Initial Results of a Machine Learning-based Real Time Disruption Predictor on DIII-D

C. Rea[1], K. Erickson[2], R.S. Granetz[1], R. Johnson[3], N. Eidietis[3], K. Montes[1], R.A. Tinguely[1]

[1] *MIT Plasma Science and Fusion Center, Cambridge, MA, US*

[2] *Princeton Plasma Physics Laboratory, Princeton, NJ, US*

[3] *General Atomics, San Diego, CA, US*

A disruption prediction algorithm, named DPRF (Disruption Prediction using Random Forests), has recently been implemented to run in real time in the DIII-D plasma control system (PCS). DPRF is developed on the basis of the Machine Learning Random Forests (RF) algorithm and using an extensive database of more than 10 000 DIII-D discharges, both disruptive and non-disruptive. The algorithm uses 9 plasma parameters that are derived from several real time diagnostic signals and real time EFIT equilibrium reconstructions. The list of used signals is provided in Table 1. DPRF is trained on all types of major disruptions occurring during the flattop phase, without differentiation by cause, and follows a binary classification scheme: the algorithm is trained to recognize time samples *close to* the disruption event (i.e. during the 350 ms preceding the disruption) and samples that are non-disruptive or *far from* the disruption. We followed the assumption that discharges that eventually disrupt present a transition from a safe to a disruptive phase in the plasma parameter space. Thanks to the RF *white box* features, DPRF provides probabilities associated to its predictions, i.e. a disruptivity signal, now incorporated in the DIII-D PCS. RF also provides a way to interpret the prediction results (e.g., which signals contributed to triggering an alarm). By identifying the causes underlying the disruption events, a better understanding of disruption dynamics is achieved, and a clear path toward the design of disruption avoidance strategies can be provided.

The database and Random Forest model. The application to the DIII-D PCS developed from an extensive work dedicated to the construction of a database that collected a wide range of different plasma parameters during several years of DIII-D experimental campaigns (from 2014 through 2017). Our choice of parameters to include in the database is based partly on our own tokamak operational experience, and partly on those specified in the relevant literature [1–4]. A detailed description of the database construction can be found in Section 2 of [5, 6]. With respect to [6], DPRF is trained using a much larger set of data that comes mainly from the raw real time environment, given that DPRF is intended to be a real-time disruption warning application. Therefore, we also sub-selected from the database only a reduced number of parameters, available in real time to the PCS. These are reported in Table 1. The details of the RF algorithm [7] have already been discussed in previous papers [5, 6], to which we refer for a more detailed discussion of the algorithmic methodology and all the derived implications.

Variable name	Importance*	Signal description
n_equal_1_normalised	0.198	$B_r^{n=1}$ perturbed radial field of nonrotating modes, normalised to B_{tor}
q95	0.184	Safety factor at the 95% flux surface, q_{95}
n/nG	0.165	Greenwald density fraction, n/n_G
ip_error_frac	0.111	Fractional error between measured and programmed plasma current, $(I_p - I_{prog})/I_p$
li	0.093	Normalised internal inductance, ℓ_i
betap	0.079	Poloidal beta, β_p
Vloop	0.062	Loop voltage, V_{loop} [V]
Wmhd	0.061	Stored plasma energy, W_{th} [J]
Te_width_normalised	0.047	Electron temperature profile width, normalised to plasma minor radius

Table 1: *List of signals used for the development of DPRF on DIII-D PCS.*

* *The signals are ordered according to the values of their relative importance in the trained model. The importance metric is a feature of tree-based models; for a detailed explanation, please refer to [5, 6].*

Summarizing, the forests are developed by growing a large number of independent, de-correlated decision trees, thus collecting a parallel set of predictions. The trees are usually fully grown: starting from a root node, the decision paths are obtained through bootstrapped samples of the input features (i.e., the plasma signals from Table 1) and develop branches that partition such features on the basis of their real values (no feature scaling or normalization is actually required). The final prediction is aggregated, using majority voting, from a large number of trees. DPRF is trained using a forest of 500 decision trees, and this number was chosen on the basis of the Out-Of-Bag error rate stabilization. Tree-based models are attractive algorithms due to their accessible interpretability: using the Gini impurity measure it is possible to obtain an estimate of the relative importance of the predictor variables. The second column in Table 1 reports on the relative importance ranking extracted from the training set used to develop DPRF.

Real Time Implementation. DPRF is trained using `scikit-learn` [8], the open-source Python library, through the OMFIT framework [9]. To integrate DPRF in the real time DIII-D environment, the trained forest was translated into C, the PCS-compatible language[1]. In Figure 2 we also show the time traces of three of the most relevant input features: the Greenwald density fraction n/n_G, the safety factor $q95$, and the *n_equal_1_normalised* locked mode indicator.

[1] `sklearn-porter`, https://github.com/nok/sklearn-porter, D. Morawiec, unpublished.

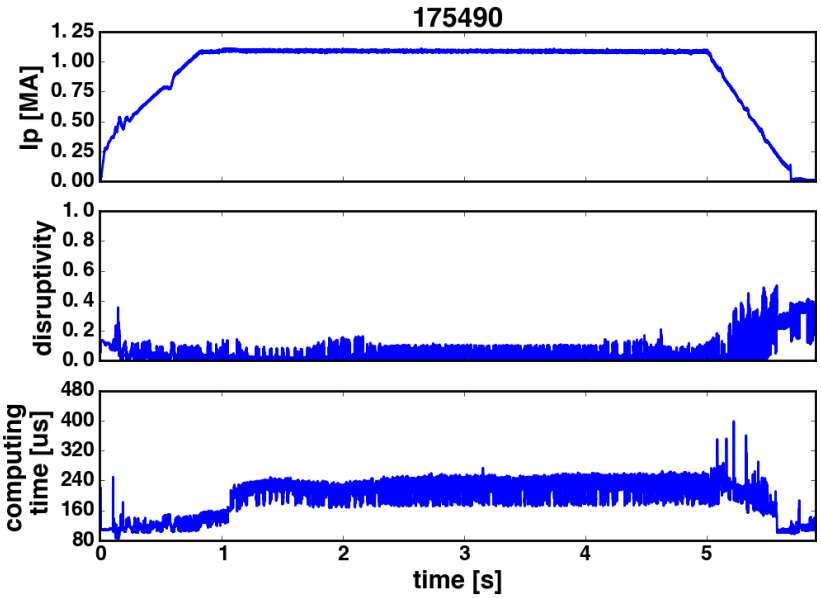

Figure 1: *Example of a DIII-D non-disruptive discharge, shot 175490. The disruptivity level stays below 20% for the whole duration of the flattop phase.*

Figures 1 and 2 show two examples of discharges during which DPRF ran its background calculations. The disruptivity predictions are shown in the second panels of both figures, together with the average computing time for DPRF predictions (ranging around 250-300 μs). It is interesting to notice that, for this particular case, the disruptivity signal increases above 60% before the impending disruption (represented by the dashed red vertical line across the panels) with more than 150 ms warning time.

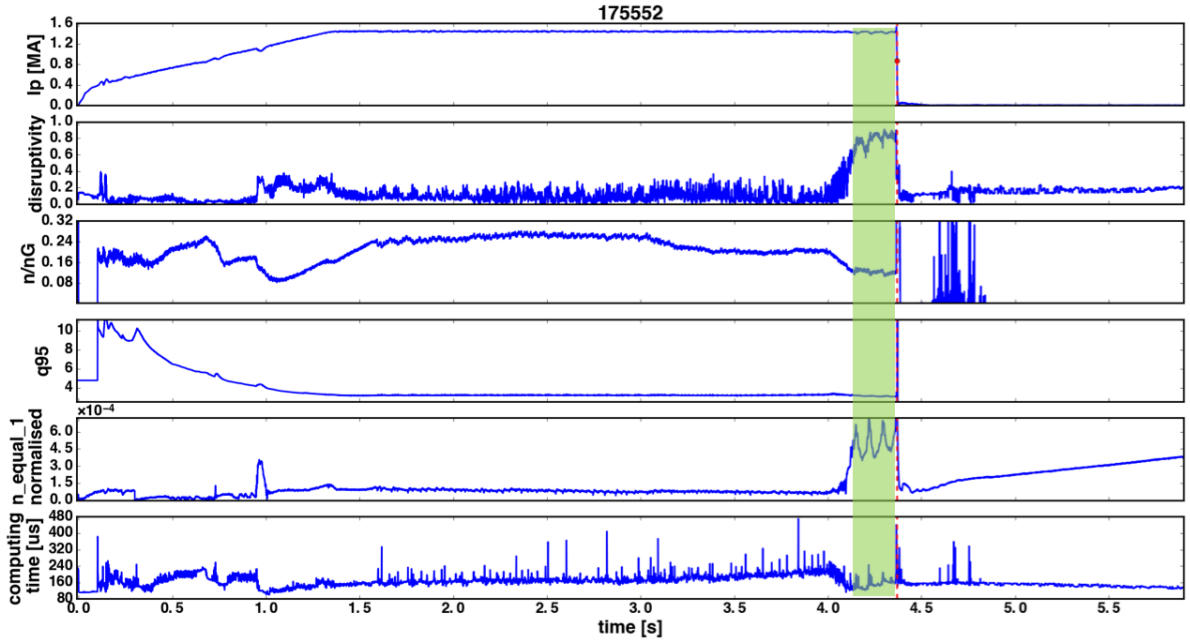

Figure 2: *Example of a DIII-D disruptive discharge, shot 175552. The red dashed vertical line represents the time of the disruption event. The disruptivity signal in the second panel rises from values of 10-20% to values greater than 60% with more than 150 ms warning time. The green box highlights the last 250 ms before the disruption: in Figure 3 we show the feature contributions to the predicted probabilities in that time frame.*

Feature Contribution. It is possible to compute the final DPRF disruptivity in terms of the average contribution of each feature[2].

[2] `treeinterpreter`, https://github.com/andosa/treeinterpreter, A. Saabas, unpublished.

For the last 250 ms of shot 175552 (where the disruptivity signal is greater than 60%), we report in Figure 3 the average contribution of each feature to the predicted probability, together with the final average disruptivity in that time window. *n_equal_1_normalised*, n/nG and $q95$ provide the major contributions to the predicted probability of an impending disruption.

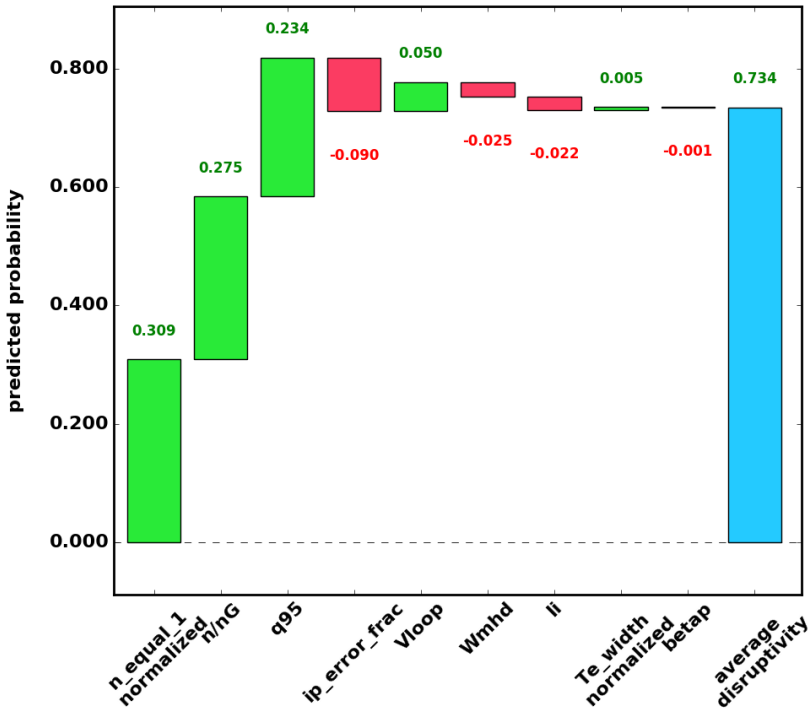

Figure 3: *Waterfall chart of feature contributions for shot 175552. The blue bar shows the average disruptivity in the considered time frame before the disruption occurrence, while positive contributions are represented by the green bars and negative ones are shown in red. The information contained in the Forest's decision paths defines the feature contributions to the final disruptivity.*

Conclusions. DPRF has proved to be very robust over almost four months of operation: less than 6% of non-disruptive discharges triggered a disruption alarm during the current flat-top. The ML-based algorithm was also successfully exploited during an ITER baseline scenario DIII-D discharge to change the plasma current ramp down rate and actively avoid an impending disruption on the basis of DPRF real-time disruptivity warning. By combining explainable decisions with real-time accurate predictions, novel disruption avoidance strategies can be provided.

Acknowledgments. This work was supported by the U.S. Department of Energy under DE-FC02-04ER54698, DE-SC0014264 and DE-AC02-09CH11466. DIII-D data shown in this paper can be obtained in digital format by following the links at https://fusion.gat.com/global/D3D_DMP. DISCLAIMER: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] C. Windsor et al., Nuclear Fusion **45**, 5 (2005)

[2] B. Cannas et al., Nuclear Fusion **47**, 11 (2007)

[3] J. Vega et al., Fusion Engineering and Design **88**, 6-8 (2013)

[4] S. Gerhardt et al., Nuclear Fusion **53**, 6 (2013)

[5] C. Rea and R.S. Granetz, Fusion Science and Technology, 1-12 (2018)

[6] C. Rea et al., Plasma Physics and Controlled Fusion **60**, 8 (2018)

[7] L. Breiman, Machine Learning **45**, 1 (2001)

[8] F. Pedregosa et al., Journal of Machine Learning Research **12** (2012)

[9] O. Meneghini et al., Nuclear Fusion **55**, 8 (2015)