AI/ML Opportunities in the IIoT

Ramon Perez

Will Goodrum

ELDER RESEARCH
DATA SCIENCE • MACHINE LEARNING • AI

Before we get started

Questions

Please use the questions tab to enter questions throughout the webinar

Questions will be answered during the Q&A session at the end

Slides

A pdf of the slides will be posted on the webinar website next week

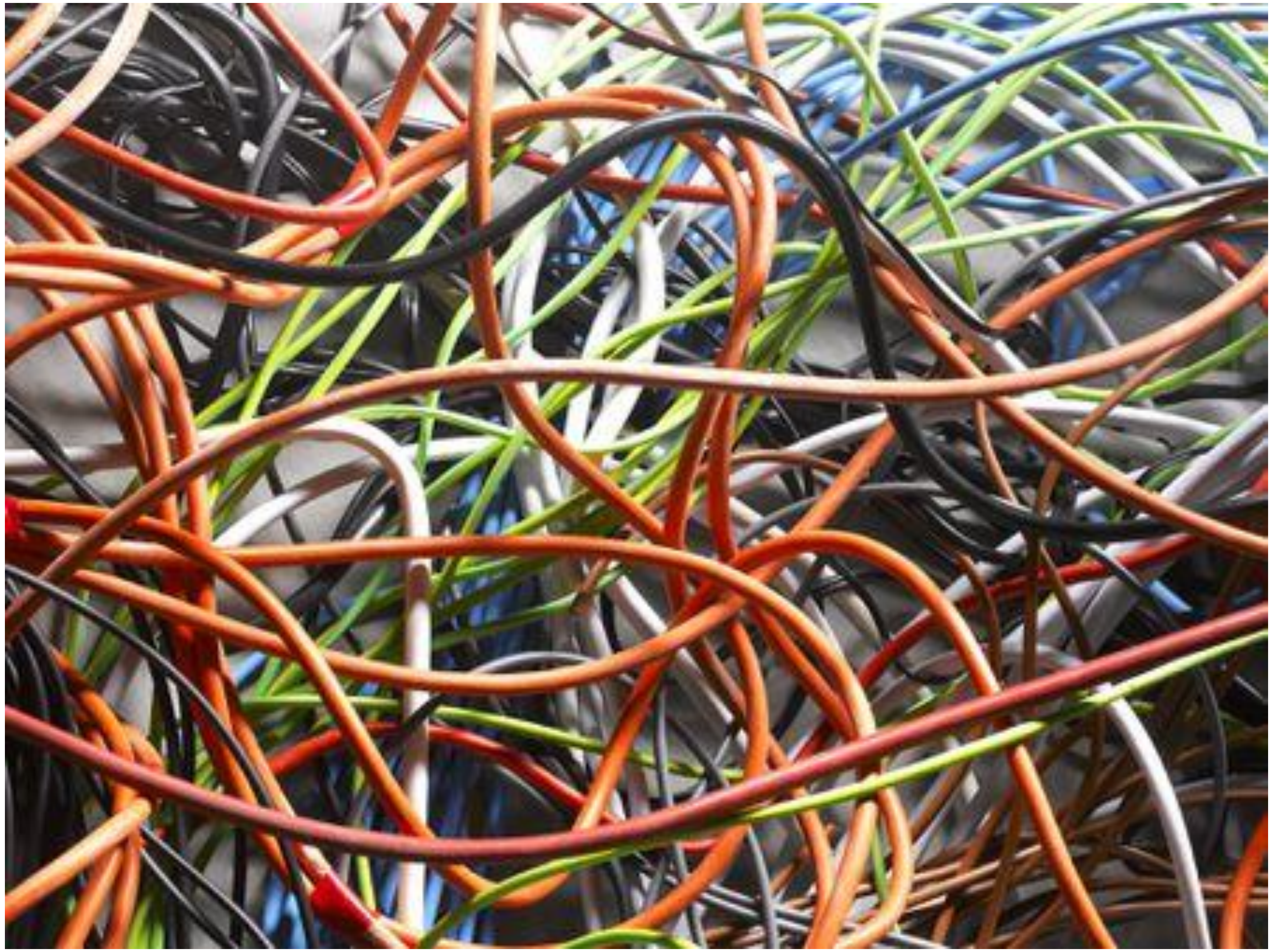
or email paul.derstine@elderresearch.com to request a copy

Problem during the webinar?

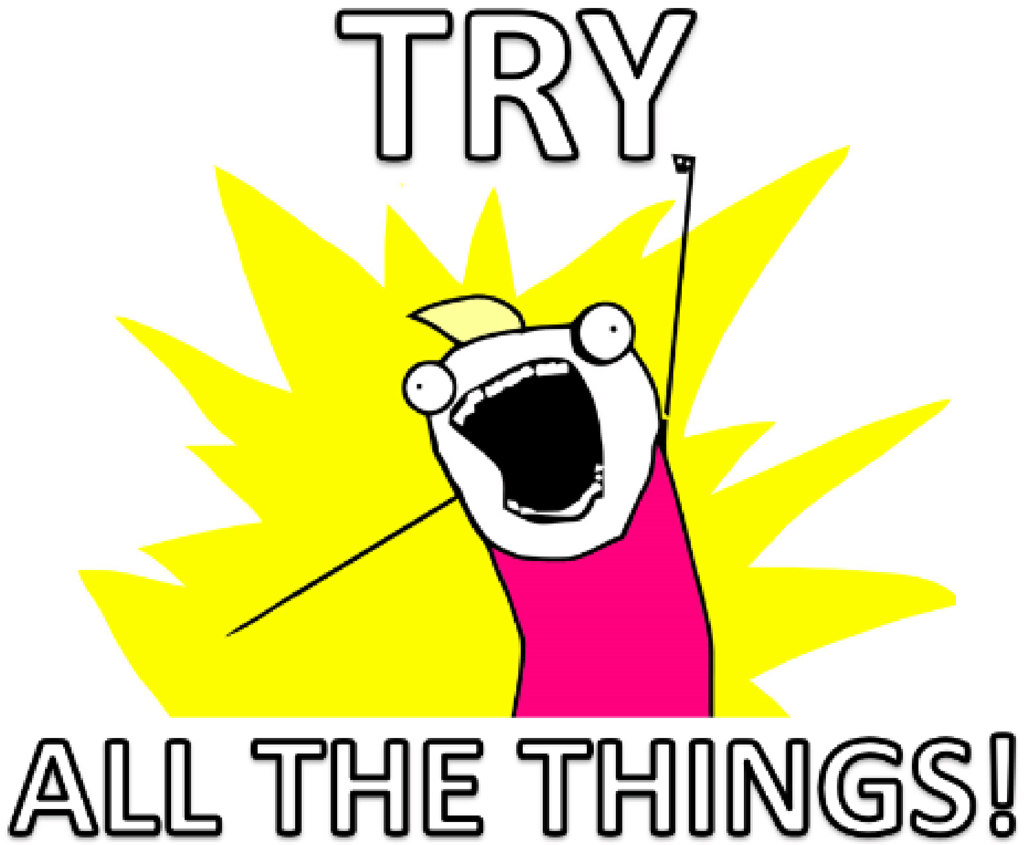
Simply refresh your browser screen

Is the IIoT. . . A black box?

Is the IIoT. . . A mess of wires?



Is IIoT Strategy. . . Trying everything?



Try ALL THE THINGS!

Is IIoT strategy. . . A magical unicorn?

Data

AI/ML

Value

Now that we're done with the memes/preconceptions

IIoT AI/ML challenges

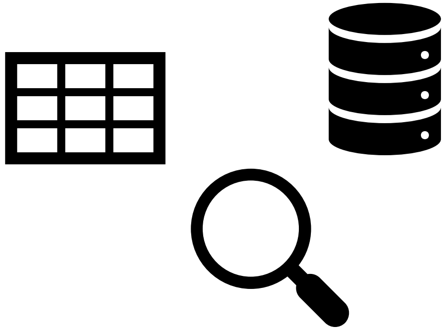
Controversial Statement #1

IIoT AI/ML challenges are the same as in other fields

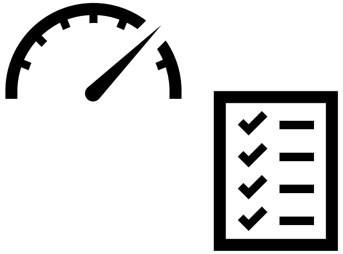
General AI/ML Challenges (not unique!)

Clearly-Defined Business Need

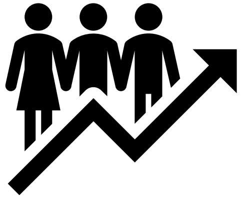
Data Availability



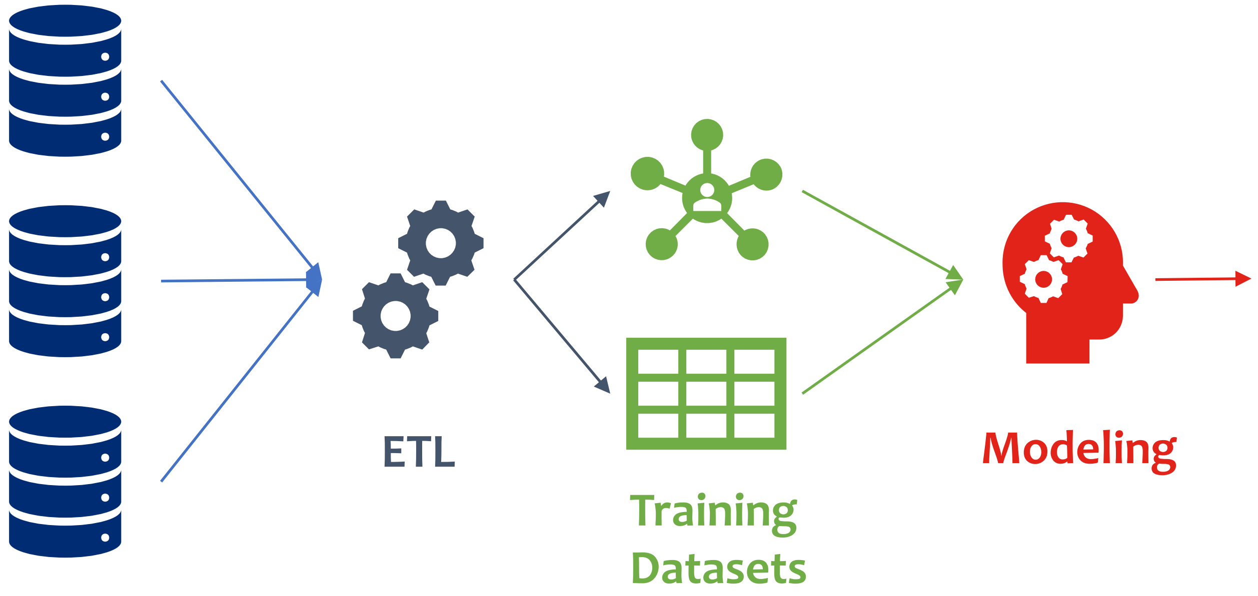
Data Quality



Actionable Outcomes



Example: Actionable Outcome

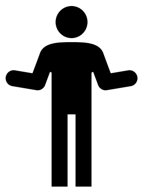


Recommendation:

Possibly anomalous energy usage by pump on chill water circuit. Reduce demand.

Operations Says:

But it's a critical care facility. We won't reduce demand (unless we *have* to)



The major IIoT challenge

Controversial Statement #2

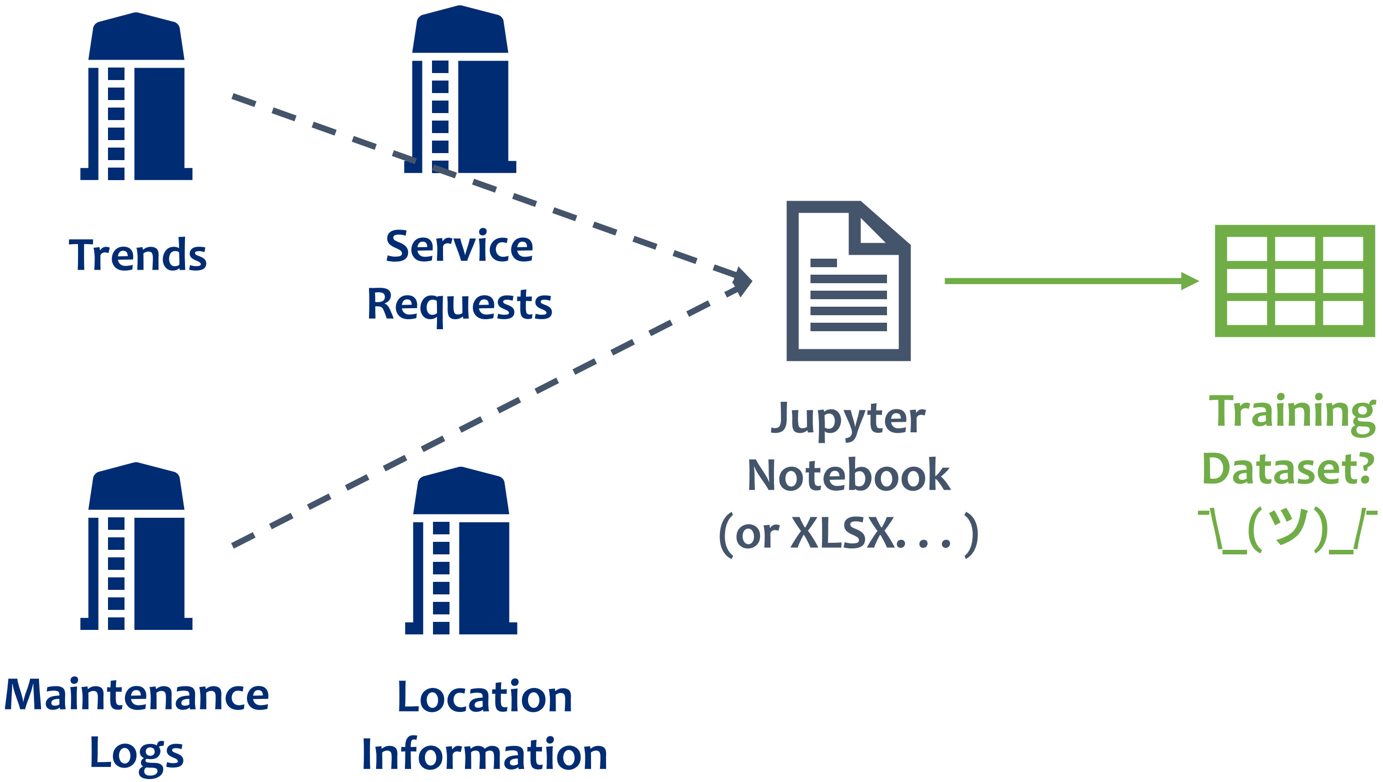
The ***major*** IIoT challenge is Data Engineering

Example: Predictive Maintenance

What does an AI need to learn to predict that maintenance is required?

Example: Predictive Maintenance

What does reality tend to look like when trying to predict maintenance?



IIoT AI/ML opportunities: old problems, new techniques

Controversial Statement #3

IIoT benefits come through applying new techniques to old problems

"I think we're in good enough shape to start making the same mistakes again."

Old Problems, New Techniques

Old Problems		New Techniques
• Predictive Maintenance	→	Neural Network Classification
• Root Cause Analysis	→	Graph Analytics/Clustering
• Maintenance Postmortem	→	Natural Language Processing
• Estimating Maintenance Costs	→	Time Series Forecasting

And now: a detailed case study

Disease Event Detection using wearable medical devices

Ramon Perez

21 August 2019

Executive Summary

Background

- A health tech startup designed a sensor that is implanted in the body (intentionally vague to protect NDA)
- Intended to discover early indicators of a particular disease event among major organs
- Produces a time series signal of the area over a 60 second trace
- Currently testing in laboratory on sheep subjects
- Intent to move on to human trials and FDA approval
- Product would be used to monitor patients with previous cases of the disease
- Elder Research engaged to provide machine learning models using sensor data for disease event detection

Project Goals

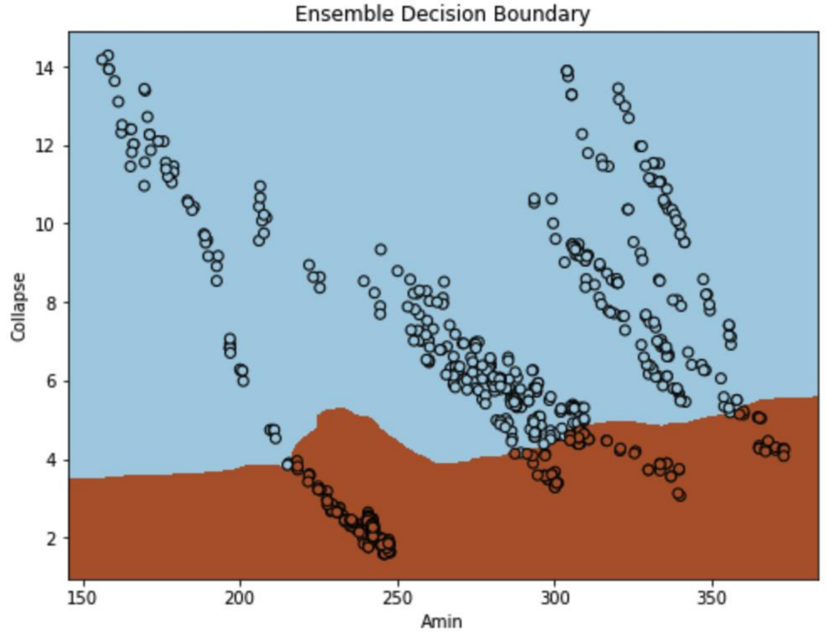
- Original Goal: Classify traces as either High, Moderate, or Normal risk
- Modified Goal: Classify traces as either High or Normal
- Given: 700+ traces from four sheep collected 30 days apart
- Labels: Three sheep had traces labeled High-Moderate-Normal simulated in the lab. One sheep had only Moderate-Normal traces

Key Decisions

- Deliver a binary classifier on normal/high
- Relabel the moderates as normal/high using label spreading
- Deliver 2 models
  - Ensemble model as Python command line deployment & as a Docker container w/ RESTful API
  - Convolutional Neural Network w/ Transfer Learning as a Docker container w/ RESTful API
- Define input data type
  - Ensemble consumes 2 parameters
  - CNN consumes entire 60-second trace as JSON array
- Define output:
  - Predicted class: Normal or High
  - Class probabilities

Results

- Delivered 2 models: Ensemble and CNN w/ Transfer Learning
- Client will use Ensemble and set aside CNN for now
- Ensemble provides high accuracy with ease of use
- CNN provides patient calibration, but more complex
- Final Ensemble architecture:
  - Logit & KNN with weighted voting
  - Leave-One-Sheep-Out model validation
  - Synthetic labeling
- Final Ensemble performance:
  - Very high mean holdout accuracy of 96.93%
  - Mean sensitivity of 93.84%, Mean specificity of 98.94%
  - Specific sheep accuracy between 93.9% and 99.4%



Disclaimer

- We have only built with four sheep
- We have not seen dry vessels in this data
- We currently have no human data
- There is no guarantee that a newly observed sheep will have similar characteristics to prior sheep
- There is no guarantee that humans will be similar to sheep
- CNN offers robustness at the expense of accuracy and ease of deployment
- Ensemble offers accuracy and ease of deployment at the expense of robustness
- Models provide probability of high vs. normal risk, but should not be considered a final diagnosis

Data Discovery

Feature Selection

- Principal Components Analysis (PCA)
  - Factor loadings on A_{min}, A_{max}, Collapse & AxisRatio
- XGBoost feature selection
  - Collapse produces strong signal in a boosted tree model
  - AxisRatio also strong here
- AxisRatio (ultrasound) must be discarded
  - This data is not produced by the sensor
  - Models built on this will fail at run-time

PCA loadings matrix

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Amin	47.26	3.50	2.26	-0.19	0.01	0.00	0.00	0.00	-0.0	0.0
Amax	52.33	-3.52	-1.96	0.18	-0.01	-0.00	-0.00	-0.00	0.0	-0.0
Collapse	0.49	-2.38	-1.48	-0.54	0.02	0.00	0.00	0.00	-0.0	0.0
RespFrequ	0.01	-0.00	-0.01	-0.01	-0.01	-0.01	-0.02	-0.00	-0.0	-0.0
RespMagn	0.03	-0.05	-0.02	0.01	0.01	0.01	-0.00	0.01	-0.0	-0.0
CardFrequ	-0.09	0.16	-0.09	-0.07	-0.20	0.02	0.00	-0.00	-0.0	-0.0
CardMagn	0.00	0.00	-0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.0	-0.0
VenousResistance	-0.00	-0.00	-0.00	-0.00	0.00	0.00	0.00	-0.00	-0.0	-0.0
CRC	0.01	0.05	-0.05	-0.02	-0.06	-0.08	0.00	0.00	-0.0	0.0
AxisRatio	1.98	10.19	-1.80	0.00	0.01	0.00	-0.00	-0.00	-0.0	-0.0

XGBoost Feature Importance

All animals - including `AxisRatio`

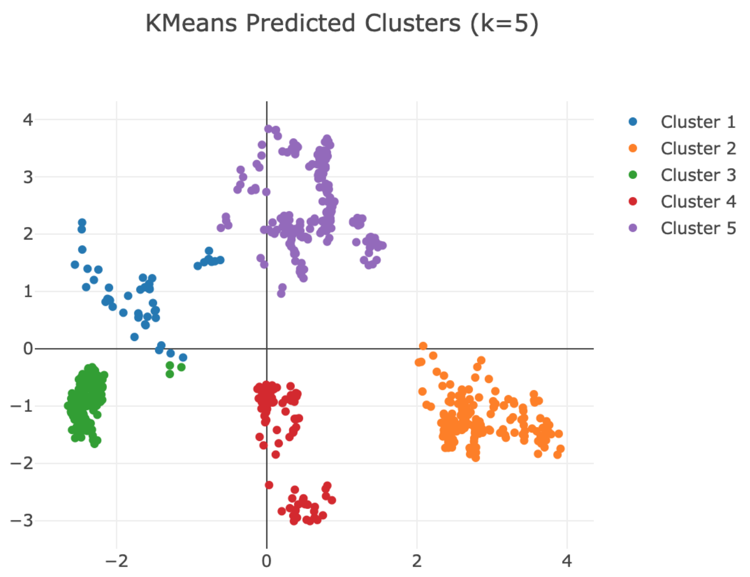
```
1  Collapse            0.451145
2  AxisRatio           0.125019
3  CardMagn            0.100377
4  RespFrequ           0.070460
5  Amin                0.061697
6  CRC                 0.048218
7  CardFrequ           0.047457
8  RespMagn            0.044913
9  VenousResistance    0.030398
10 Amax                0.020315
```

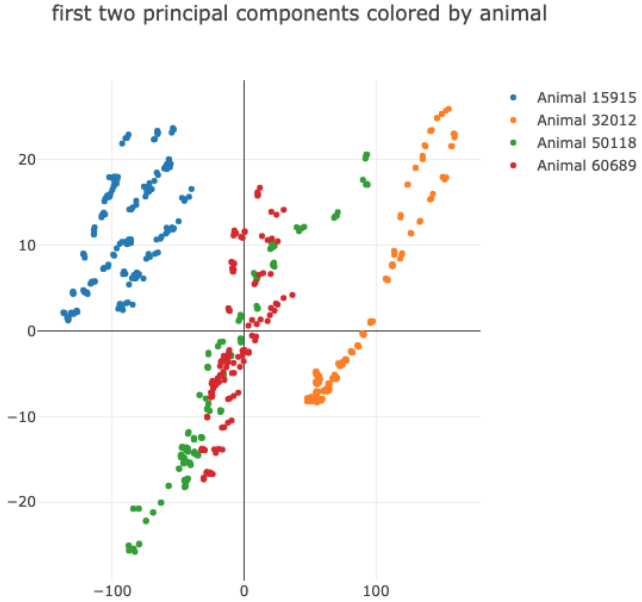
All animals - excluding `AxisRatio`

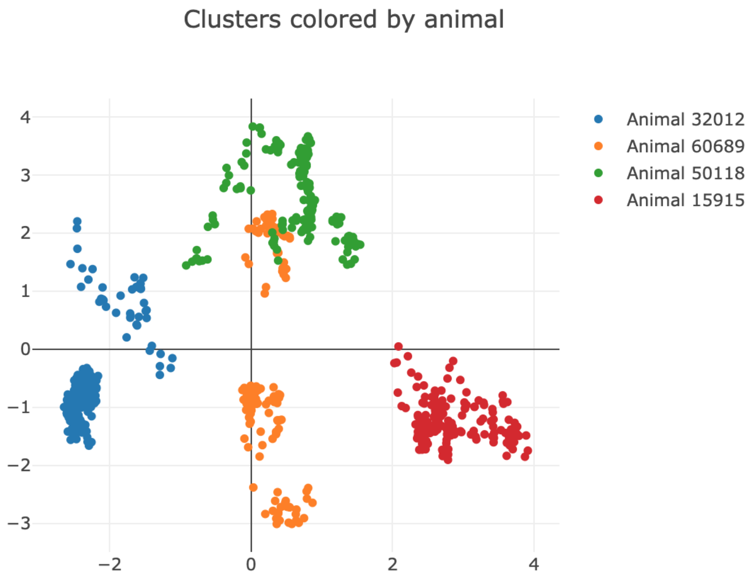
```
1  Collapse            0.477545
2  CardMagn            0.091555
3  RespFrequ           0.088890
4  CardFrequ           0.071127
5  VenousResistance    0.063737
6  Amin                0.062838
7  RespMagn            0.057246
8  CRC                 0.051179
9  Amax                0.035884
```

Clustering

- Clustering on principal components
  - Clear distinctions between animal groups
  - Animals 15915 & 32012 push to left and right sides of distribution (respectively)
  - Animals 50118 & 60689 more similar and centrally distributed (see below)
- K-means clustering
  - Four or five distinct groups (image right)



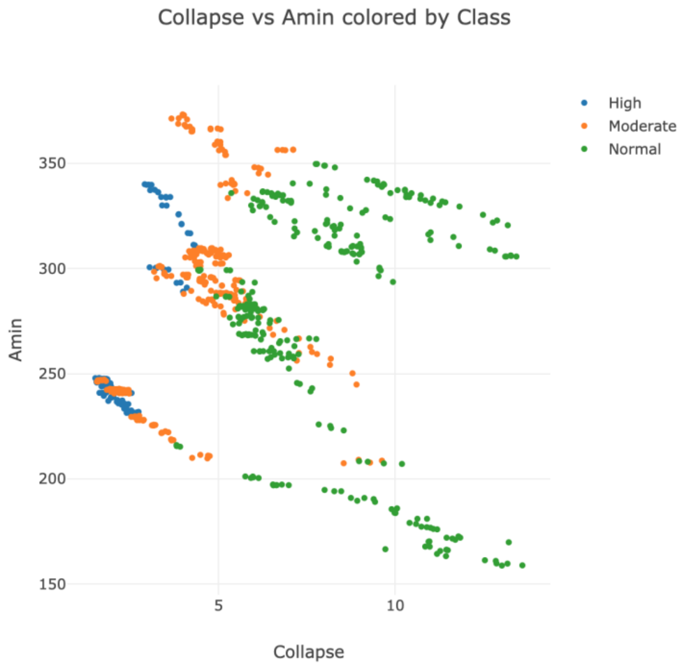


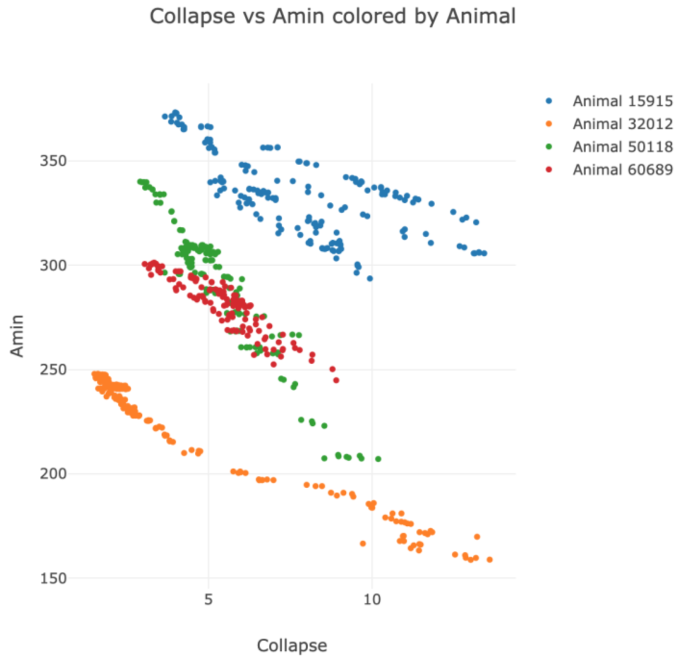


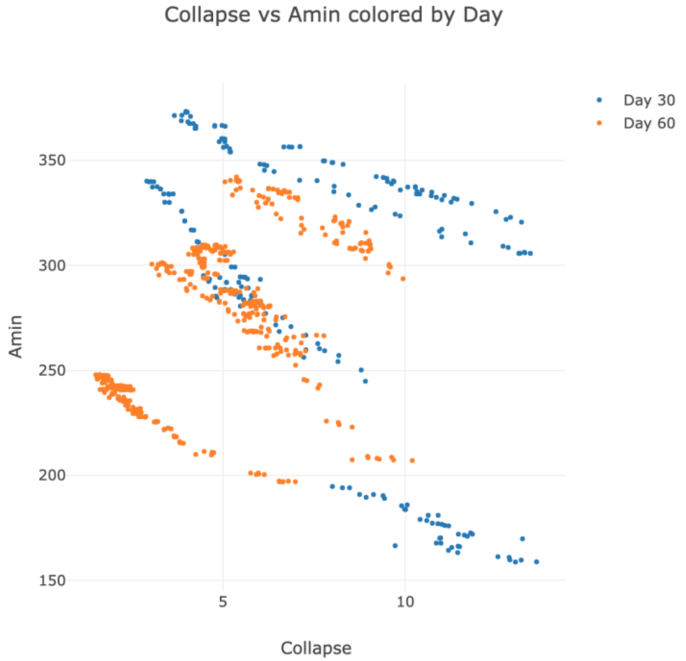
ELDER RESEARCH
DATA SCIENCE • MACHINE LEARNING • AI

Reduced Features

- Allows maximum signal from minimum inputs
- Limit to just Collapse & A_{min}
- A_{max} & A_{min} are highly correlated, so one can be removed
- A_{min} is consistently chosen by XGBoost over A_{max}
- Simplifies deployment
- Applies only to Ensemble Model
- However, we risk reducing robustness against unobserved future sheep (or human) samples







XGBoost Feature Importance

All Animals

1	Collapse	0.692809
2	Amin	0.193777
3	Amax	0.113414

Animal 32012

1	Amin	0.563150
2	Collapse	0.283565
3	Amax	0.153285

Animal 60689

1	Amin	0.563150
2	Collapse	0.283565
3	Amax	0.153285

Animal 50118

1	Amin	0.563150
2	Collapse	0.283565
3	Amax	0.153285

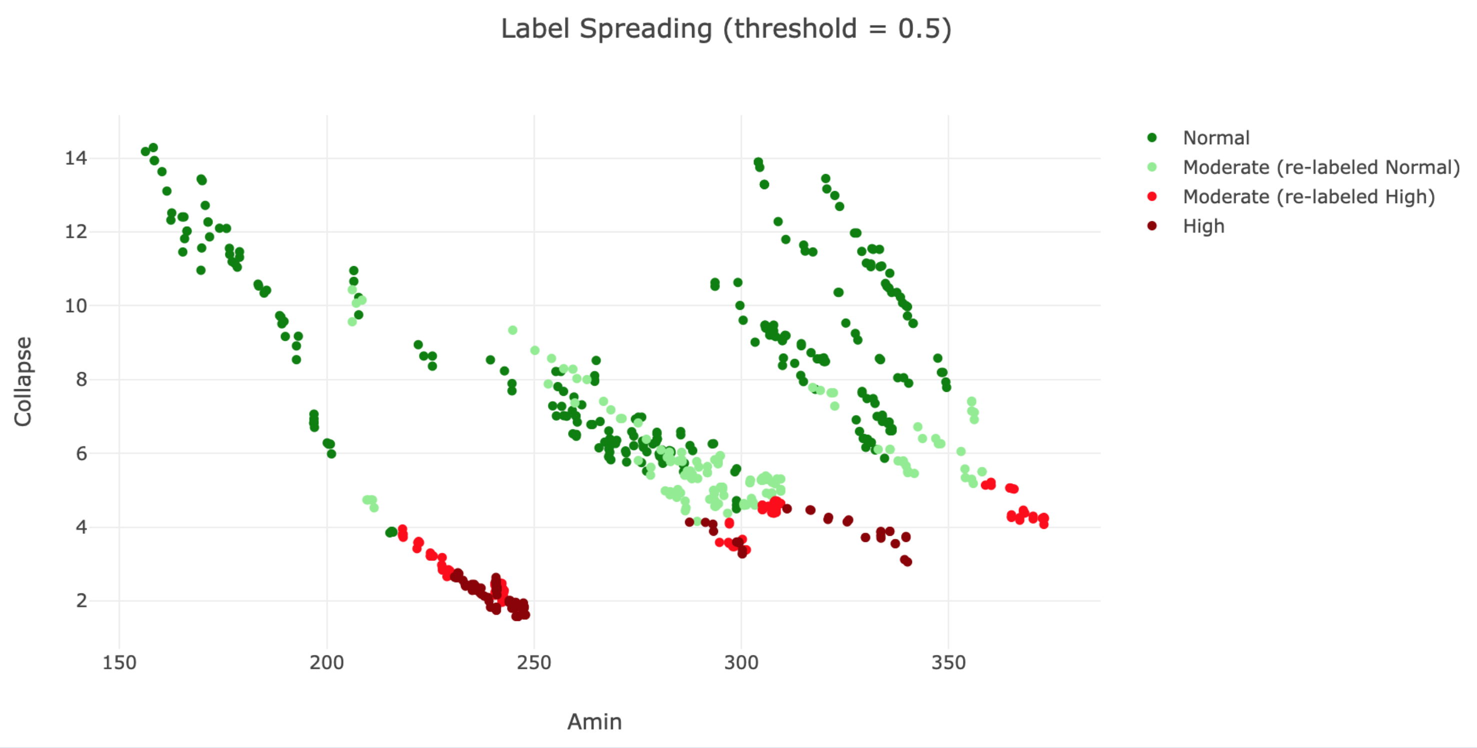
Animal 15915

1	Collapse	0.671775
2	Amin	0.237477
3	Amax	0.090748

Model Design & Build

Label Spreading

- Semi-supervised learning technique
- Treat the Normal and High labels as true labels (high confidence)
- Treat the Moderate labels as unknown labels (low confidence)
- Represents data as a graph where edge weights are node similarity measured by a Radial Basis Function (RBF)
- Relabels all Moderates as either Normal or High (see image)
- Citation: Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, Bernhard Schoelkopf. Learning with local and global consistency (2004)

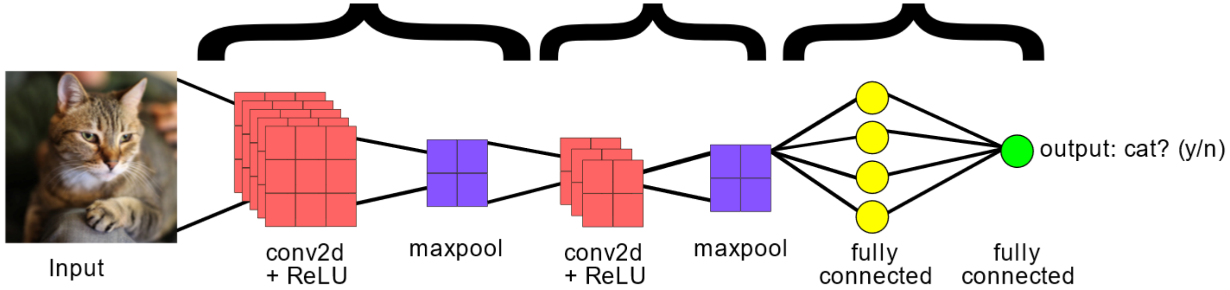


Convolutional Neural Network (CNN)

- Experimented with various options:
  - Window size, stride length
  - Train w/ few traces (3 to 6) from each sheep
  - Leave One Sheep Out (LOSO) vs 80/20 sample validation
  - 3 class (High/Moderate/Normal) classifier
  - Binary classifier (High/Normal)
  - Synthetic labels vs. only true labels
  - Build on 3 sheep (all traces), use transfer learning on 4th sheep using minimal (2 or 3) traces
- Chose Binary classifier w/ synthetic labels
- Transfer learning (simulated patient personalization) offered significant performance gains

ARCHITECTURE & LAYOUT

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_4 (Conv1D)            (None, 200, 100)          5100
_________________________________________________________________
activation_5 (Activation)    (None, 200, 100)          0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 200, 50)           125050
_________________________________________________________________
activation_6 (Activation)    (None, 200, 50)           0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 200, 25)           12525
_________________________________________________________________
activation_7 (Activation)    (None, 200, 25)           0
_________________________________________________________________
global_average_pooling1d_2 ( (None, 25)                0
_________________________________________________________________
dense_4 (Dense)              (None, 100)               2600
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_5 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_6 (Dense)              (None, 2)                 202
_________________________________________________________________
activation_8 (Activation)    (None, 2)                 0
=================================================================
Total params: 155,577
Trainable params: 155,577
Non-trainable params: 0
_________________________________________________________________
```



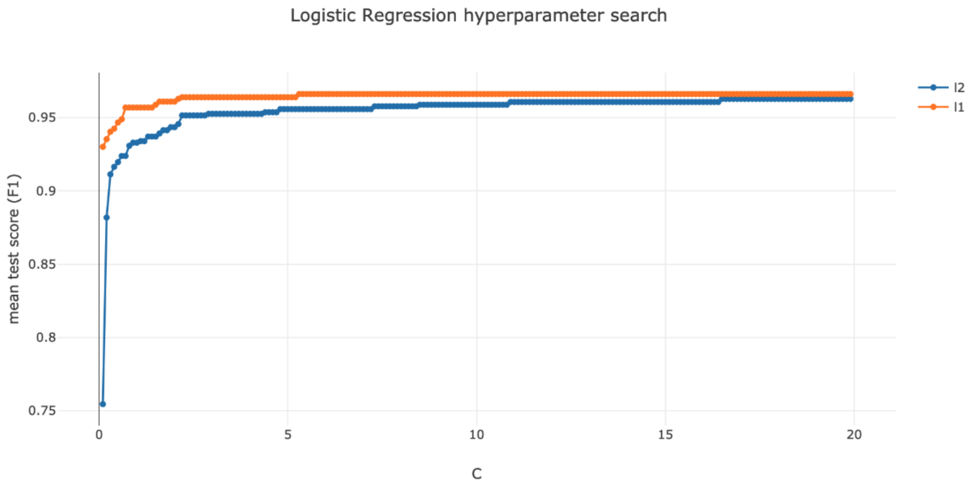
A graphical example of a CNN (note: not the actual architecture)

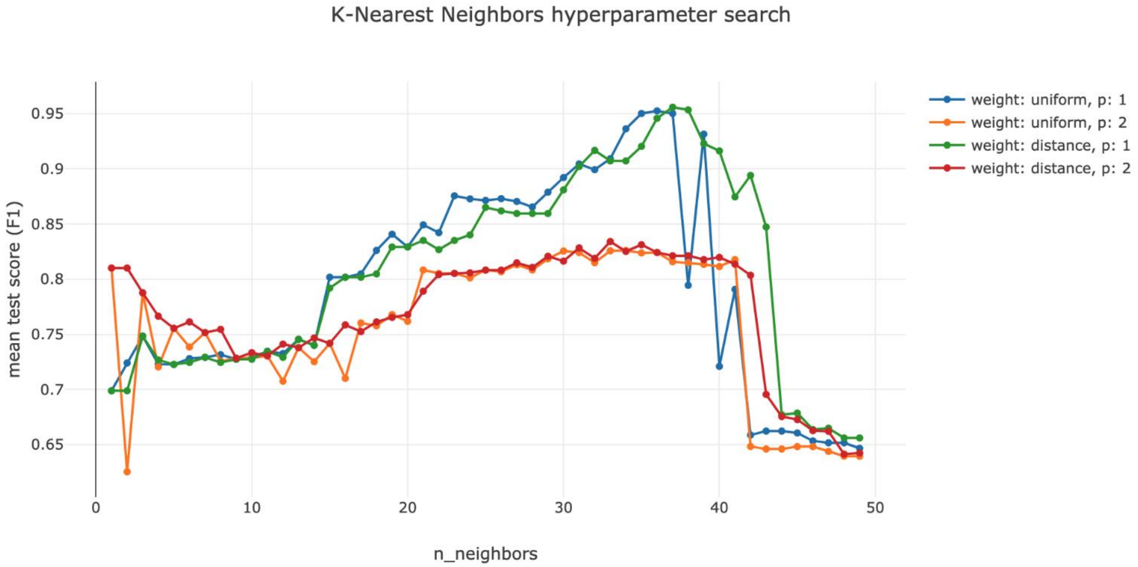
Ensemble Modeling

- Binary classifier (High/Normal) w/ synthetic labels; same as CNN but w/o transfer learning
- Used only Collapse, A_{min} generated from the traces using calculations from Client
- Used voting method to select from multiple models
- Logistic Regression (Logit), Random Forests, K-Nearest Neighbors (KNN), and XGBoost
- Hyperparameter tuning on each model using weighted F1 scores
- Cross-validated grid search weighted KNN three times greater than Logit; rejected the others
- Additional content on Confluence

	w1	w2	w3	w4	voting	rank_test_score	mean_test_score	std_test_score
139	1	0	3	0	soft	1	0.970162	0.013999
134	1	0	2	0	soft	2	0.968262	0.014523
519	4	0	4	0	soft	2	0.968262	0.014523
269	2	0	4	0	soft	2	0.968262	0.014523
129	1	0	1	0	soft	2	0.968262	0.014523

W1: Logit, W2: RandomForest, W3: KNN, W4: XGBoost



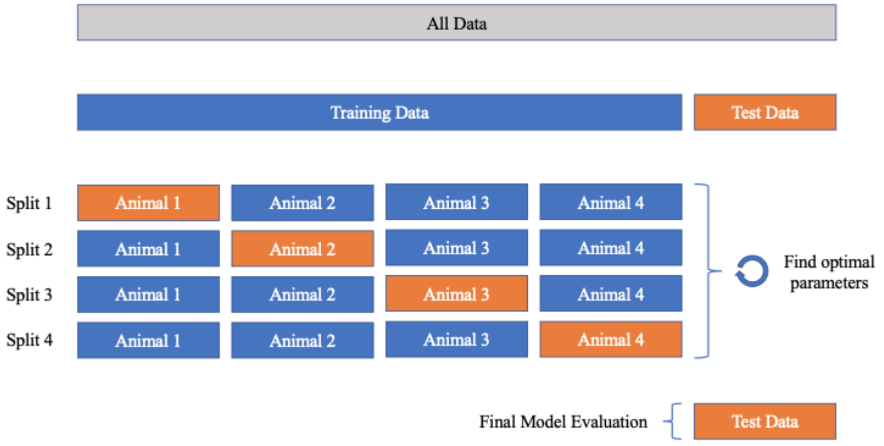


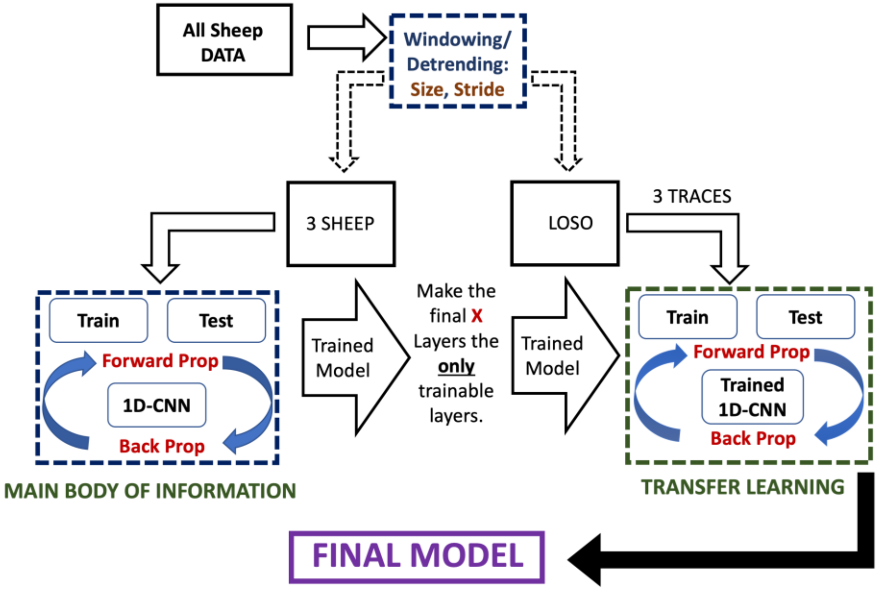
ELDER RESEARCH
DATA SCIENCE • MACHINE LEARNING • AI

Model Validation

Leave One Sheep Out (LOSO) with 80/20 Holdout sample

- Used to avoid model overfit, given only 4 sheep observed
- Ensemble model:
  - Hold out 80/20 train/test split from all traces
  - From training set: conduct parameter tuning on 3 sheep, test on the 4^{th}
  - Rotate this way through all 4 combos
  - Evaluate individual model performance against unseen 20% holdout sample
- CNN has an extra step
  - Build model on 3 sheep
  - Use transfer learning on 2 new traces from 4^{th} sheep to calibrate model
  - One Normal, one High trace
  - Test performance on remaining traces from 4^{th} sheep
  - Simulates new patient in a doctor's office
  - Rotate through all 4 sheep combos

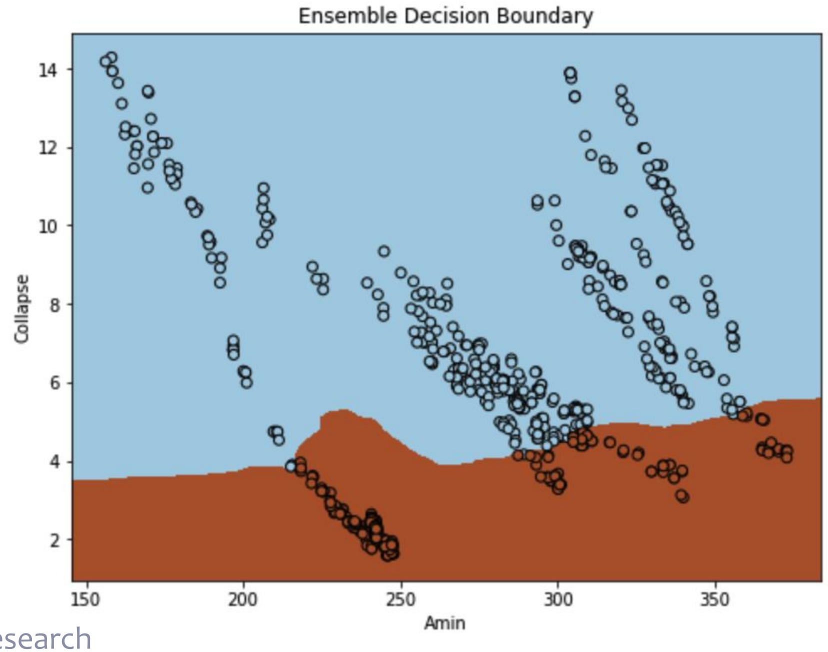


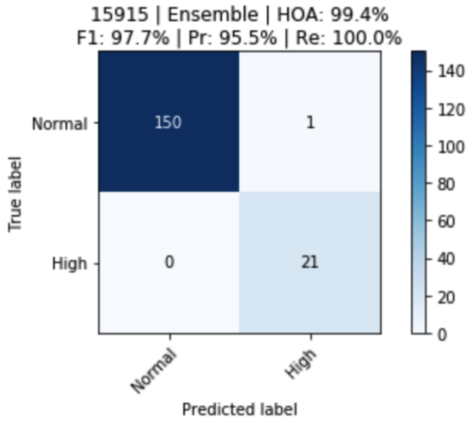


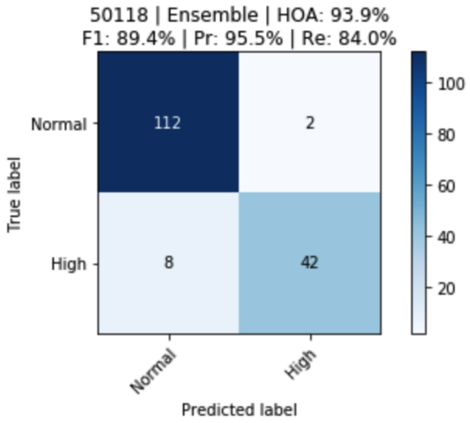
Final Model Performance

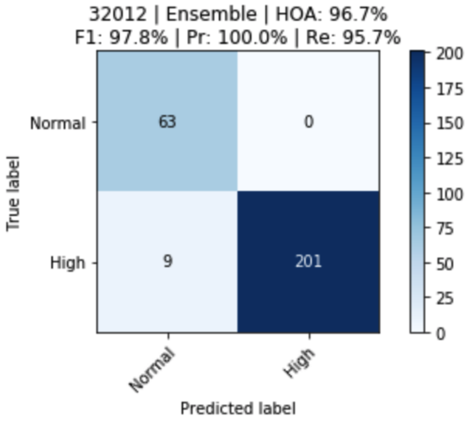
Ensemble Model

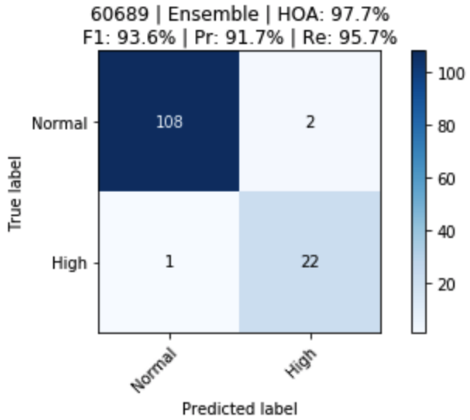
- Final model: binary classifier (high/normal) w/ LOSO & synthetic labels
- Model chose Logit and KNN for final decision boundary
- Very high mean holdout accuracy of 96.93%
  - Mean sensitivity of 93.84%, Mean specificity of 98.94%
- Specific sheep accuracy between 93.9% and 99.4% (see images)
- Additional content on Confluence







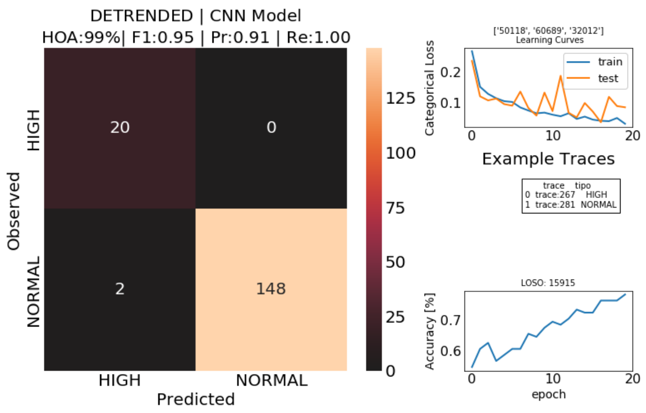




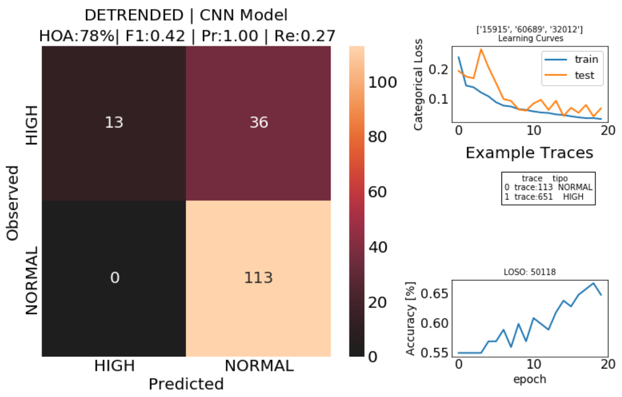
Sheep	HOA	F1	Precision	Recall
1	99%	0.98	0.96	1.00
2	94%	0.89	0.96	0.84
3	97%	0.98	1.00	0.96
4	98%	0.94	0.92	0.96

CNN Model

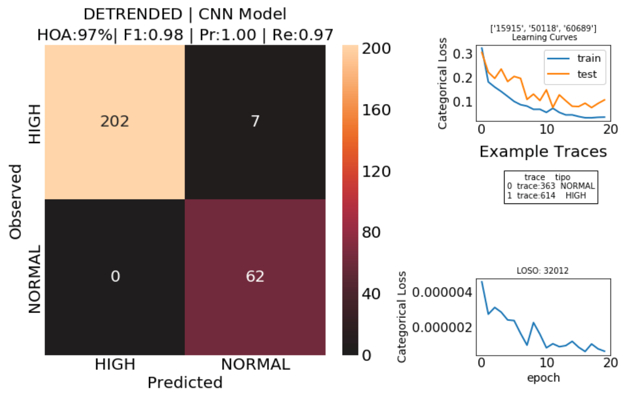
15915



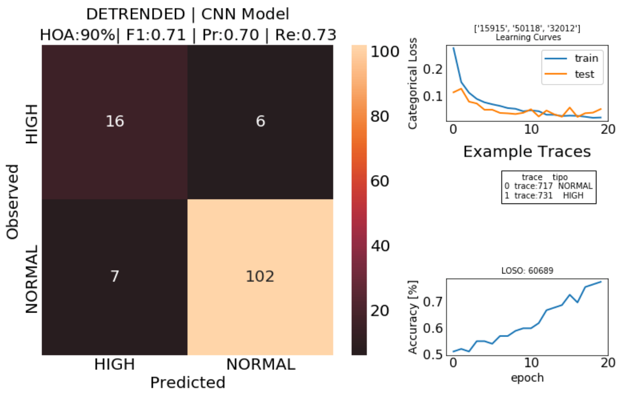
50118



32012*



60689



Average Metrics from 10 Iterations

Sheep	HOA	F1	Precision	Recall
1	87.6%	0.85	0.90	0.89
2	73.3%	0.72	0.72	0.73
3	97.0%	0.97	0.97	0.97
4	90.8%	0.90	0.93	0.89

- Final model: binary classifier (high/normal) w/ transfer learning, LOSO, & synthetic labels
- Very good performance on 3 of 4 sheep
- Requires storing trained model weights calibrated to each patient (but not PII)
- More robust approach to handle future unobserved sheep (patients) through calibration
- Lower overall performance than ensemble model
- Trade off of performance versus robustness (future-proof) solution
- Additional content on Confluence

Model Delivery

Delivery Approaches

- Ensemble Model
  - [Docker container]
  - [Command Line: Serialized Python model]
  - RESTful Web Service: Flask API
  - Input: 2 trace features (A_{min}, Collapse)
  - Output: predicted class (Normal/High) probability [0,1]

- CNN w/ Transfer Learning
  - [Docker container]
  - RESTful Web Service: Flask API
  - Input: JSON array w/ 60 sec trace
  - Output: predicted class (Normal/High) probability [0,1]
  - Calibration step
    - Requires 1 High, 1 Normal trace
    - Could be administered by physician
    - Persists a patient specific CNN
    - Model object contains weights and architecture parameters
    - No PII needed or stored
  - API uses calibrated model object for further trace classifications

Questions?