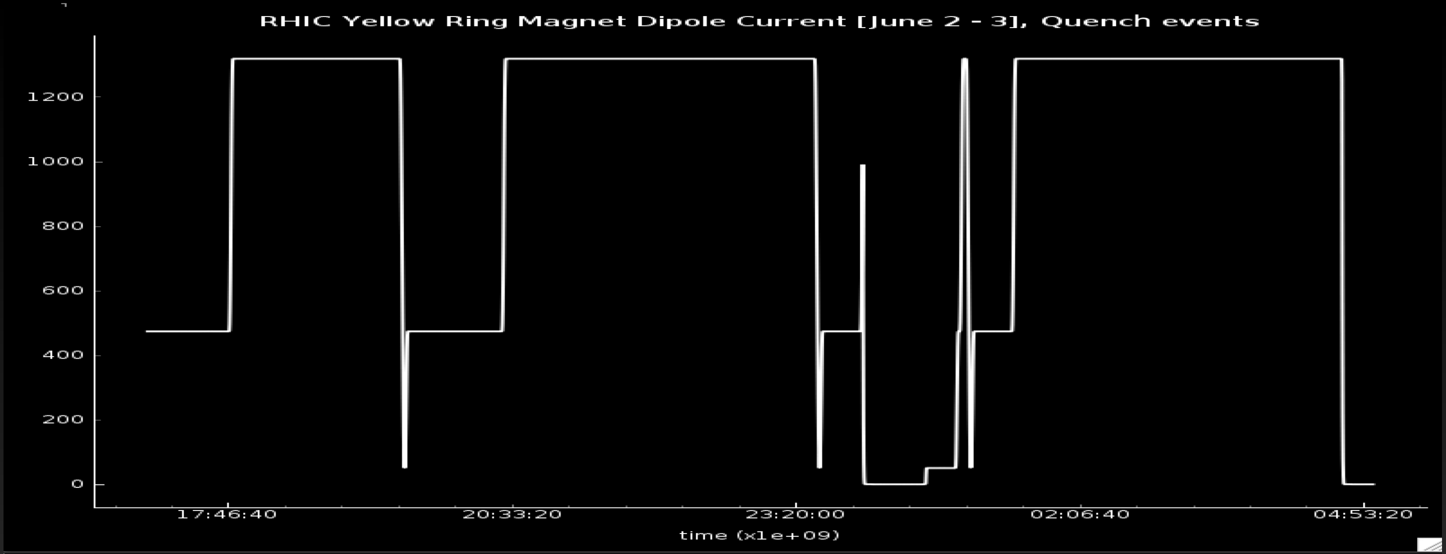
Orange = negative values
Blue = positive values

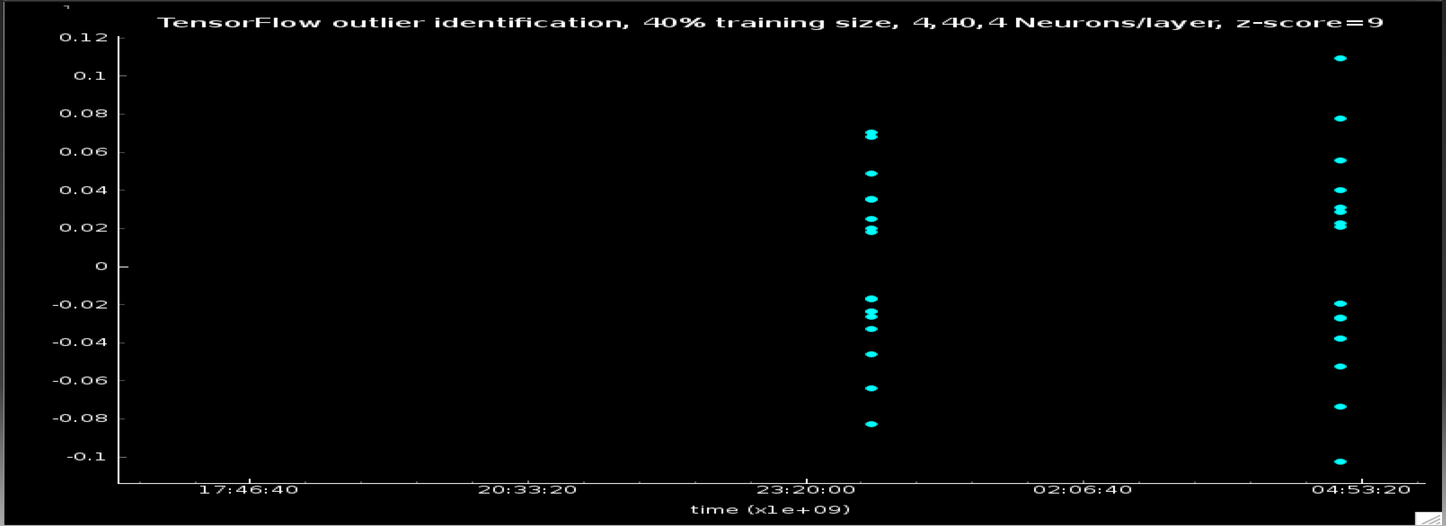
http://playground.tensorflow.org

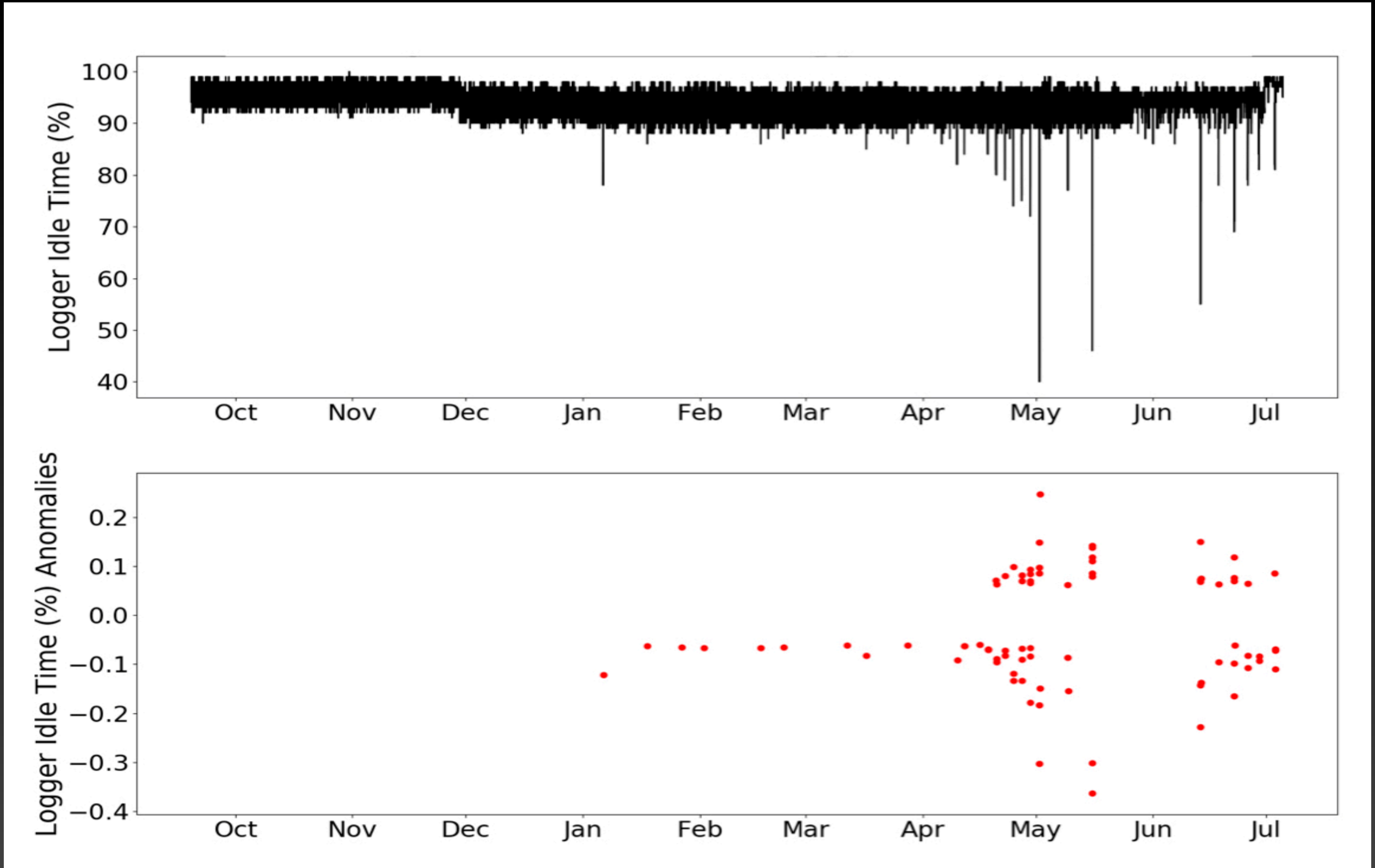

tfNeuralFrame
Data is
Fed in and
preprocessed
userscript.py
neuralframe.py
cae()
Definitions
Datafile
filename.csv/.neu
np.loadtxt(filename.csv)
Preprocessing
Calls to generic.py
Network
Network: size
Weights: type,range
Cost Function: type
Optimization: method, options
Training: method, options
../Networks
generic.py
nfBuild
nfCost
nfReg
nfOpti
nfTrain
../libs
Anomaly
identification
analysis.py
zoutlier()
procdata.py
Not used in this example
Anomaly Detection
Tails
Z-Score
Build/Train/Anomaly
Call nf.cae()
Output
fitsubtracted data
anomaly times
anomy yvalues
'Build' Network &
Training



RHIC Blue Ring Main Dipole Current [May 8, 2017 - May 10, 2017]
5000
4000
3000
2000
1000
0
08
08
08
23:20:00
09
09
09
09
03:06:40
10
10
time (x1e+09)
Neural Analysis Settings
Neurons/layer :
4
20
4
Activation Functions :
None
tanh
linear
Options: None, tanh, sigmoid, linear
Weights & Type :
-0.06
0.06
uniform
Tied (T/F)
Bias weights/layer :
-0.06
0.06
uniform
Cost Function :
mse
Options: mse, sse
Optimize, learn rate :
gd
0.1
Training Size :
50
Shuffle train samples:
(T/F)
Early Stopping :
2
Frob-Norm Coeff. :
1e-10
Max # Train Epochs :
300
Z-Score Signif. Value:
3.0
Two tailed test :
2
Cancel
Apply
Apply and Quit


RHIC Yellow Ring Magnet Dipole Current [June 2 - 3], Quench events
1200
1000
800
600
400
200
0
17:46:40
20:33:20
23:20:00
02:06:40
04:53:20
time (x1e+09)


TensorFlow outlier identification, 40% training size, 4,40,4 Neurons/layer, z-score=9
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
17:46:40
20:33:20
23:20:00
02:06:40
04:53:20
time (x1e+09)


Logger Idle Time (%)
100
90
80
70
60
50
40
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Logger Idle Time (%) Anomalies
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul

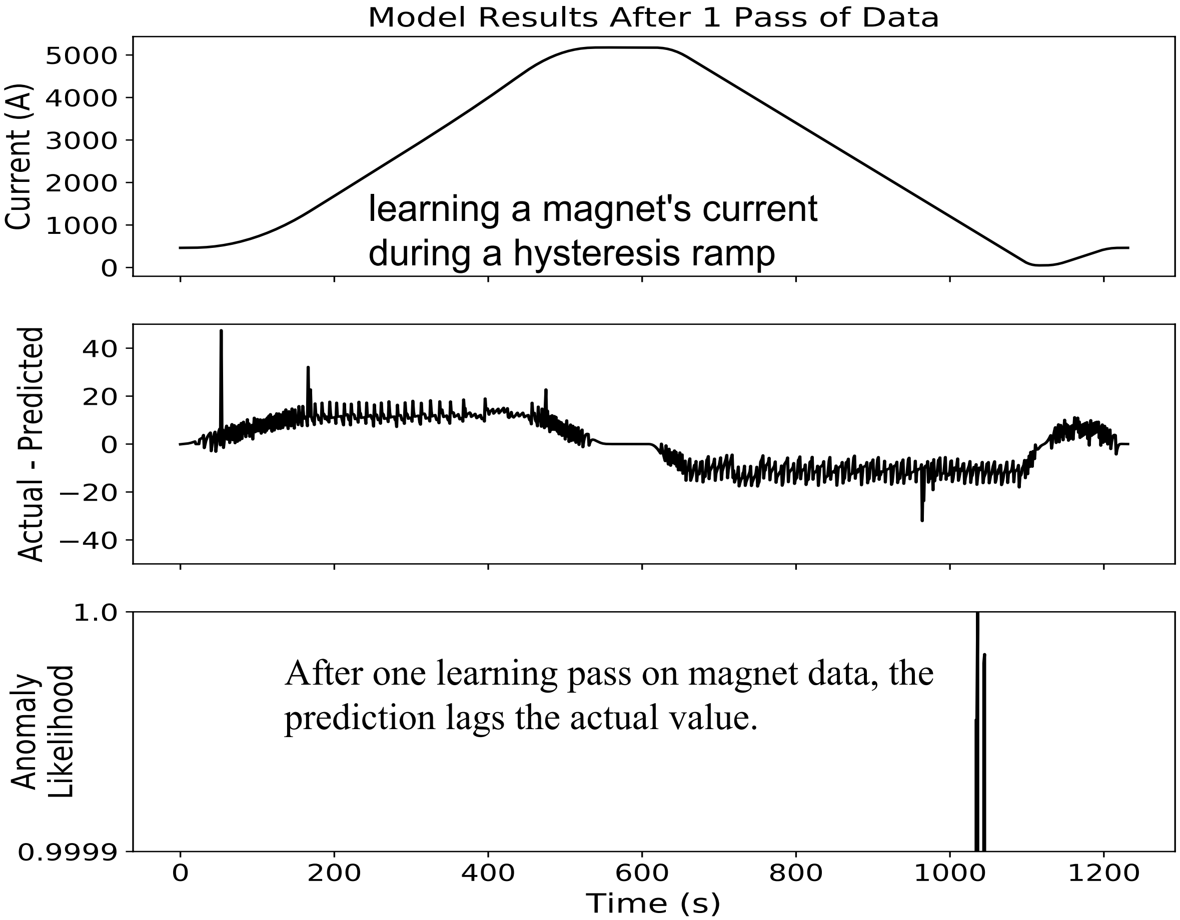
NuPIC Neurocomputing approach

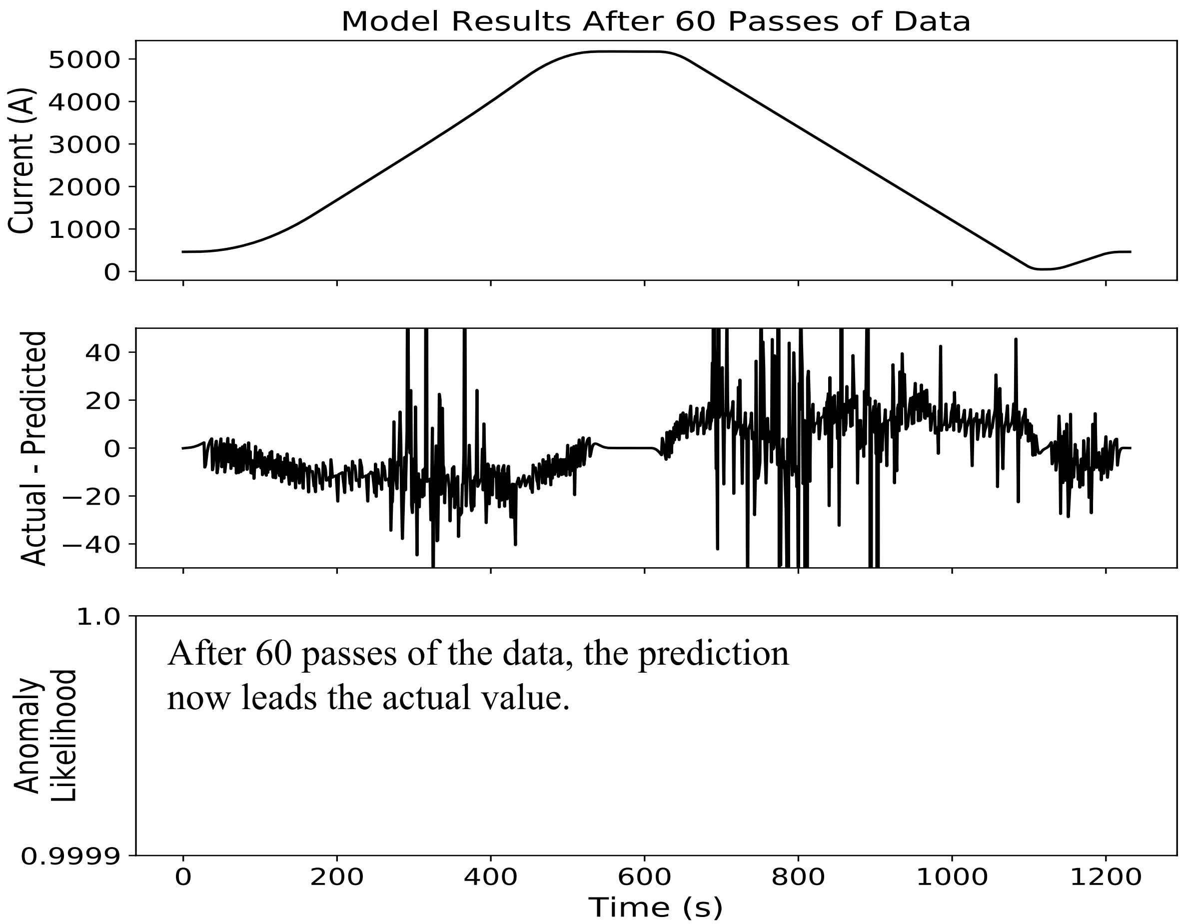
- "On Intelligence", by Jeff Hawkins and Sandra Blakeslee
- models the way the human cortex learns
- is a machine intelligence framework, based on models of how animal brains function, focusing on prediction
- Library is an open source project with various implementations (C++, Java, Python, Clojure)

NuPIC

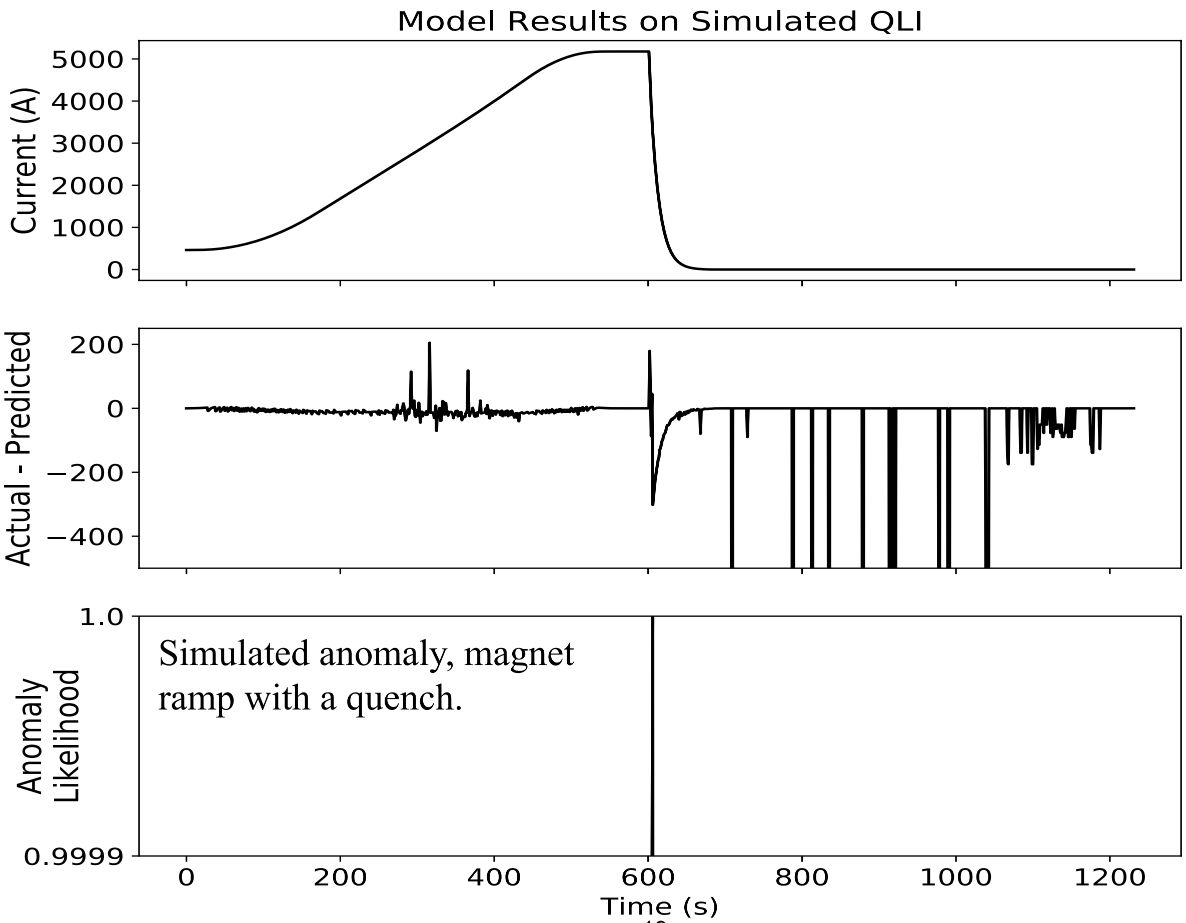
(Numenta Platform for Intelligent Computing)

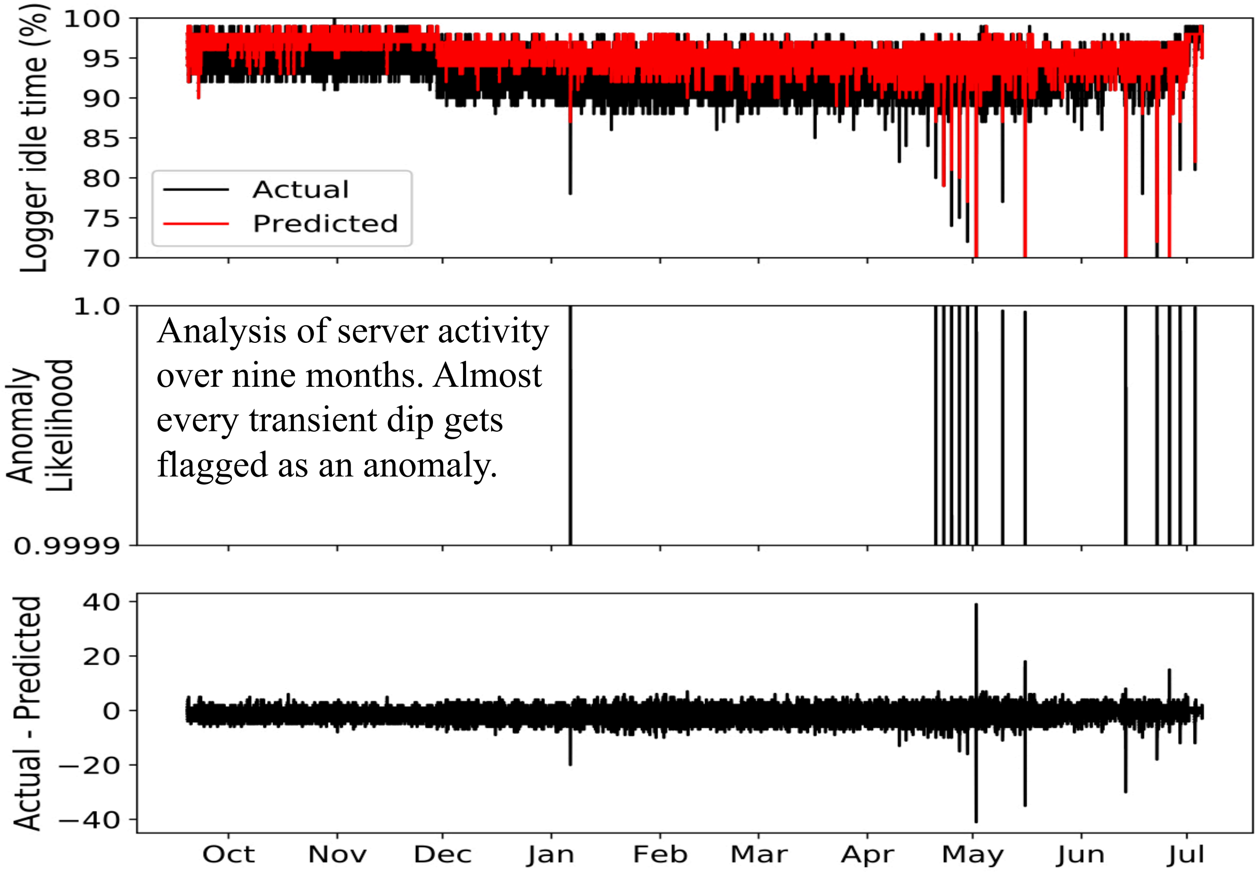
- Sparse Distributed Representation (SDR)
  - A ’natural’ data type for the human cortex
  - Can be visualized as sparse arrays filled with 1’s and 0’s, where <20% are 1’s
  - Input data is encoded into an SDR
- Spatial Pooler (SP)
  - A stack of SDRs
  - Columns of cells connect to a subsample of cells in the input
  - Connections are called the synapses, are modified during learning to pool spatial patterns in the input
- Temporal Learning (TL) / Temporal Memory (TM)
  - Learns sequences of the active columns in the SP
  - SP learns to identify patterns, TM learns the sequential contexts of those patterns
- Anomaly detection compares the predicted cells against the next set of active cells and produces an anomaly score
- Swarms – NuPIC process to determine initial parameters


Model Results After 1 Pass of Data
Current (A)
5000
4000
3000
2000
1000
0
learning a magnet's current
during a hysteresis ramp
Actual - Predicted
40
20
0
−20
−40
Anomaly
Likelihood
1.0
0.9999
After one learning pass on magnet data, the
prediction lags the actual value.
0
200
400
600
800
1000
1200
Time (s)



Model Results After 60 Passes of Data
Current (A)
5000
4000
3000
2000
1000
0
Actual - Predicted
40
20
0
−20
−40
Anomaly Likelihood
1.0
0.9999
After 60 passes of the data, the prediction now leads the actual value.
0
200
400
600
800
1000
1200
Time (s)

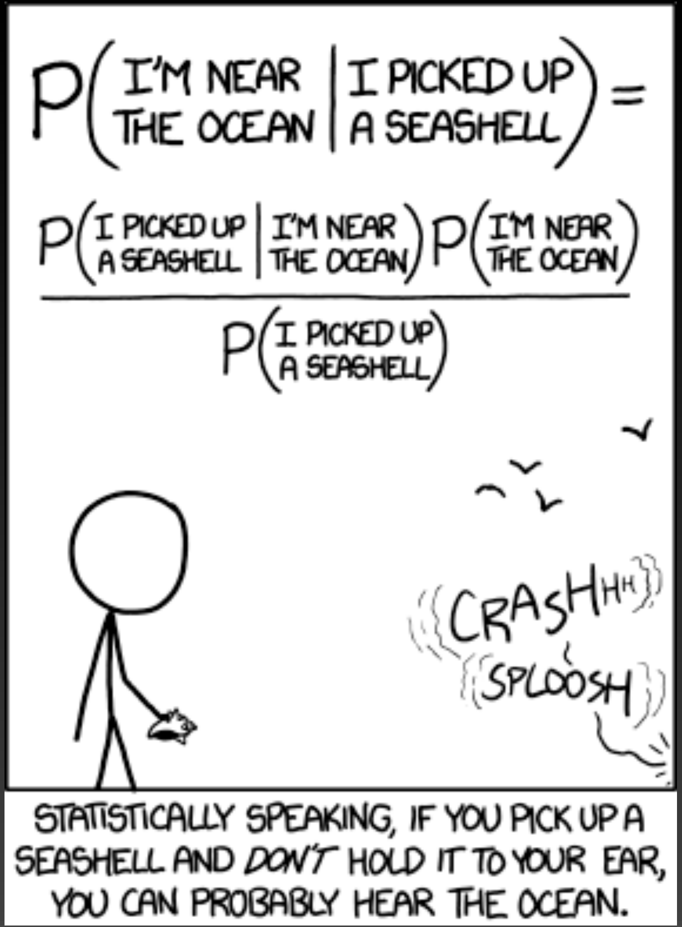

Model Results on Simulated QLI
Current (A)
5000
4000
3000
2000
1000
0
Actual - Predicted
200
0
−200
−400
Anomaly Likelihood
1.0
0.9999
Simulated anomaly, magnet ramp with a quench.
0
200
400
600
800
1000
1200
Time (s)



Analysis of server activity over nine months. Almost every transient dip gets flagged as an anomaly.

Future Work

- Continue to study our ability to find anomalies in different kinds of data (e.g., orbit data, power supplies, server stats, network stats, etc.)
- Look at combining anomaly detection with data correlation
  - E.g., how well do anomalies in different signals correlate?
- Eventually formulate various tools
  - For operations
    - Early warning? Is such a tool possible? Useful?
    - Improved data mining? More quickly find interesting data in a large repository.
  - For control systems
    - Load and ’health’ of systems
    - Changes in network load and behavior



https://xkcd.com/1236/