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**

- **Goal**: 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 (∼ $10^5$ time samples)

<table>
<thead>
<tr>
<th>Signal Description</th>
<th>Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>% error of plasma current and</td>
<td>$(I_p - I_{prog})/I_p$</td>
</tr>
<tr>
<td>programmed current</td>
<td></td>
</tr>
<tr>
<td>Poloidal beta</td>
<td>$\beta_p$</td>
</tr>
<tr>
<td>Greenwald density fraction</td>
<td>$n/n_G$</td>
</tr>
<tr>
<td>Safety factor at 95% of minor radius</td>
<td>$q_{95}$</td>
</tr>
<tr>
<td>Plasma internal inductance</td>
<td>$\ell_i$</td>
</tr>
<tr>
<td>Radiated power fraction, $P_{rad}/P_{input}$</td>
<td></td>
</tr>
<tr>
<td>Loop voltage [V]</td>
<td>$V_{loop}$</td>
</tr>
<tr>
<td>Stored plasma energy [J]</td>
<td>$W_{mhd}$</td>
</tr>
<tr>
<td>$n = 1$ mode amplitude normalized to $B_{tor}$</td>
<td>$\Delta B_{n=1}^{\phi}/B_\phi$</td>
</tr>
<tr>
<td>$T_e$ profile width normalized to minor radius</td>
<td>$T_e/a$</td>
</tr>
</tbody>
</table>
Binary Classification Based on Disruptive Phase Assumption

- Focus on dataset of 1258 plasma discharges (disruptive & non-disruptive) from DIII-D 2015 campaign (∼ $10^5$ time samples)
- Classify sample $t$ using class label threshold, $\tau_{class}$
  - 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)

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

K. Montes – Theory and Simulation of Disruptions Workshop, PPPL, July 17 2018
Classification of Time-Samples Using Random Forest

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

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

K. Montes – Theory and Simulation of Disruptions Workshop, PPPL, July 17 2018
• 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
• 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)

```
Was the alarm triggered?

Yes

Did the shot disrupt?

Yes

True Positive

No

False Positive

No

Did the shot disrupt?

Yes

False Negative

No

True Negative
```
Parameter Optimization

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

RF level
Alarm level (post-processing)
Parameter Optimization

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

Training Set  $\rightarrow$  Test Set

RF level  $\leftrightarrow$  Alarm level (post-processing)
Parameter Optimization

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

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

---

K. Montes – Theory and Simulation of Disruptions Workshop, PPPL, July 17 2018
K-Fold Cross Validation

Pseudocode:

For i in [1,5]:
    For each class label time, \( \tau_{\text{class}} \):
        Train random forest on \( X \neq X(i) \)
        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)

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

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

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

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

(Sensitivity to the disruptive class)  
(Sensitivity to the non-disruptive class)
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 $\tau_{\text{class}}$
- Tested random forest predictor on entire test set using optimized $d, w$

Non-Disruptions (217)

- 215 True Negatives
- 2 False Alarms

Disruptions (36)

- 27 Predicted Disruptions
- 9 Missed Warnings
Test Set Interpretability

Average Contributions for all Predicted Disruptions

\[
\Delta B^{n=1}/B_\phi = 0.24
\]
\[
n/n_G = 0.21
\]
\[
\text{Others + Bias} = 0.12
\]
\[
\text{Disruptivity} = 0.71
\]
False Predictions

- Most false predictions show small or negative contributions from $q_{95}$, the normalized $n = 1$ radial field component, and/or $n/n_G$.
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. $q_{95}$
  2. $\Delta B_{n=1}^n / B_\phi$
  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
Disruption Warning Database

- SQL databases with Matlab, IDL, and Python queries
- All disruptions included, regardless of cause
- \( \sim 40 \) 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

<table>
<thead>
<tr>
<th>Device</th>
<th>Discharges</th>
<th>Time Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Mod</td>
<td>5507</td>
<td>498,925</td>
</tr>
<tr>
<td>EAST</td>
<td>14713</td>
<td>1,209,217</td>
</tr>
<tr>
<td>DIII-D</td>
<td>10258</td>
<td>2,356,519</td>
</tr>
<tr>
<td>KSTAR</td>
<td>4219</td>
<td>773083</td>
</tr>
</tbody>
</table>
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

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