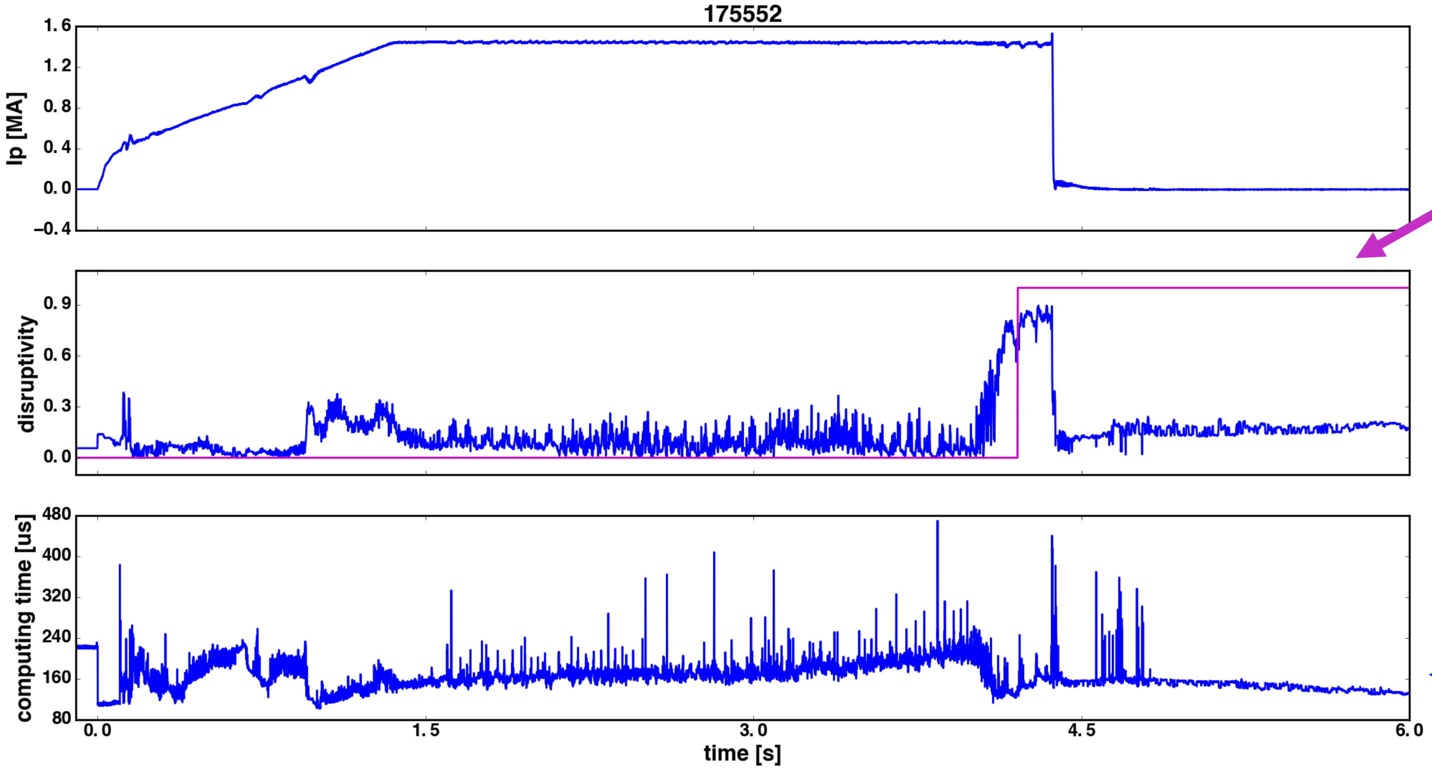
Cross Validation and Interpretation of a Machine-Learning based Disruption Predictor on DIII-D

K. Montes, C. Rea, and R.S. Granetz

Theory and Simulation of Disruptions Workshop, PPPL
July 17th 2018

Machine Learning Disruption Predictor Overview

- <u>Goal:</u> To develop a *robust*, *data-driven* algorithm that successfully *predicts* disruption events with *sufficient warning time*
- Databases of relevant parameters on DIII-D, Alcator C-Mod, EAST, and KSTAR
- Implemented a real-time predictor running in the plasma control system on DIII-D

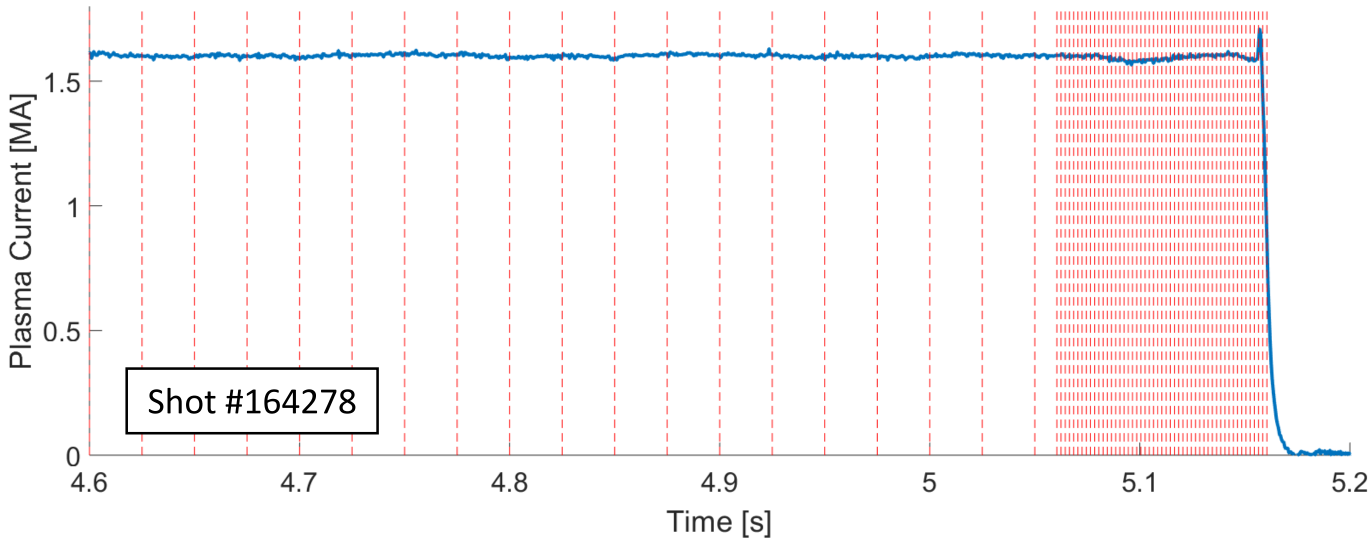


warning alarm
is triggered ~150 ms before the disruption occurs

algorithm computing time ranges between 160-250 microseconds, with spikes depending on the tree depth evaluation for that particular sample

Disruption Warning Database

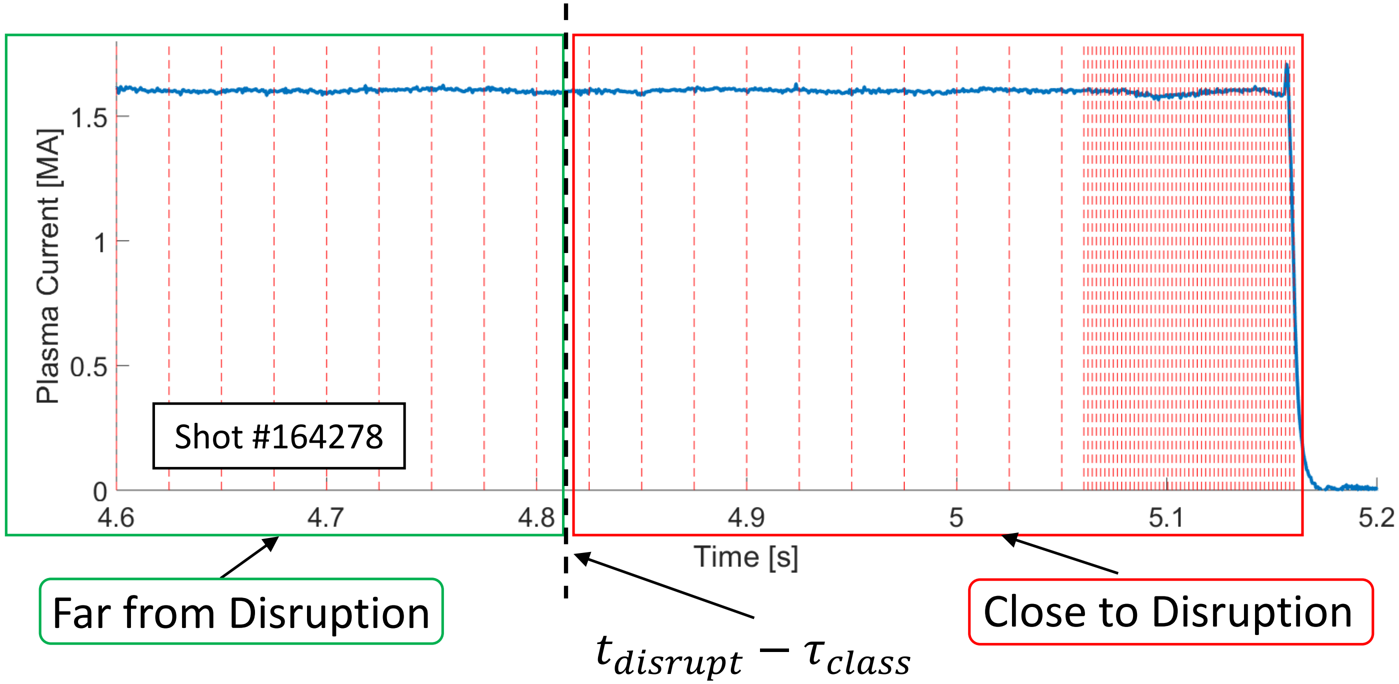
- Focus on dataset of 1258 plasma discharges (disruptive & non-disruptive) from DIII-D 2015 campaign ($\sim 10^5$ time samples)



Signal Description	Variable Name
% error of plasma current and programmed current	$(I_p - I_{prog})/I_p$
Poloidal beta	β_p
Greenwald density fraction	n/n_G
Safety factor at 95% of minor radius	q_{95}
Plasma internal inductance	ℓ_i
Radiated power fraction,	P_{rad}/P_{input}
Loop voltage $[V]$	V_{loop}
Stored plasma energy $[J]$	W_{mhd}
$n = 1$ mode amplitude normalized to B_{tor}	$\Delta B^{n=1}/B_\phi$
T_e profile width normalized to minor radius	T_e/a

Binary Classification Based on Disruptive Phase Assumption

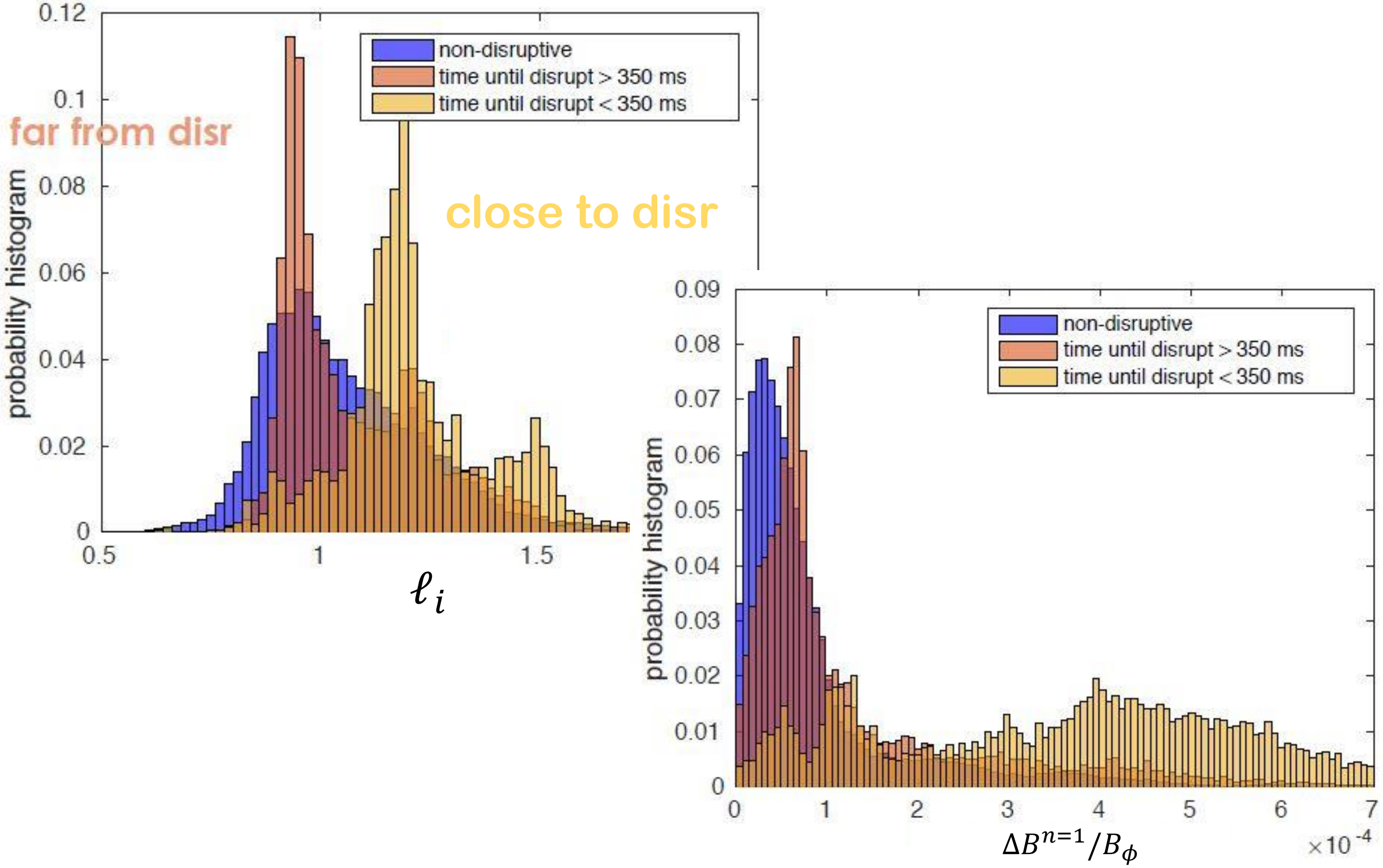
- Focus on dataset of 1258 plasma discharges (disruptive & non-disruptive) from DIII-D 2015 campaign ($\sim 10^5$ time samples)
- Classify sample t using class label threshold, τ_{class}
  - **close to disruption** ($t_{disrupt} - t \leq \tau_{class}$)
  - **far from disruption** (either $t_{disrupt} - t > \tau_{class}$ or sample is from non-disruptive shot)



Signal Description	Variable Name
% error of plasma current and programmed current	$(I_p - I_{prog})/I_p$
Poloidal beta	β_p
Greenwald density fraction	n/n_G
Safety factor at 95% of minor radius	q_{95}
Plasma internal inductance	ℓ_i
Radiated power fraction,	P_{rad}/P_{input}
Loop voltage $[V]$	V_{loop}
Stored plasma energy $[J]$	W_{mhd}
$n = 1$ mode amplitude normalized to B_{tor}	$\Delta B^{n=1}/B_\phi$
T_e profile width normalized to minor radius	T_e/a

Classification of Time-Samples Using Random Forest

- Preliminary analysis – chose $\tau_{class} = 350\ ms$ based on physics parameter distributions
- Published time-sample classification results in recent *Plasma Physics and Controlled Fusion*



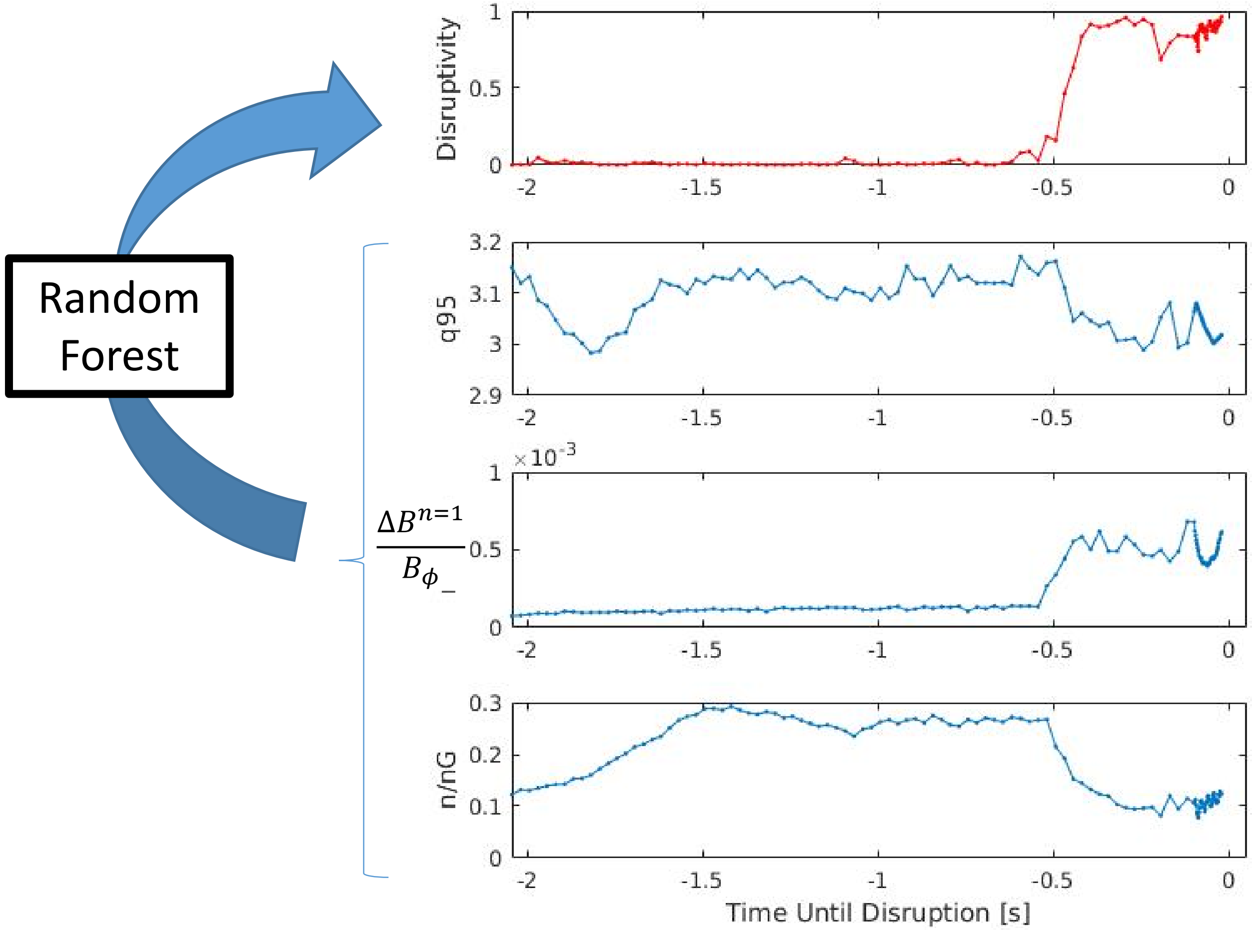
True label \ Predicted label	far from disruption or non-disruptive	close to disruption
far from disruption or non-disruptive	True Negatives 35454	False Positives 220
close to disruption	False Negatives 388	True Positives 1000

[C. Rea et al. PPCF 80 084004 (2018)]

From Time-Sample Predictions to Real-Time Alarms

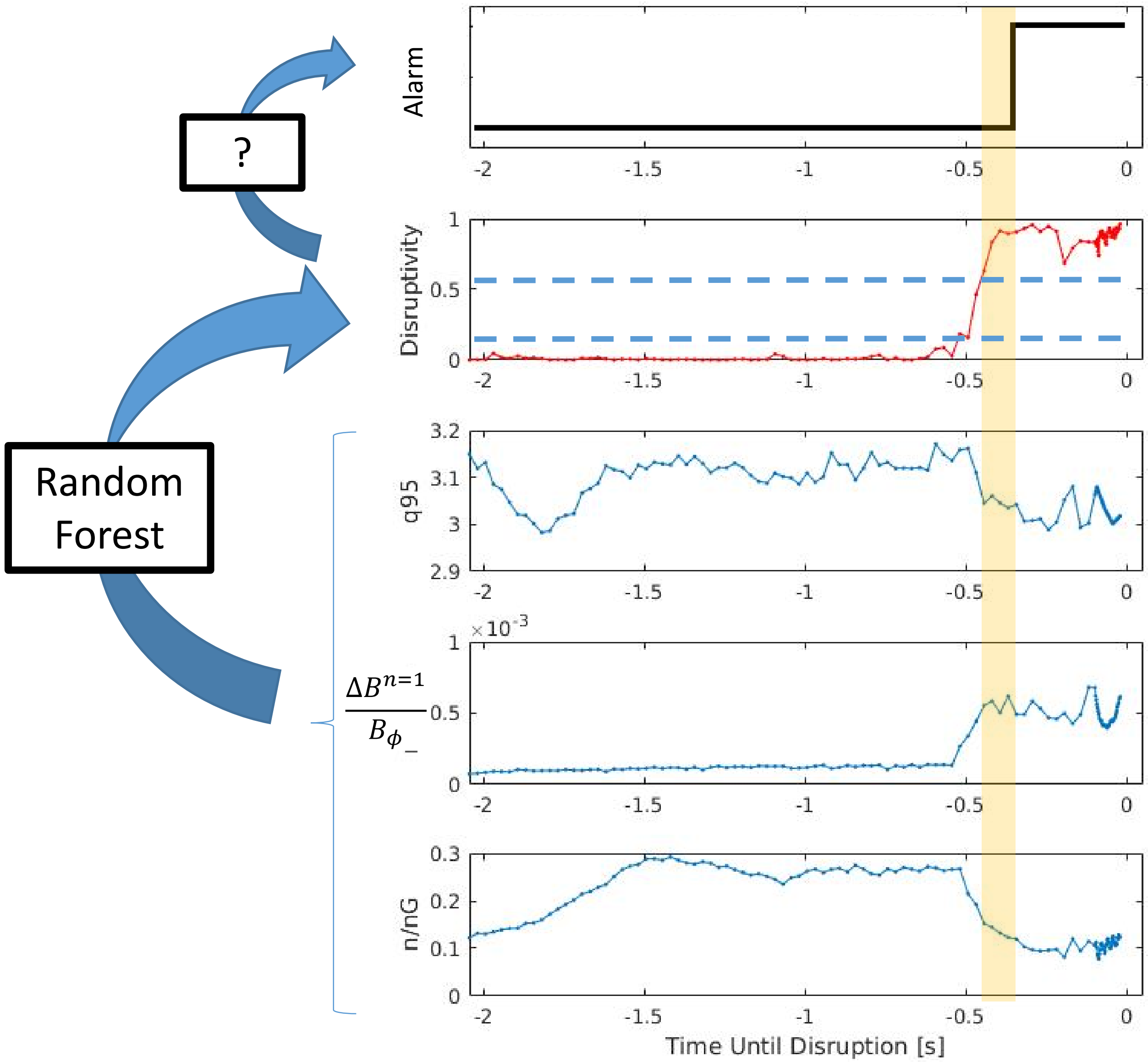
- Each tree in the random forest outputs one of two possible outputs:
  - 0 (far from disruption)
  - 1 (close to disruption)

- Final RF output is the average of the individual tree predictions – we call this the **disruptivity**



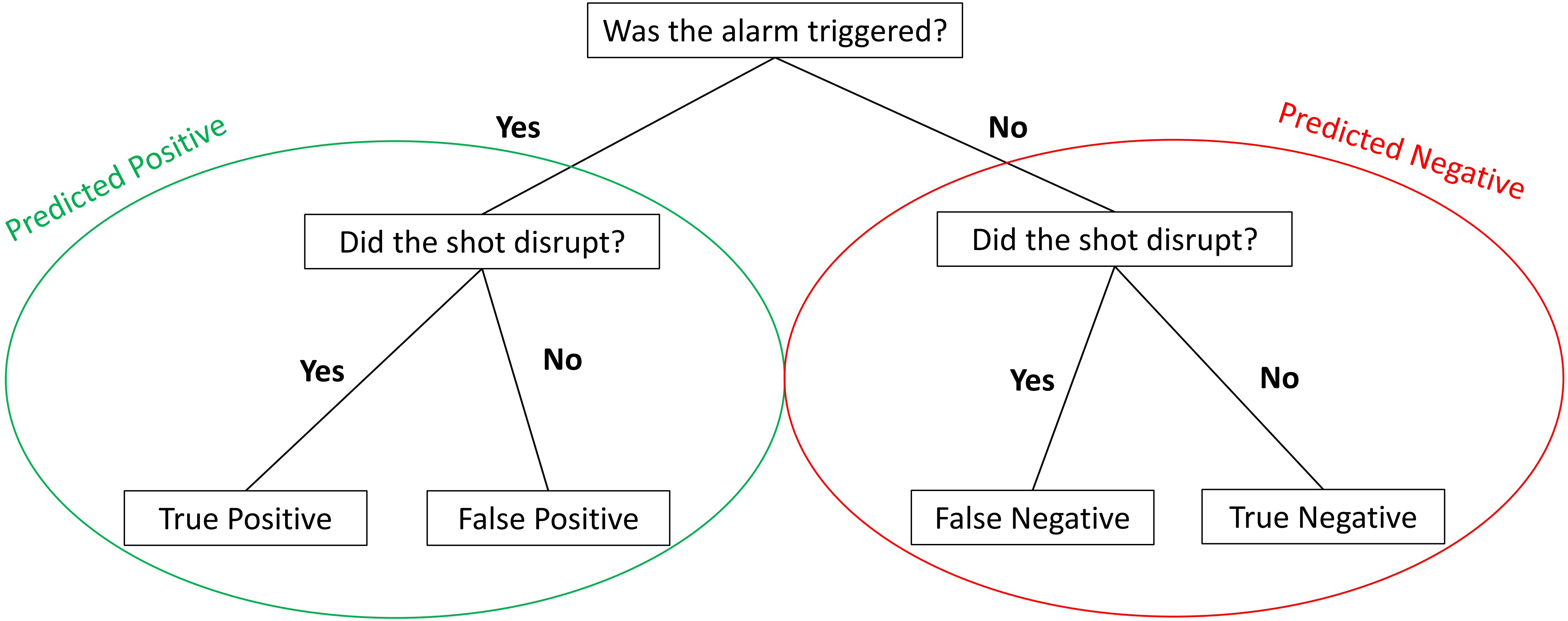
From Time-Sample Predictions to Real-Time Alarms

- Each tree in the random forest outputs one of two possible outputs:
    - 0 (far from disruption)
    - 1 (close to disruption)
- Final RF output is the average of the individual tree predictions – we call this the **disruptivity**
- How do we use the disruptivity to trigger an alarm?
    - Trigger when disruptivity exceeds hysteresis threshold for a specific time window



Shot-by-Shot Binary Classification

- Classify each shot according to whether or not it disrupted:
- **Disruption** (positive class) or **Non-disruption** (negative class)



Parameter Optimization

1. Time-Sample Class Label Threshold (τ_{class})
2. Disruptivity Threshold (d)
3. Time Window Size (w)

RF level

Alarm level (post-processing)

Parameter Optimization

1. Time-Sample Class Label Threshold (τ_{class})
2. Disruptivity Threshold (d)
3. Time Window Size (w)

RF level

Alarm level (post-processing)

Training Set

Test Set

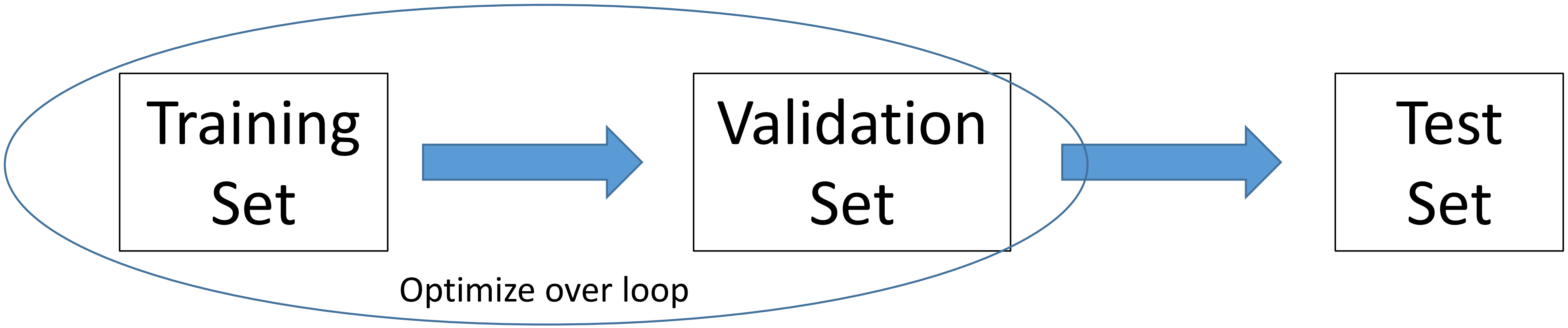
Parameter Optimization

1. Time-Sample Class Label Threshold (τ_{class})
2. Disruptivity Threshold (d)
3. Time Window Size (w)

RF level

Alarm level (post-processing)

- Need cross-validation process to ensure a robust performance metric and the model's generalization capabilities



K-Fold Cross Validation

Iteration 1	Validation	Train	Train	Train	Train
Iteration 2	Train	Validation	Train	Train	Train
Iteration 3	Train	Train	Validation	Train	Train
Iteration 4	Train	Train	Train	Validation	Train
Iteration 5	Train	Train	Train	Train	Validation

Pseudocode:

```
For i in [1,5]:
    For each class label time, τ_class:
        Train random forest on X ≠ X(i)
        Get time-slice predictions on X = X(i)
        For each disruptivity/window pair:
                Test alarm simulation on X = X(i)
                Calculate performance metrics
After loop:
    Average performance metrics over all 5
    iterations for each parameter triplet

    Pick best disruptivity threshold, window, and
    class label time (triplet that maximizes F1)
```

Maximizing the F1 Score (Figure of Merit)

- Grid Search:
  - Disruptivity $d \in [0.1, 0.95]$
  - Alarm Window $w \in [5, 405]\ ms$
  - Class Label Time $\tau_{class} \in [25, 800]\ ms$

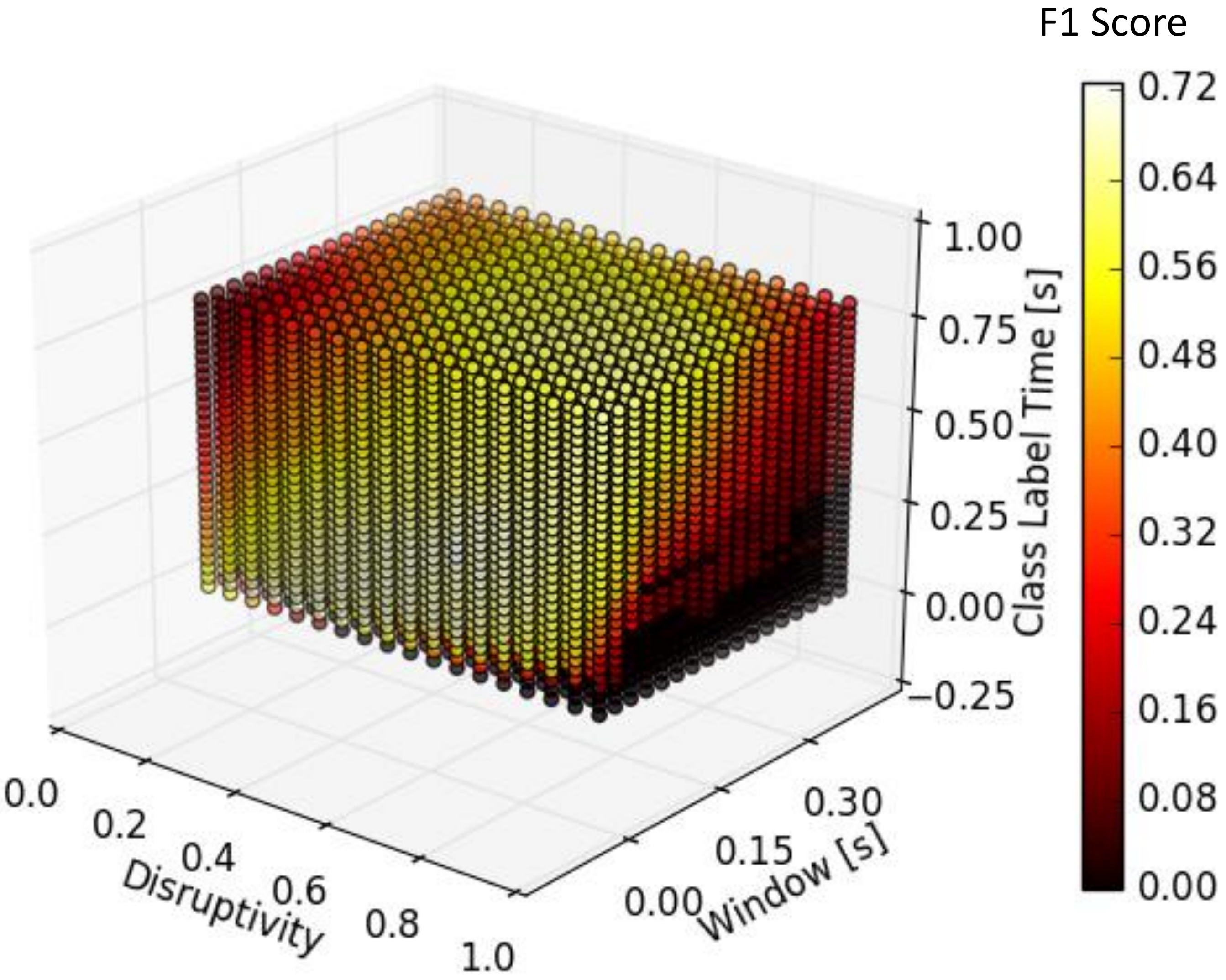
$$Recall = \frac{TP}{TP + FN}$$

(Sensitivity to the **disruptive** class)

$$Precision = \frac{TP}{TP + FP}$$

(Sensitivity to the **non-disruptive** class)

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

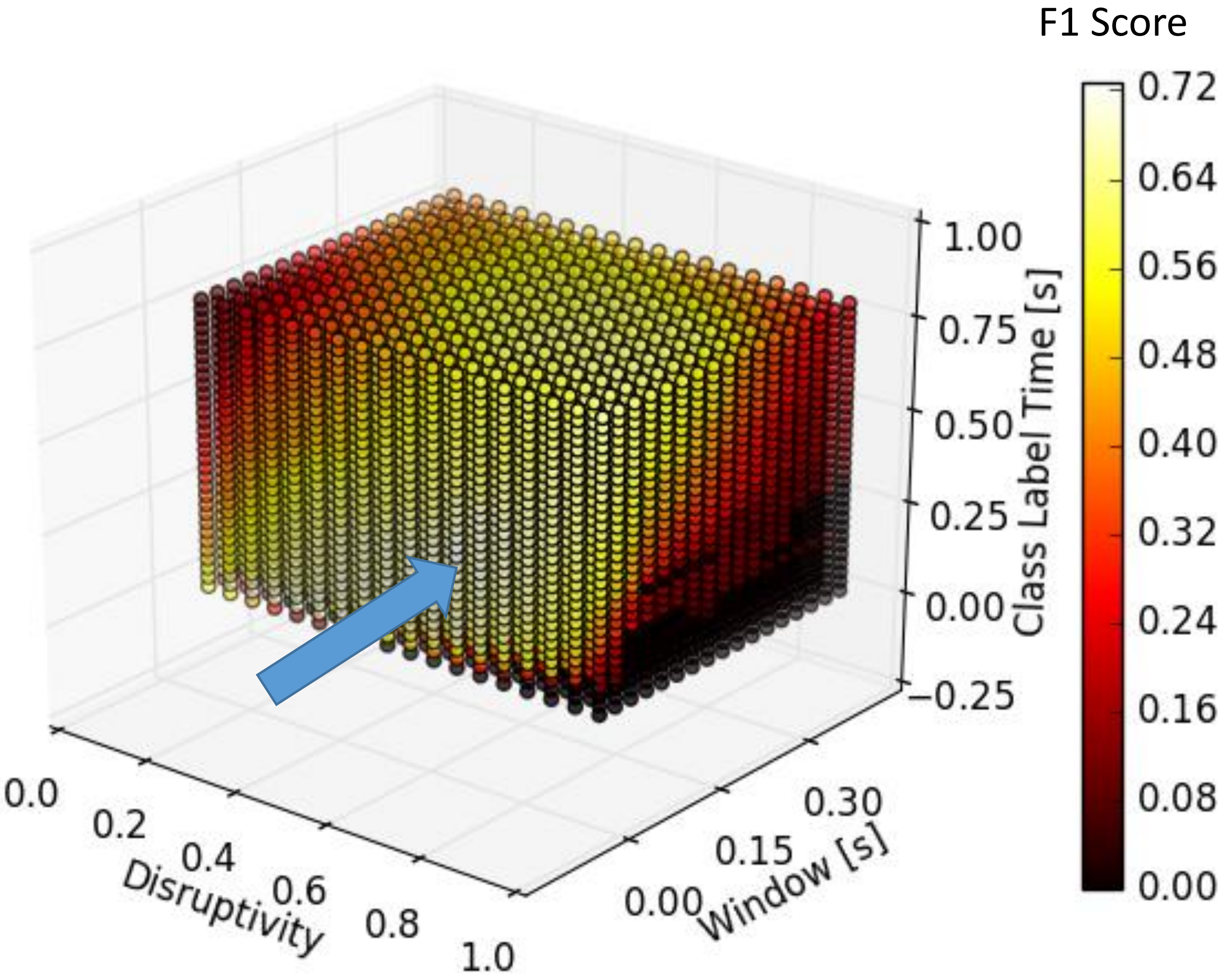


Maximizing the F1 Score (Figure of Merit)

- Best operational point (with highest average F1 score):

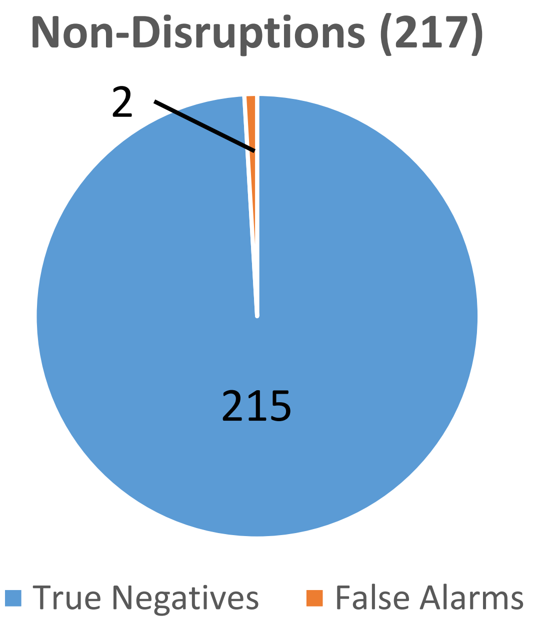
$$[d, w, \tau_{class}] = [0.65, 5ms, 325ms]$$

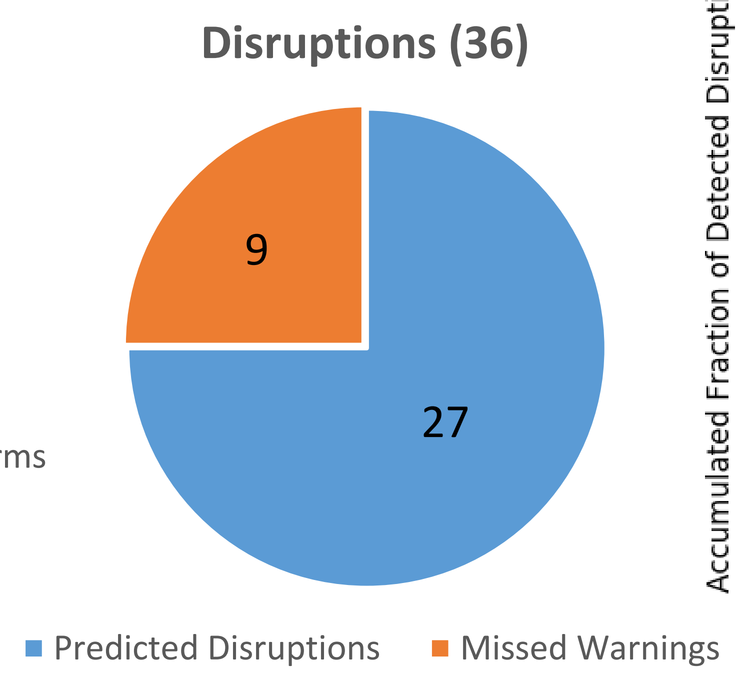
- Shorter alarm windows tend to yield better F1 scores
- Class label time threshold ($\tau_{class} = 325ms$) consistent with univariate analysis

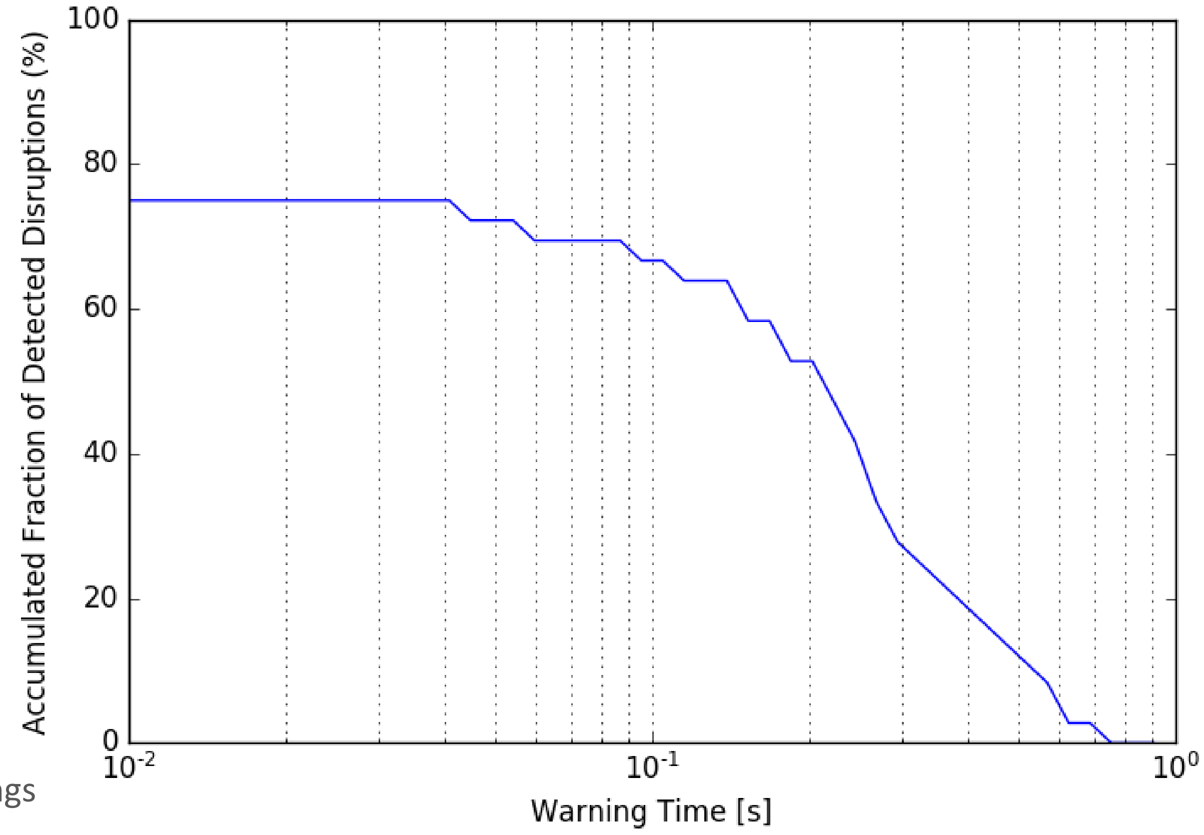


75% of Test Set Disruptions Predicted > 40ms in Advance

- Trained random forest on entire training set using optimized τ_{class}
- Tested random forest predictor on entire test set using optimized d, w

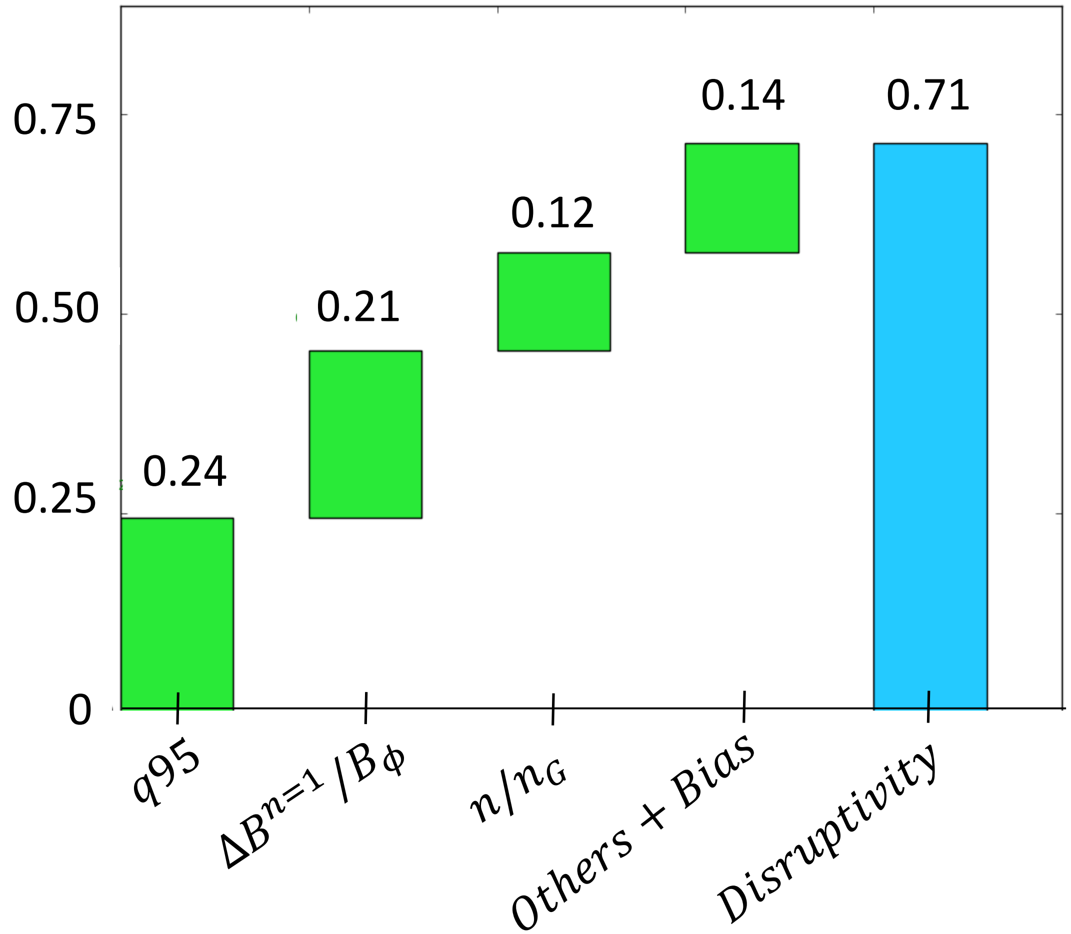


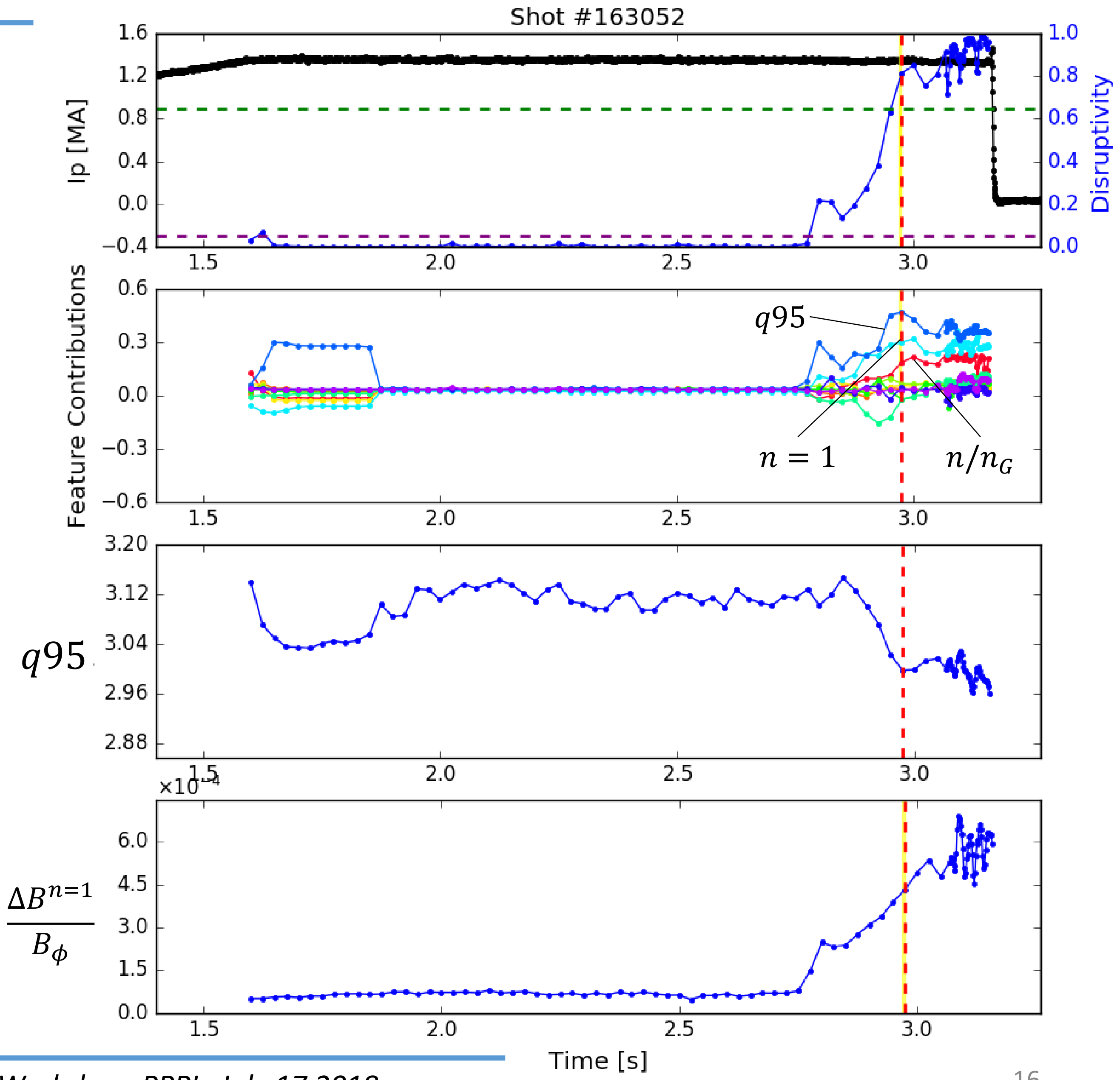




Test Set Interpretability

Average Contributions for all Predicted Disruptions

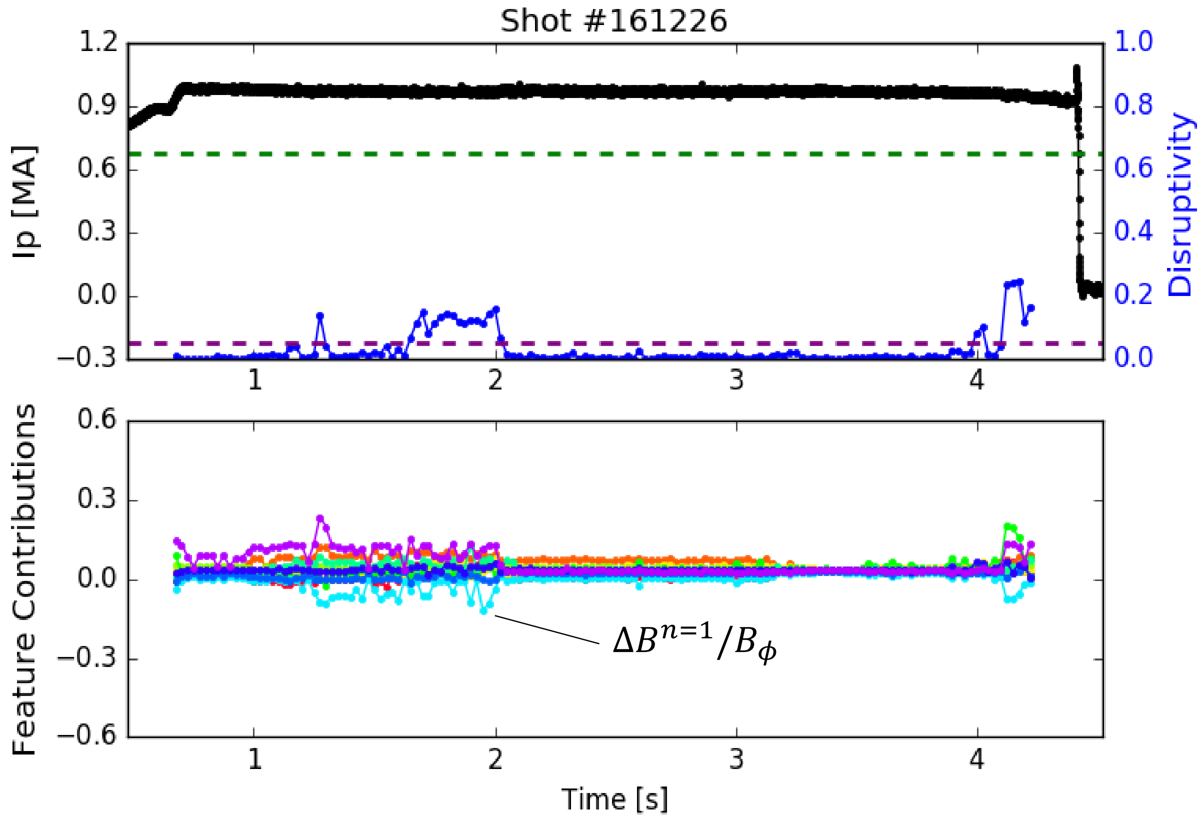




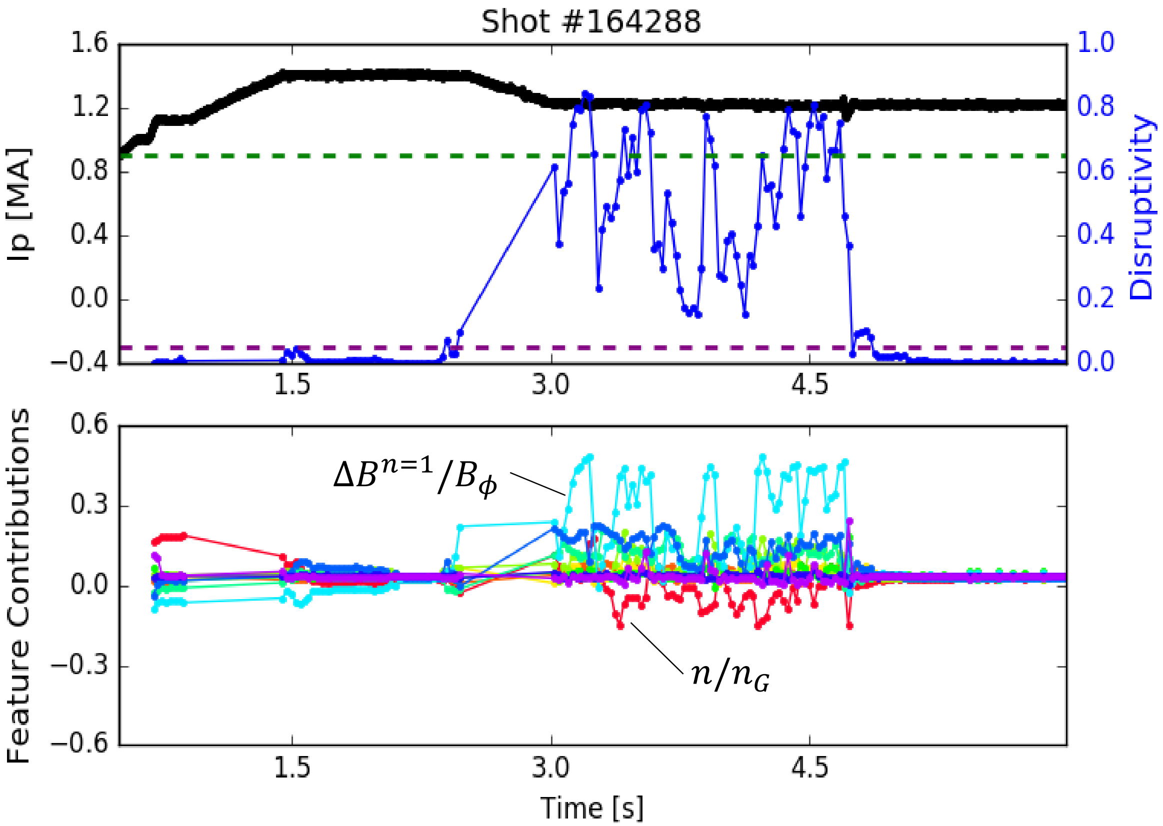
False Predictions

- Most false predictions show small or negative contributions from $q95$, the normalized $n = 1$ radial field component, and/or n/n_G

<u>False Negative</u>



<u>False Positive</u>



Summary

- Our data-driven cross-validation procedure validates our univariate analysis of the distinction between ‘disruptive’ and ‘non-disruptive’ phases on plasma discharges
- Of the 10 signals in our DIII-D 2015 database, the 3 most relevant are:
  1. $q95$
  2. $\Delta B^{n=1}/B_\phi$
  3. n/n_G
- Our model runs at very low cost (low false positive rate), and predicts $\approx 75\%$ of disruptions

Future Work

- Improve cross-validation procedure with a time-dependent metric, so that the ‘best’ operational point is a function of the physics parameters
- Compare results to an algorithm that incorporates time-dependency
- Test robustness of results by applying to larger database of different campaigns and facilities
- Expand set of input physics parameters

Backup Slides

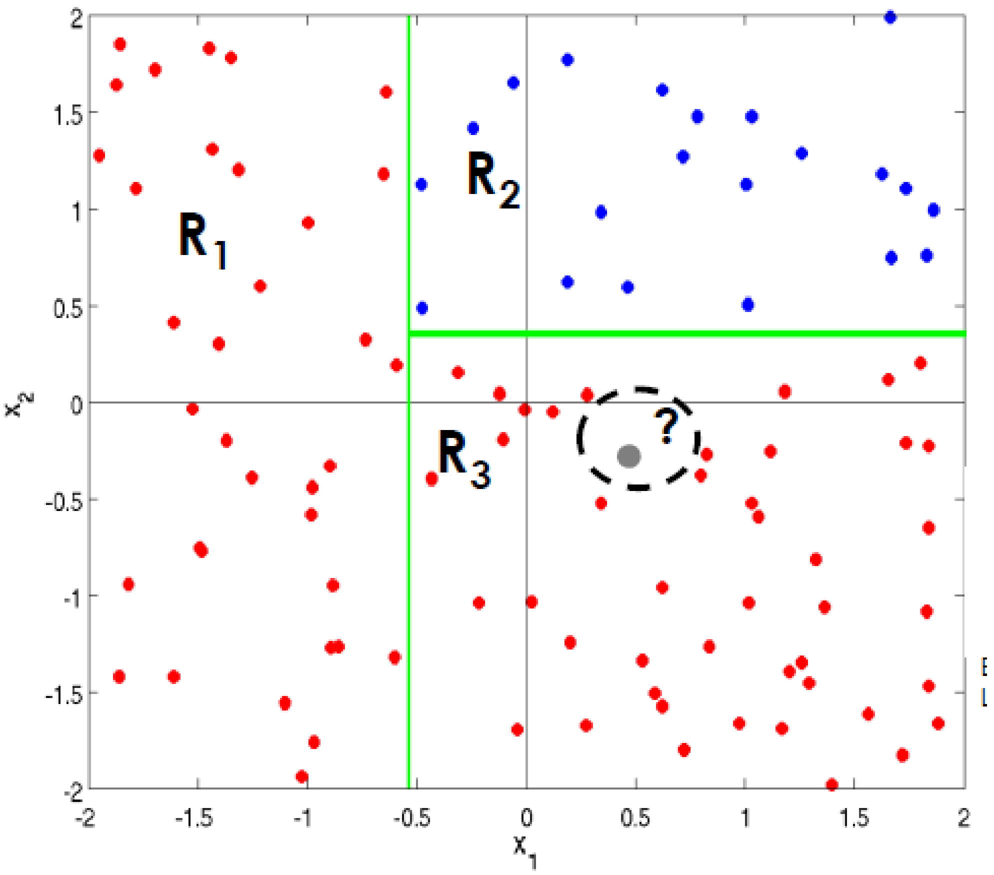
Disruption Warning Database

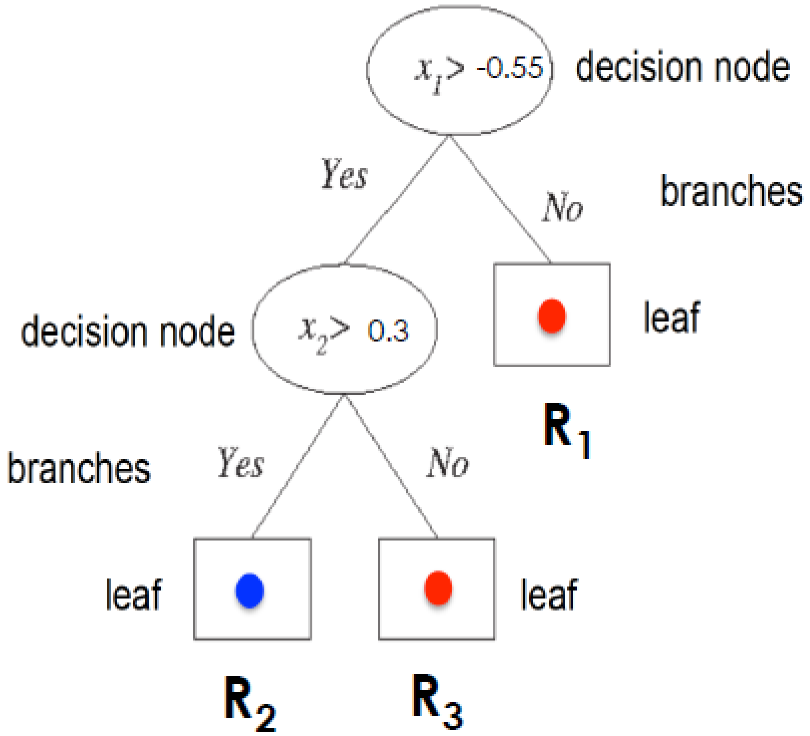
- SQL databases with Matlab, IDL, and Python queries
- All disruptions included, regardless of cause
- ~ 40 plasma parameters at each time sample/record
- Parameters potentially available in real time
- During training, we avoid using…
  - Non-causally filtered data
  - Intentional disruptions
  - Disruptions caused by hardware failure (specifically check for feedback control on plasma current or UFOs events)
  - Time samples not in the flattop phase

Device	Discharges	Time Samples
C-Mod	5507	498,925
EAST	14713	1,209,217
DIII-D	10258	2,356,519
KSTAR	4219	773083

Random Forest

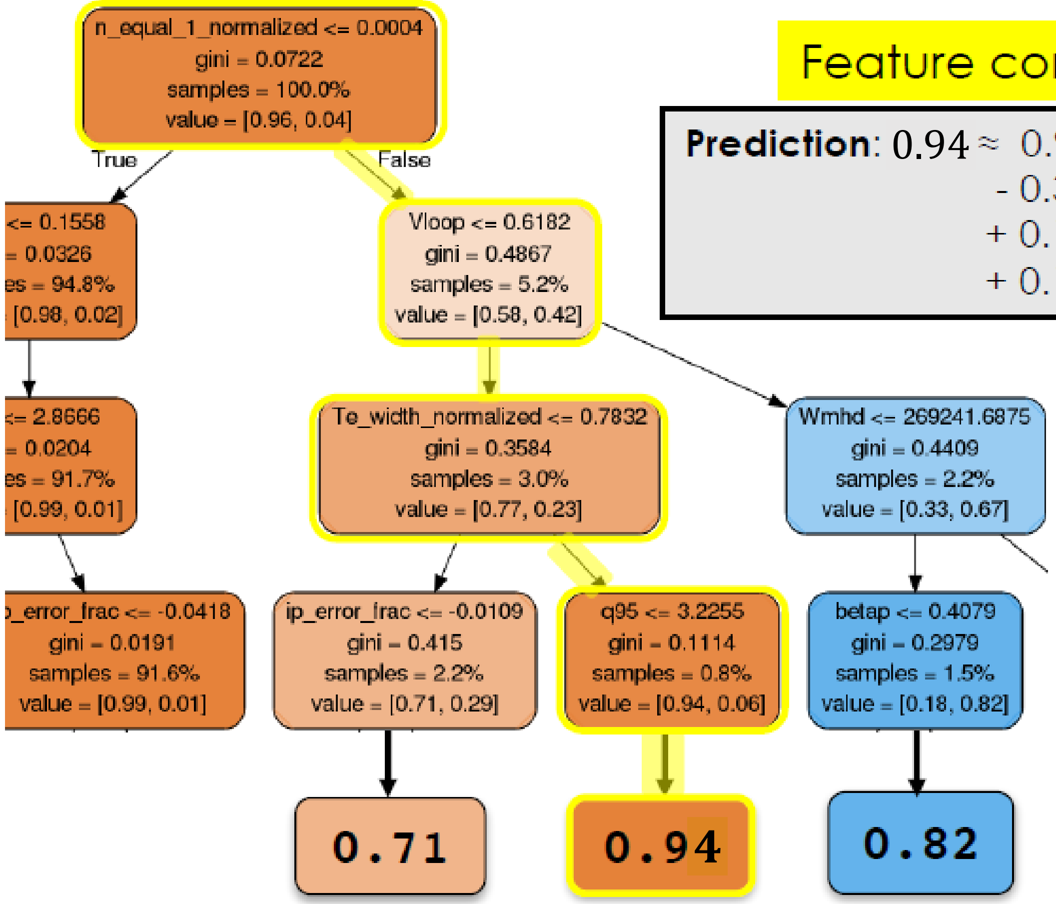
- An ensemble of many uncorrelated **classification and regression trees**
- At each node in the each tree, the data set is split on a random feature by **minimizing impurity**





Test Set Interpretability

- For feature vector x, can express **disruptivity** $f(x)$ as sum of K feature contributions & bias term
- Tracking feature contributions can give idea of drivers of disruptive behavior



Feature contribution breakdown:

Prediction: 0.94 ≈ 0.96 (**model bias**)
- 0.38 (**loss**, `Vloop`)
+ 0.19 (**gain**, `Te_width_normalized`)
+ 0.17 (**gain**, `q95`)

Prediction function for one tree:

$$f(x) = b + \sum_{k=1}^{K} contrib(x, k)$$

Prediction function for a *forest* of J trees:

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} b_j + \sum_{k=1}^{K}\left(\frac{1}{J}\sum_{j=1}^{J} contrib_j(x, k)\right)$$

Real-Time Implementation

- Has run continuously in DIII-D PCS for more than 850 discharges
  - 66% non-disruptive
  - 6% flattop disruptions
  - 28% rampdown disruptions
- Feature contributions potentially available in real time for interpretation
- Low false positive rate $(< 4\%)$ on non-disruptive discharges