PAPER

Machine learning for disruption warnings on Alcator C-Mod, DIII-D, and EAST

To cite this article: K.J. Montes *et al* 2019 *Nucl. Fusion* **59** 096015

Machine learning for disruption warning on Alcator C-Mod, DIII-D, and EAST

K J Montes[1], C Rea[1], R S Granetz[1], R A Tinguely[1], N Eidietis[2], O M Meneghini[2], D L Chen[3], B Shen[3], B J Xiao[3], K Erickson[4], M D Boyer[4]

[1] Massachusetts Institute of Technology, Plasma Science and Fusion Center, Cambridge, MA USA
[2] General Atomics, San Diego, CA USA
[3] Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, Anhui, China
[4] Princeton Plasma Physics Laboratory, Princeton, NJ, USA

E-mail: kmontes@mit.edu

January 2019

Abstract. This paper reports on disruption prediction using a shallow machine learning method known as Random Forests, trained on large databases containing only plasma parameters that are available in real-time on Alcator C-Mod, DIII-D, and EAST. The database for each tokamak contains parameters sampled $\sim 10^6$ times throughout $\sim 10^4$ discharges (disruptive and non-disruptive) over the last 4 years of operation. It is found that a number of parameters (e.g. P_{rad}/P_{input}, ℓ_i, n/n_G, $B_{n=1}/B_0$) exhibit changes in aggregate as a disruption is approached on one or more of these tokamaks. However, for each machine, the most useful parameters, as well as the details of their precursor behaviors, are markedly different. When the prediction problem is framed using a binary classification scheme to discriminate between time slices "close to disruption" and "far from disruption", it is found that the prediction algorithms differ substantially in performance among the three machines on a time slice-by-time slice basis, but have similar disruption detection rates (∼80-90%) on a shot-by-shot basis after appropriate optimisation. This could have important implications for disruption prediction and avoidance on ITER, for which development of a training database of disruptions may be infeasible. The algorithm's output is interpretable using a method that identifies the most strongly contributing input signals, which may have implications for avoiding disruptive scenarios. To further support its real-time capability, successful applications in inter-shot and real-time environments on EAST and DIII-D are also discussed.

1. Introduction

Application of artificial intelligence using machine learning (ML) methods for generating real-time warnings of impending disruptions in tokamaks is currently being developed [1–4] because approaches based on first-principles plasma physics may be too complex to be of practical use, particularly in real-time. Prediction algorithms like neural networks, support vector machines, and manifold learning techniques have been studied on JET [5–7] ASDEX-U [8, 9], and DIII-D [10] with similar approaches. However, these algorithms are often trained and tested on limited datasets from a single tokamak, and few cross-machine comparison studies have been implemented [11, 12]. The methods typically used in previous work have also generally relied on black-box ML algorithms, which have few methods available to interpret their predictions [13].

In an attempt to address the interpretability shortcoming, the ML work described in this paper uses a white-box supervised learning approach, which necessarily requires a large database for training and testing. However, on future high-power fusion reactors, the compilation of a large database of disruptions is problematic. If a universal ML algorithm that is proven to work on multiple present-day devices can be developed, it may resolve this conundrum. Hence, a multi-machine investigation as begun in this paper is needed.

With that in mind, we have developed databases of disruption-relevant parameters on a number of tokamaks, three of which are featured here, and used these to train similar ML prediction algorithms for the three machines. We note that we have restricted the parameter set to include only those signals which can be available in real-time in present-day tokamaks. Furthermore, we train and test our prediction algorithms on all discharges in the databases, without regard to any particular type of disruption.

In Section 2 we will describe the databases in some detail. Note that the databases contain information on both *offline* (i.e., minimally post-processed) data and the respective *real-time* counterpart (i.e., the data that the plasma control system has available in real-time). The former is used in Section 3-5 to discuss the offline developed methodology and presented in this manuscript, while the latter is adopted for all the real-time applications of the methodology. In particular, in Section 3 we compare and contrast the behavior of several plasma parameters amongst machines. Section 4 describes the development and optimisation of the Random Forests (RF) ML algorithms, and compares their disruption prediction performance on the three machines. In Section 5, we introduce a method for interpreting the output of our prediction algorithm as a sum of contributions from the individual input signals. In Section 6 we describe the implementation of a real-time RF-based predictor in the DIII-D plasma control system, and the between-shot testing of an RF algorithm on purposely-triggered VDE's on EAST. A summary and conclusions follow in Section 7.

2. The Databases Available on the Three Devices

In order to train and test disruption prediction algorithms on the three tokamaks, we have created similar disruption warning databases for Alcator C-Mod, DIII-D, and EAST by compiling values for a number of disruption-relevant parameters sampled at many times throughout all plasma discharges, disruptive and non-disruptive, from the 2014-2017 campaigns for DIII-D and EAST, and 2014-2016 campaigns for C-Mod. These databases are in the form of an SQL table for each machine, and can therefore be accessed by many commonly used scientific software packages (Matlab, IDL, Python, etcetera). Each record in the SQL database tables consists of a shot number, a time value, and the values of 50-60 disruption-relevant plasma parameters measured on the specified shot at the specified time. Records do not include information from previous time slices, since quantities reflecting average values or standard deviations over specified time windows are not used.

The choice of which parameters to include in the databases is guided by our knowledge of the plasma physics inherent in disruption phenomena. Many of the disruption-relevant parameters are based on the disruption detection study on NSTX-U previously published by Gerhardt [14], and include diagnostic measurements such as I_p error $[= I_p - I_p(programmed)]$, radiated power fraction $[= P_{rad}/P_{input}]$, the Greenwald density fraction $n/n_{Greenwald}$, Z_{error} $[= Z(centroid) - Z(programmed)]$, as well as a number of equilibrium parameters derived from EFIT reconstructions (q_{95}, ℓ_i, elongation, etcetera). Although not critical for this investigation, many of these physics parameters are normalised to machine size or B-field where appropriate in order to facilitate future multi-machine studies. It is important to note that the set of parameters we have chosen can, in principle, be available in real-time to a plasma control system (PCS). Therefore the algorithms we develop are suitable for use in real-time, running on the PCS (an example is mentioned in Section 6, and complete details are found in [15]). In order to keep the size of the databases to a manageable level while still capturing the desired evolution of parameters prior to a disruption, non-uniform time sampling has been used, with relatively moderate sampling rates throughout all discharges, plus higher sampling rates for a fixed period of time before each disruption. For Alcator C-Mod, sampling is done every 20 ms on all shots, which have a typical flattop duration of $\sim$ 1 s, and additional sampling is done every 1 ms during the 20 ms period before each disruption. For DIII-D, all shots ($\sim$ 3 s flattop) are sampled every 25 ms, and additional sampling is done every 2 ms for the 100 ms period before each disruption. For EAST, all shots ($\sim$ 6 s flattop) are sampled every 100 ms (some discharges are 100 s long), and additional sampling is done every 10 ms for the 250 ms period before each disruption. The choice of sampling rates and pre-disruption periods is based on a general survey of the disruption precursor timescales in each machine, done with the aim of capturing high frequency information relevant to the oncoming disruption. Data sampling rates can easily be adjusted if analysis of the database parameters indicates a need to do so.

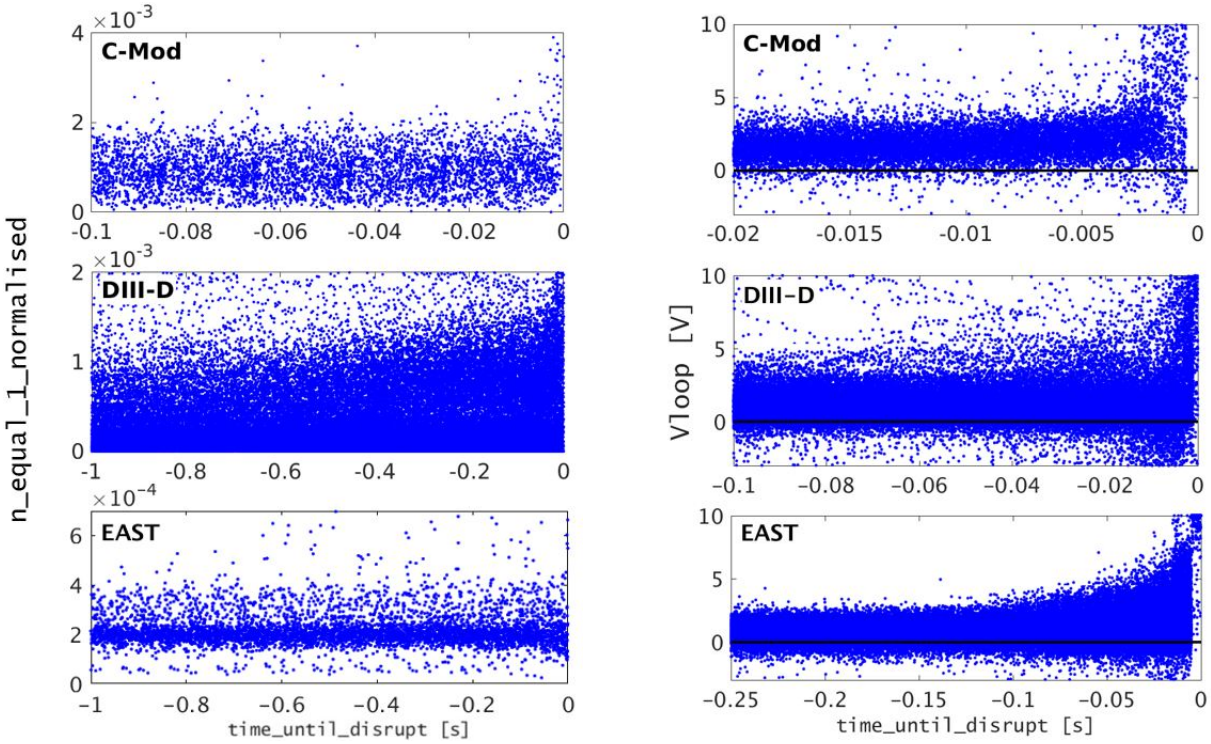
The disruption warning databases for C-Mod, DIII-D, and EAST contain parameter values for 0.5, 3.0, and 1.2 million time slices from more than 5000, 13000, and 14000 discharges respectively, and addition of parameters to the database is ongoing. Many of the plasma parameters are derived from EFIT [16] reconstructions. In order to avoid excessive interpolation we have run our own EFITs on all the discharges at the times we desire for our databases, using causal smoothing where needed. Avoiding non-causal filtering is absolutely necessary to ensure credible disruption prediction algorithms for real-time use.

3. Univariate Feature Analysis on C-Mod, DIII-D, and EAST

In the work described in this paper, we have concentrated on disruption prediction during the period of the plasma current flattop exclusively. This enables us to study disruptions that occur during steady-state operation, in a consistent heating and confinement regime. This narrow scope is justified by previous works showing differences in disruptivity during the ramp-down phase, and control room experience reflecting generally inadequate PCS control near the end of the discharge. Although disruptions certainly occur during rampup and rampdown, in ITER and future reactors (as well as many EAST discharges) the rampup and rampdown phases will be a negligible fraction of the discharge duration.

Through detailed examination of our databases for the three machines, we have found a number of plasma parameters that exhibit identifiable changes in behavior as disruptions are approached on one or more of these tokamaks, for a notable fraction of flattop disruptions. Examples include radiated power fraction (P_{rad}/P_{input}), internal inductance ℓ_i (current profile peakedness), Greenwald fraction (n/n_G), $n = 1$ locked mode indicator, T_e profile width, and a number of other commonly measured plasma parameters. However, each individual parameter behaves markedly different on each machine. These different behaviors are a reflection of the fact that the different machines do not have identical operational spaces and therefore do not have the exact same set of disruption types. Illustrative examples are shown in Figure 1.

In Figure 1(a) the evolution of `n_equal_1_normalised` is shown as a flattop disruption is approached, for thousands of disruptions on each machine. This parameter is a proxy for the $n = 1$ toroidal Fourier harmonic of the perturbed magnetic field of non-rotating modes, and is then normalised to the machine's toroidal magnetic field. Each machine has different hardware configuration for the magnetic sensors; we refer to [17] for a detailed discussion on how `n_equal_1_normalised` is computed on C-Mod and DIII-D, given the different sensor hardware available. For what regards EAST, `n_equal_1_normalised` is calculated from the saddle loop signals, after compensating for the pickup from resonant magnetic perturbation (RMP) coils.



(a) (b)

Figure 1: Behavior of the $n = 1$ locked mode proxy (a) and loop voltage (b) is markedly different on the three tokamaks. Disruptions time is at $t = 0$ s on the right edge of each graph. Note the different time scales and vertical scales for each machine.

The first panel of Figure 1(a) shows that, although `n_equal_1_normalised` tends to increase on a notable fraction of C-Mod disruptions, it does not do so until just a few ms before the disruption time, which is too short to be of practical use. On DIII-D, `n_equal_1_normalised` tends to increase slowly before disruptions, starting roughly a half-second before the disruption time. And on EAST, `n_equal_1_normalised` does not show any change of behavior as disruptions are approached.

Another example is given in Figure 1(b), showing the loop voltage on each machine as flattop disruptions are approached. On EAST, a large fraction of disruptions are preceded by an increase in loop voltage, starting about 100 ms before the disruption time. This behavior is less pronounced and there is much less warning time on C-Mod and DIII-D. Similar contrasting behavior between machines is also seen for the radiated power fraction, the normalised I_p error, and others. The fact that the same plasma parameters generally show markedly different evolution leading up to disruptions could complicate the successful development of a universal disruption predictor.

Table 1: Input signals used for developing DPRF on each tokamak.

Variable name	Signal description
`n_equal_1_normalized` [a]	Perturbed field of nonrotating modes ($n = 1$ Fourier component), normalised to toroidal magnetic field $B^{n=1}/B_{tor}$
`q95`	Safety factor at the 95% flux surface
`Greenwald_fraction`	Greenwald density fraction n/n_G
`ip_error_frac`	Fractional error between measured and programmed plasma current
`li`	Normalized internal inductance
`betap`	Poloidal beta
`Vloop`	Loop voltage V_{loop} [V]
`Wmhd`	Stored plasma energy [J]
`Te_width_normalized` [b]	Width of quadratic approximation to electron temperature profile, normalized to plasma minor radius
`radiated_fraction`	Total radiated power divided by total input power

[a]For the EAST DPRF, the non-normalized n=1 Fourier amplitude is used as an input because the toroidal B-field measurement is unavailable for a significant number of discharges.
[b]For the C-Mod DPRF, the T_e profile width is excluded from the list of input features because this data is missing from a significant number of discharges.

4. DPRF Development and its Performances on the Different Devices

The Machine Learning model we adopted to develop our disruption predictor on the three different tokamaks is based on the Random Forests algorithm; we will refer to it using the abbreviation DPRF, Disruption Predictor using Random Forests. The methodological details of the Random Forests algorithm can be found in the original paper from Breiman [18] and in previous publications from the authors [17, 19].
RF is a supervised algorithm, meaning that class labels need to be assigned to each sample in the available datasets, via human supervision. If the assigned class labels are discrete, then the algorithm is defined as a classifier, whereas if the class labels are continuous the algorithm is referred to as a regressor. In particular, DPRF is a supervised classification algorithm, where the class labels are assigned depending on a threshold in time, specific for each device, chosen on the basis of the univariate analysis on the aforementioned plasma signals (see Section 3). The assigned class labels are discrete and binary: the data sample belongs either to a class labeled "close to a disruption" or to a class labeled "far from a disruption". This classification implicitly assumes that it is possible to detect a transition in time from a safe operational regime to a disruptive one and is another instance of incorporating physics knowledge into the AI workflow.

RF are ensemble learners: the algorithm learns by developing a large collection of independent, de-correlated predictors (i.e., the individual decision trees in the forest). Each tree is a hierarchical data structure created by recursively partitioning the dataset available [20]. Ideal partitions are created by probing the input feature space, given by the plasma signals from Table 1. They are chosen to obtain the largest information

Table 2: Number of discharges included in the datasets for each machine.

	C-Mod		DIII-D		EAST	
	Disruptive	Non-Disruptive	Disruptive	Non-Disruptive	Disruptive	Non-Disruptive
Train	532	2886	867	5074	1689	4738
Test	134	722	217	1269	423	1185

gain by minimizing an impurity measure, i.e. the Gini index [18], that measures the classification error associated with each pair of (feature, value) tested. Each statistical test is done using the features' real values - no feature scaling or normalisation is actually required. Starting from a root node (the initial decision, i.e. a statistical test on one randomly chosen input feature), the decision paths are obtained by learning on training subsets, obtained via a random sampling with replacement from the original dataset (i.e. bootstrapped samples). The prediction on which class label to assign is provided individually by each tree for a particular feature vector sample, and the final forest prediction is obtained via aggregation, using majority voting.

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. For further reading on Random Forests applications, please refer to [17, 19].

Our choice of parameters to include in these applications is based partly on our own tokamak operational experience and partly on those specified in the relevant literature [12, 14, 21, 22]. All the signals reported in Table 1 represent relevant physics triggers to disruption events, such as low-density or high radiated power disruptions or locked mode-driven ones.

A strong assumption in the development of DPRF is the selection of only the flattop portion of the discharges to train DPRF; therefore the plasma current flattop phase represents the validity range for any performance metrics, as well as for the classifier's predictions. This also implies that the focus of this predictive algorithm are disruptions happening during the flattop, regardless of the particular chain of events, and not rampup or rampdown ones (even though such data are available in the SQL databases). Still with these restrictions, our dataset is comprised of a large number of both disruptive and non-disruptive discharges, as shown in Table 2.

4.1. Time slice performances

Before discussing the details of a shot-by-shot analysis, we report DPRF performances in terms of a confusion matrix for each device (Figure 2) as a benchmark for comparison.

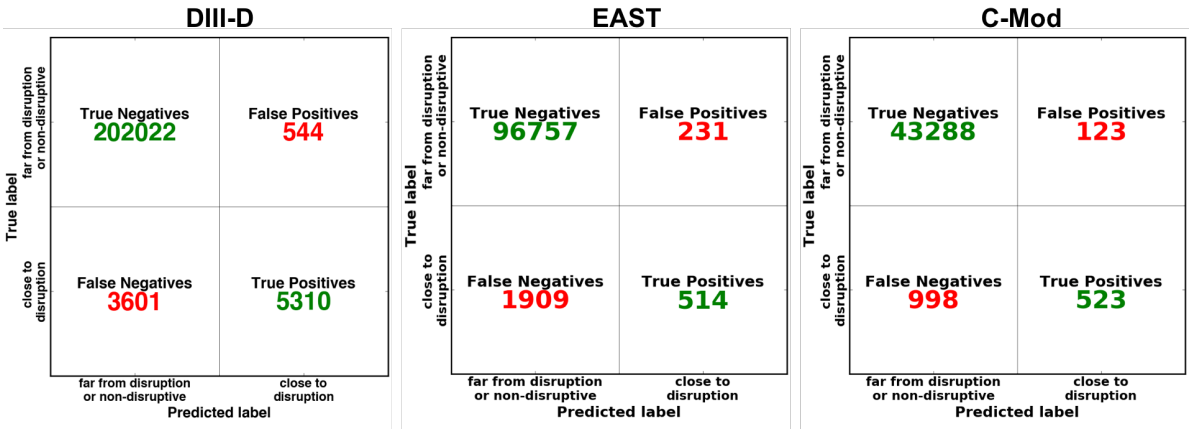


Figure 2: DPRF performances on a time slice basis are summarised in a confusion matrix for each tokamak. The positive class refers to a time slice with an assigned "close to disruption" label, while the negative class refers to a "far from disruption" time slice.

DPRF was trained using a different threshold for the class label separation, τ_{class}, on each machine: for DIII-D, disruptive time slices are labelled starting from 350 ms before the disruption event; on EAST the discrimination threshold is set at 100 ms; while on C-Mod, a 40 ms threshold is chosen. The class label separation times on DIII-D and EAST were chosen from observation of signal temporal behavior as described in Section 3. In contrast, the threshold time for C-Mod was set at a minimum value that is practically useful for disruption warning purposes, since a proper threshold choice was not made clear by a similar univariate analysis.

The performances reported in Figure 2 are obtained using the aforementioned thresholds to discriminate between the disruptive label (i.e., the positive class) and the non-disruptive one (i.e., the negative class). The fraction of correctly predicted disruption samples varies considerably and is far from perfect, ranging from $\sim 60\%$ for DIII-D, to just $\sim 22\%$ for EAST. It is important to note that these performance metrics are very different when evaluated on a shot-by-shot basis, as described next.

4.2. Optimised mapping of time slice predictions to shot predictions

Signal measurements invariably have some noise, and ML algorithms are not perfect, so it is not necessarily wise to declare that a disruption is imminent based solely on the RF output, or *disruptivity* value, for a single time slice. In order to provide an accurate warning of an impending disruption, we desire to evaluate the performance of DPRF on a shot-by-shot basis, for which the temporal distribution of time-sample predictions is taken into account. We do this by using a hysteresis threshold system in the following manner: if the disruptivity remains above a *low threshold* for a certain time interval (the *alarm window*) after having exceeded a *high threshold*, the warning alarm is trig-

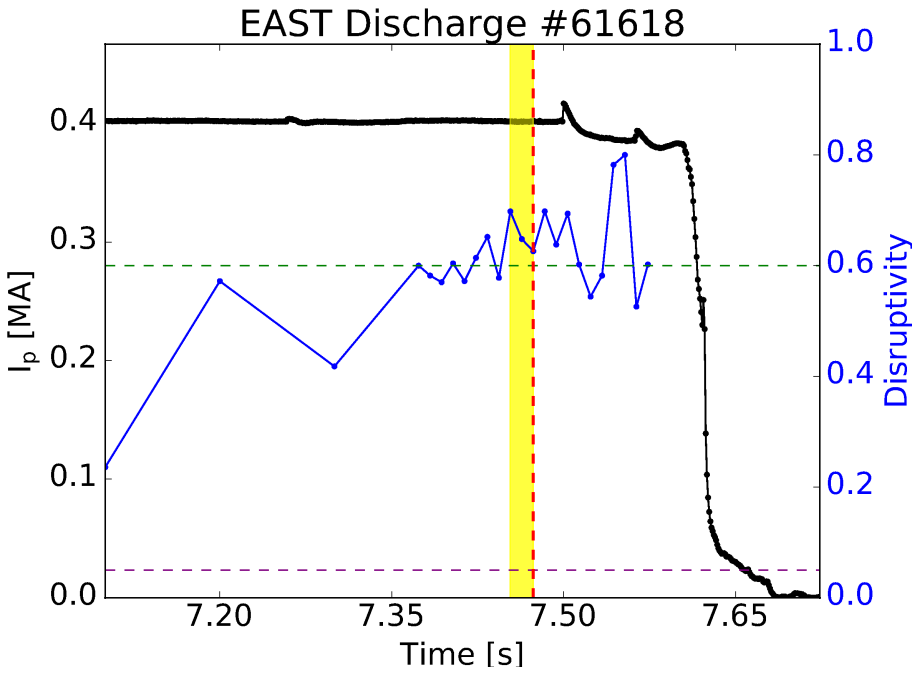


Figure 3: A disruptive shot on EAST warned by a DPRF algorithm with a 0.6 high threshold (green), 0.05 low threshold (purple), and 20 ms alarm window (yellow); the alarm trigger time (red) comes $\sim$ 30 ms before the first current spike

gered. An example of a successfully warned disruption using this scheme is shown in Figure 3. Note that the low threshold and alarm window together may act to make the trigger algorithm robust to a noisy disruptivity signal, which is important for real-time application.

With a defined alarm trigger method, we can now extend the continuous time slice predictions to binary shot-by-shot predictions. Disruptive shots (as opposed to time slices) are true positives (TP) if the alarm is triggered before the disruption time, and false negatives (FN) otherwise. Non-disruptive shots that trigger the alarm are false positives (FP), and non-disruptive shots without an alarm trigger are true negatives (TN). Since the alarm trigger is a function of the operational parameters (i.e. the chosen disruptivity thresholds, alarm window, and τ_{class}), the number of shots in each category will vary with the operational point.

An ideal disruption warning algorithm will operate with a high precision $[TP/(TP+FP)]$ and high recall $[TP/(TP+FN)]$, since it will trigger few false alarms on healthy plasma discharges and rarely fail to trigger discharges that disrupt. Therefore, the optimum operational point can be chosen by maximizing a performance metric which accounts for both precision and recall during the algorithm's training and validation process, before its ability to generalise to unseen data is analyzed during the testing

process. To this end, we have adopted a binary classification metric called the F_γ-score, given by

$$F_\gamma = (1+\gamma^2)\frac{precision \cdot recall}{(\gamma^2 \cdot precision) + recall} \tag{1}$$

where γ can be chosen based on operational needs. For example, when $\gamma = 1$ the precision and recall are equally weighted, but higher values associate a higher cost with missed warnings (since $F_\gamma \to recall$ as $\gamma \to \infty$). One note of caution is warranted here: framing the optimisation this way will reward early triggers on disruptive discharges that may precede any causal events related to the eventual disruption. Although this was not addressed in the optimisation workflow, a post hoc analysis is included in Section 4.3 to discuss the prevalence of these early warnings in the dataset.

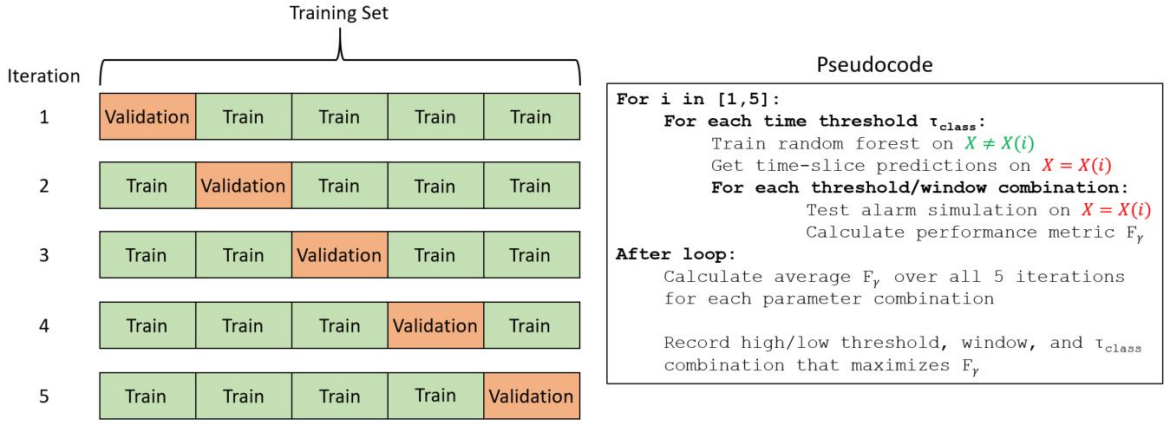


Figure 4: Shematic and pseudocode for the K-fold cross validation procedure used to optimise the DPRF operational parameters. Training set data X is randomly subdivided into K=5 subsets, with time samples of the same shot grouped together. Pseudocode shows serial analog of the parallelised validation routine developed in OMFIT [23]

To calculate the F_γ-optimised operational points robustly, a K-fold cross-validation procedure was used (see Figure 4). This was most easily done using a parallelised grid search, since each operational point requires K Random Forests to be trained and tested. The grid mesh included high and low threshold values from 0.05 to 0.95 in steps of 0.05, alarm windows from 5 ms to 405 ms in steps of 25 ms, and class values from 25 ms to 1000 ms in steps of 25 ms. Upon splitting the training set into $K = 5$ subsets, RF were trained on each combination of $K-1$ subsets and validated against the corresponding held out subset for each point in the operational grid. We then calculate the mean F_γ-score of the K training splits at each operational point and record the τ_{class}, high threshold, low threshold, and alarm window combination that corresponds to the maximum F_γ value. These optimised parameters are then used to train a DPRF model on the entire training set and apply it on the unseen test set.

Table 3: DPRF operational points optimised on the training set for each tokamak using the F_1 and F_2 scores as performance metrics; the corresponding fractions of non-disruptions (FP) and disruptions (TP) in the test set for which the alarm was triggered are included for each optimised model.

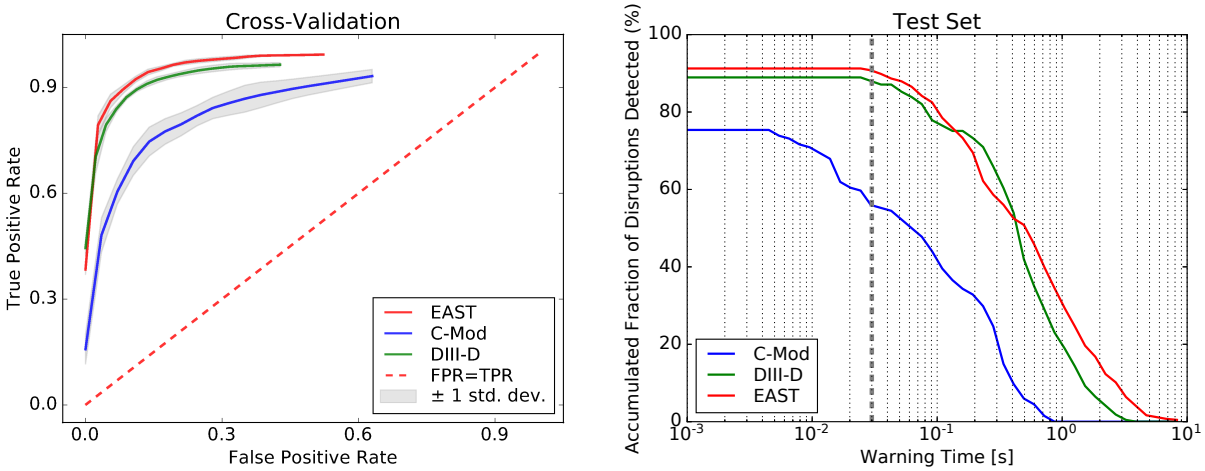
	F_1-Optimised				F_2-Optimised			
Tokamak	τ_{class}(ms)	High Threshold	FP(%)	TP(%)	τ_{class}(ms)	High Threshold	FP(%)	TP(%)
C-Mod	250	0.50	7.2	61.2	325	0.35	19.3	75.4
DIII-D	700	0.60	2.7	79.3	875	0.40	8.5	88.9
EAST	875	0.70	5.1	81.6	950	0.50	13.2	91.3

4.3. Shot-by-shot performance

In order to see how the performance varies with the metric chosen, we implemented the optimisation procedure described above for $\gamma = 1, 2$. The cross-validation results reveal that smaller alarm windows tend to have higher performance metrics: the F_γ-optimised point for each machine is found at the smallest window size, below the regular sampling period in our databases. This indicates that an ideal alarm trigger should be highly sensitive to increases in disruptivity: soon after the output exceeds the high threshold, a warning should be given. Similarly, the optimum low disruptivity threshold is found at the lower extreme of the range (0.05). At this value, the threshold does little to prevent largely scattered disruptivity signals near the high threshold from triggering an alarm, but still counteracts false alarms by allowing quick spikes in the disruptivity signal. The optimised values for both of these parameters are the same for each machine. Therefore, we only included the optimal τ_{class} and high threshold values for each machine in Table 3, along with the fraction of disruptions and non-disruptions in the test set for which an alarm is triggered. Note that the models optimised with the F_1 score have relatively low false alarm rates, yet the F_2-optimised models warn a significantly larger fraction of disruptions. Prioritizing disruption avoidance, we will focus only on the models optimised using the F_2 score from this point forward.

Two other immediate observations may be made from the F_2 results in Table 3. First, note that the optimised τ_{class} values on DIII-D and EAST are much larger than that on C-Mod. This is consistent with the ordering of the times of the distribution shifts of parameters like `n_equal_1_normalised` and `Vloop` (see Figure 1) found via univariate analysis, which show that the dynamics on C-Mod tend to evolve on a faster timescale. Secondly, since the disruptivity threshold for the alarm trigger varies for each alarm algorithm and each tokamak, we also see that the RF output should not be thought of as an injective mapping to disruption probability. Rather, the model must be calibrated separately for each machine in order to improve the predictive capability and assess its optimised performances. This also offers a partial explanation for the

poor time slice predictive capability in the binary-classification problem of Section 4.1, where only disruptivity outputs above the default value of 0.5 were considered a positive class prediction. Performance is also improved when the τ_{class} threshold is moved further back in time from the values motivated in Section 3 for each machine. This hints that the RF is detecting input signal behavior correlated with disruptions that is hidden from the bulk univariate analysis first appealed to in Section 3, which relied only on global changes in disruption-relevant signals and did not take sequential information into account as done in the shot-by-shot analysis.

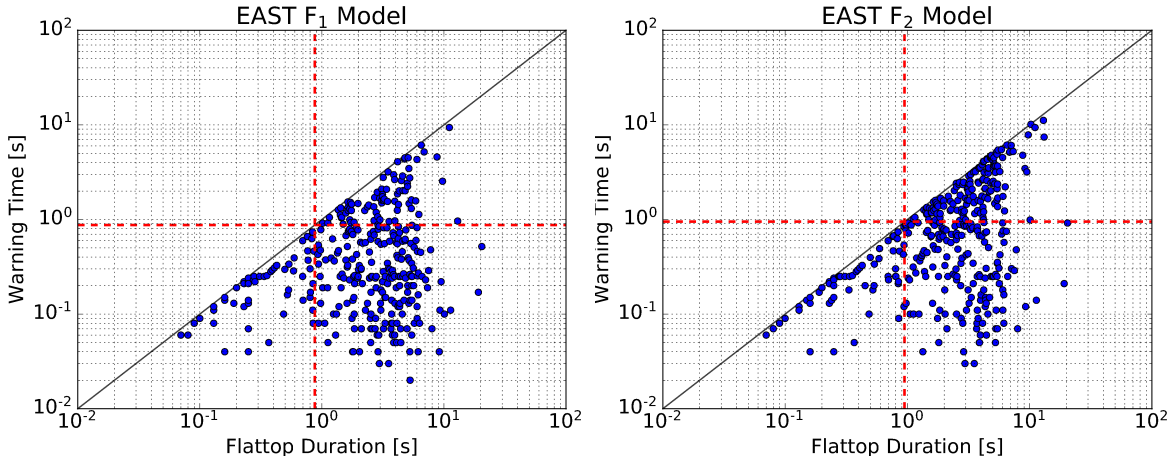


(a) (b)

Figure 5: (a) Average true and false positive rates amongst K-fold validation sets for varying high disruptivity thresholds (Section 4.2) on each tokamak, with other 3 operational parameters fixed at F_2-optimised values; (b) Cumulative warning time distributions associated with the F_2-optimised model performance on the test set for each tokamak; approximate time needed for mitigation (30 ms) is highlighted with a gray dashed line.

In addition to analyzing test set performance, one can examine the operational space from the validation process to study the sensitivity of expected performance to changes in model parameters. An example is shown in Figure 5(a), generated by varying the high disruptivity threshold along the grid mesh while keeping the other 3 operational parameters fixed at their F_2-optimised values. The figure shows the fraction of triggered disruptions and non-disruptions for each threshold, where the threshold increases along the curve from left to right. Again, the performance gap between C-Mod and the other machines is pronounced. One can also see from the standard deviation amongst the K folds that the C-Mod performance varies much more with the random splitting in the dataset. Ultimately, this information can be used to perform a cost-benefit analysis and tune the model from the F_γ-optimised operational point to achieve different objectives.

Since disruption predictions are less useful the later they appear, one also needs information about the warning time, or the time between the alarm trigger and the disruption event. A cumulative distribution of warning times for each machine is shown in Figure 5(b), where the fraction warned at time T represents the fraction of all disruptions in the test set warned at least time T in advance. Note that the majority of disruptions on each machine are warned greater than 30 ms in advance, which is on the order of the time needed for mitigation [24]. However, more disruptions are warned farther in advance at a much lower cost of false alarms on both EAST and DIII-D when compared with C-Mod. This is consistent with the univariate analysis in Section 3, which pointed out the short timescales of disruptive behavior in aggregate on C-Mod relative to those of similar behavior seen on EAST and DIII-D.



(a) (b)

Figure 6: Warning times and flattop durations for triggered disruptive discharges in the EAST test set using the DPRF algorithm optimised with the F_1 (a) and F_2 (b) scores; each blue marker represents an individual discharge, and each time is measured in reference to the time of disruption at $t = 0$. Note that the bulk of the warnings occur near or below the τ_{class} threshold corresponding to each model (red dashed line).

At the tails of each distribution, one can also see evidence of the early warning problem alluded to in Section 4.2. Over 30% of disruptions on EAST have a warning time greater than 1 s, and its distribution tail is the longest of the three machines. The extent of this problem may be explored by comparing the flattop durations for each shot with the warning time, as shown in Figure 6. Note that a fraction of the warned disruptions are triggered immediately after the start of the I_p flattop (discharges near the black line in Figure 6), a phenomenon also seen in the DIII-D and C-Mod datasets. The fraction of warnings in this category increases from the F_1 to the F_2 optimised model, as avoiding false positives becomes a lesser priority. This may be seen by comparing the

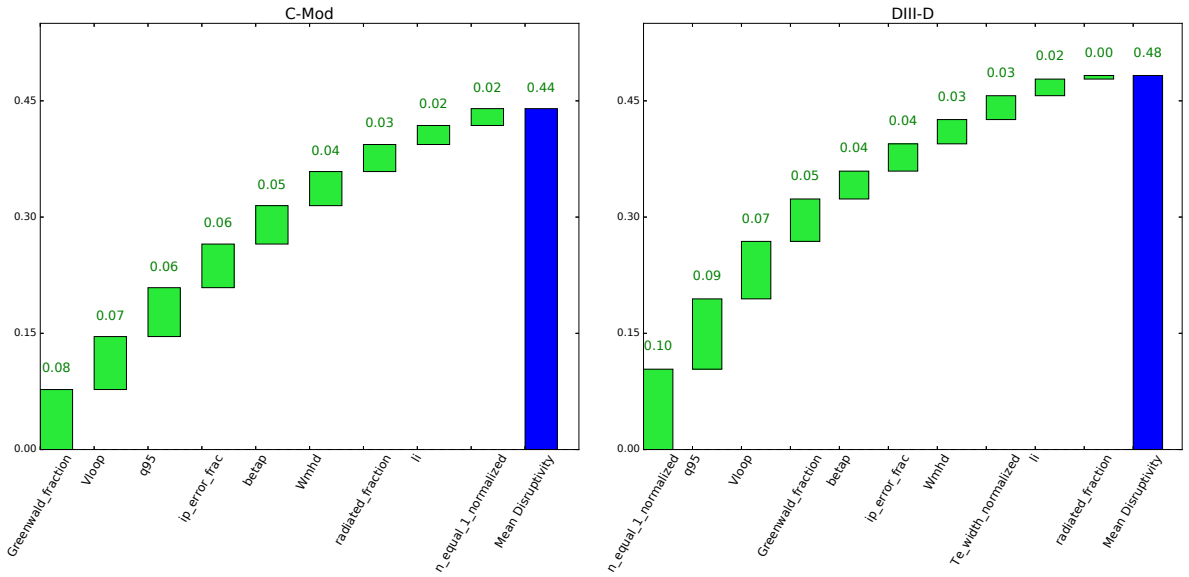
densities in the right hand quadrants of Figure 6. This phenomenon may be attributed to a rising disruptivity behavior seen on many shots during the ramp-up phase, shown for example in Figure **??**. This behavior often does not fully subside by the beginning of the flattop phase, and therefore may trigger early warnings that are not correlated with the disruption. This may provide motivation to further constrain the algorithm's region of validity for training and testing in the flattop phase. One may also see in Figure 6 that a significant fraction of the shots in the EAST dataset are located in the lower left-hand quadrant, indicating that they have a flattop phase that is shorter than the F_2-optimised $\tau_{class} = 950\ ms$ threshold for binary classification of time slices. Further work is needed to better account for these shots in the classification scheme and analyze the dynamics that are driving the early triggers on disruptive discharges.

5. Interpretation of Predictions via Feature Contribution Analysis

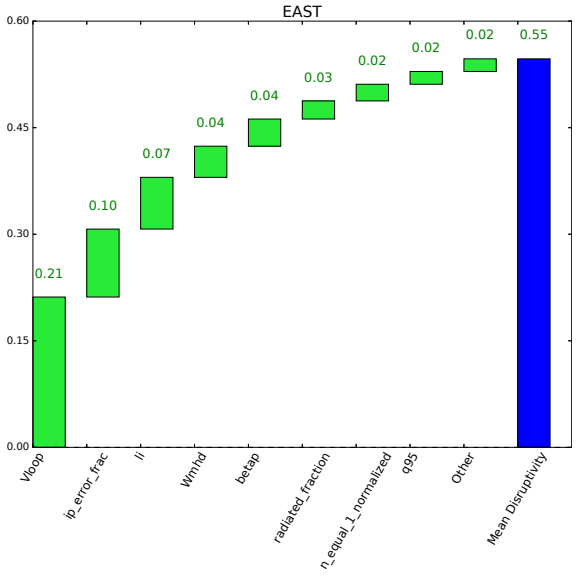
Being a resourceful machine learning algorithm, Random Forests are characterised by many *white-box* features; RF provide not only information on the training set via importance ranking for its input features [18], but rich content is stored in the decision paths of each forest tree, developed during the training process. This approach is adopted in many fields [25] to interpret the forest predictions. It is outside the scope of this paper to provide an in-depth description of this methodology, called feature contribution analyisis [26]; we refer to [15] for more details and examples.

We use this prediction interpretation method to identify the signals driving the DPRF output on any given individual discharge, which can then be used to gain an understanding of disruptive behavior in aggregate on each tokamak. It suffices to say that the feature contribution analysis involves a linear decomposition of each predicted value into the contributions coming from each of the input features. These are indeed constrained to assume positive or negative values that add together algebraically to give the RF output value for each time slice (or evaluated feature vector). In our application, a negative feature contribution value indicates that the feature's real value pushes the model towards a feature space that defines the far from disruption or non-disruptive class.

Figure 7 shows the average feature contributions and disruptivity values at the trigger time, collected for each disruptive discharge in the test sets used for each different tokamak. The waterfall charts give an idea of the strongest drivers of disruptive predictions in each dataset, revealing that the highest contributing parameters are markedly different on each machine. Note, for example, that `n_equal_1_normalised` and `q95` are the top contributing features on DIII-D, which is known to have a large fraction of locked-mode and MHD-driven disruptions, whereas these parameters are the least contributing on EAST. On EAST, we see that `V_loop` is a major contributor to disruption predictions, which is correlated with the aggregate increase in this parameter



(a) (b)



(c)

Figure 7: Mean feature contributions recorded at the trigger times of disruptive shots in the test datasets for C-Mod (a), DIII-D (b), and EAST (c); each set of mean contributions sums to the mean disruptivity at the trigger time, which is greater than the corresponding high threshold for the F_2-optimised model in Table 3 [27].

identified in Figure 1. These two machines contrast again with C-Mod, which shows that several parameters contribute relatively equally to the disruption predictions in the test set, indicating that there may be a wider variety of disruptions on C-Mod.

6. Real-Time Machine Learning-Based Algorithms on DIII-D and EAST

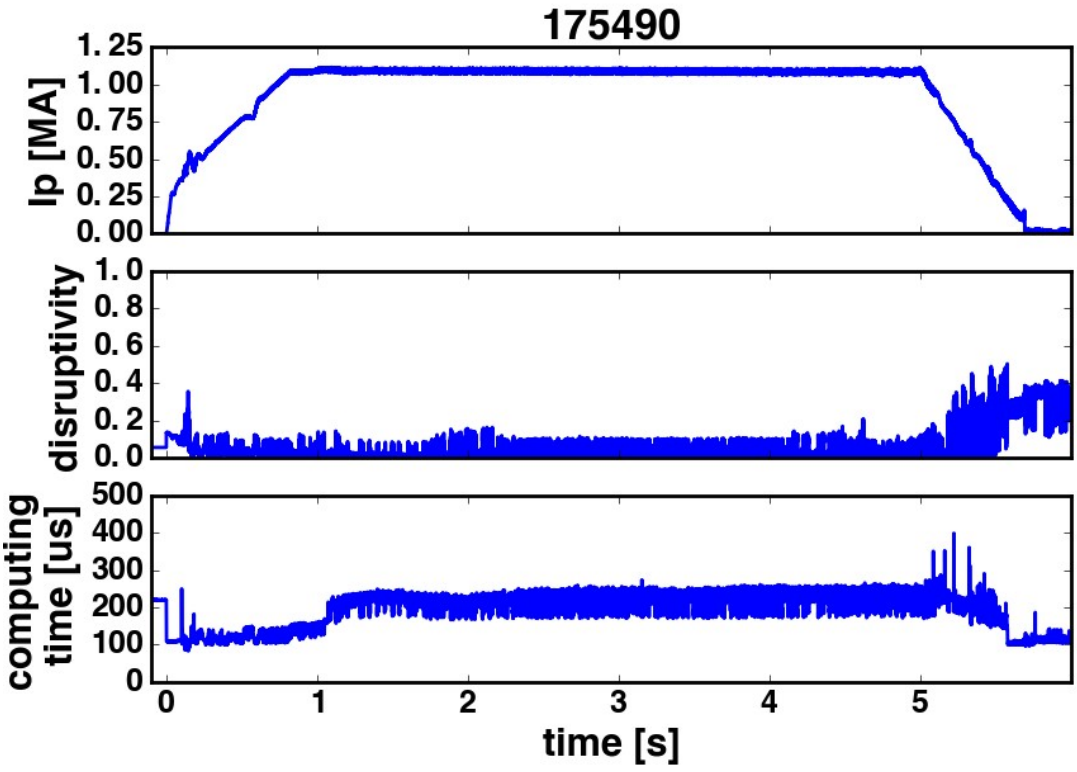


Figure 8: Example of non-disruptive discharge on DIII-D, for which DPRF ran its real-time calculations. The plasma current is reported in the upper panel, in MA. The disruptivity is shown in the central panel and ranges around 15% throughout the flattop. Most shots in our train and test datasets exhibit this kind of quiescent behavior, where the disruptivity does not rise high enough to trigger a disruption warning during the flattop phase of the discharge. In the bottom panel we report the computing time of the CPU that ran the DPRF algorithm.

6.1. The Real-Time Application on DIII-D [15]

A DPRF routine to run in the DIII-D PCS in real-time was developed by training on the same data that is furnished to the PCS in real-time, including quantities from real-time EFITs. DPRF has continuously run in the DIII-D PCS for more than 4 months

of operations, gathering data on more than 900 discharges, 66% of which were non-disruptive, 6% disrupted during the flattop, and the remaining 28% disrupted during rampup or rampdown. The training set is again limited only to the flattop portion of thousands of discharges (both disruptive and non-disruptive), spanning many years of DIII-D experiments.

In this section, we report only an example of a non-disruptive discharge, shown in Figure 8. The plasma current is shown at the very top, the disruptivity predictions are reported in the central panel of the figure, while the computing time for DPRF inference is reported in the bottom panel. It is possible to see that the CPU computing time ranges around 250-300 μs, which is definitely compatible with real-time requirements.

For more detailed examples and a full discussion on DPRF performances during 2018 campaign, we refer to [15].

6.2. EAST VDE Experiments

The DPRF algorithm was also tested on EAST between shots during experiments performed to purposely trigger Vertical Displacement Events (VDEs). For these particular experiments, EAST DPRF was trained on 7257 discharges, 5330 of which were non-disruptive ones. Furthermore, given the experimental target, three additional input signals were included, apart from those already mentioned in Table 1: the elongation, and the current centroid information (Z), plus the error between the programmed current centroid position and the actual reconstructed one (δZ).

A representative discharge is shown in Figure 9: in the first panel, the plasma current (black) and the disruptivity (blue; causally smoothed using a convolution with a Gaussian filter and a 10 ms window) are reported, while in the bottom panel it is possible to see the contributions from the 13 input features. Only the three most relevant contributions are shown in color. As explained in Section 5, the sum total of these feature contributions yields the disruptivity value at any given time. From the bottom panel of Figure 9 it is possible to see that the disruptivity signal is strongly affected by the elongation and the current centroid signals, reflecting the actual changes in the physics of the discharge.

7. Summary and Conclusions

In this paper, we have presented a methodology to fine-tune DPRF, a Random Forest algorithm for disruption prediction, that is separately optimized for three different tokamaks. This gives a basis for consistent comparison of prediction performance across multiple devices, a necessary tool for development of a universal disruption predictor. This is distinct from other adaptive algorithms trained *from scratch* [4, 5, 28] that have

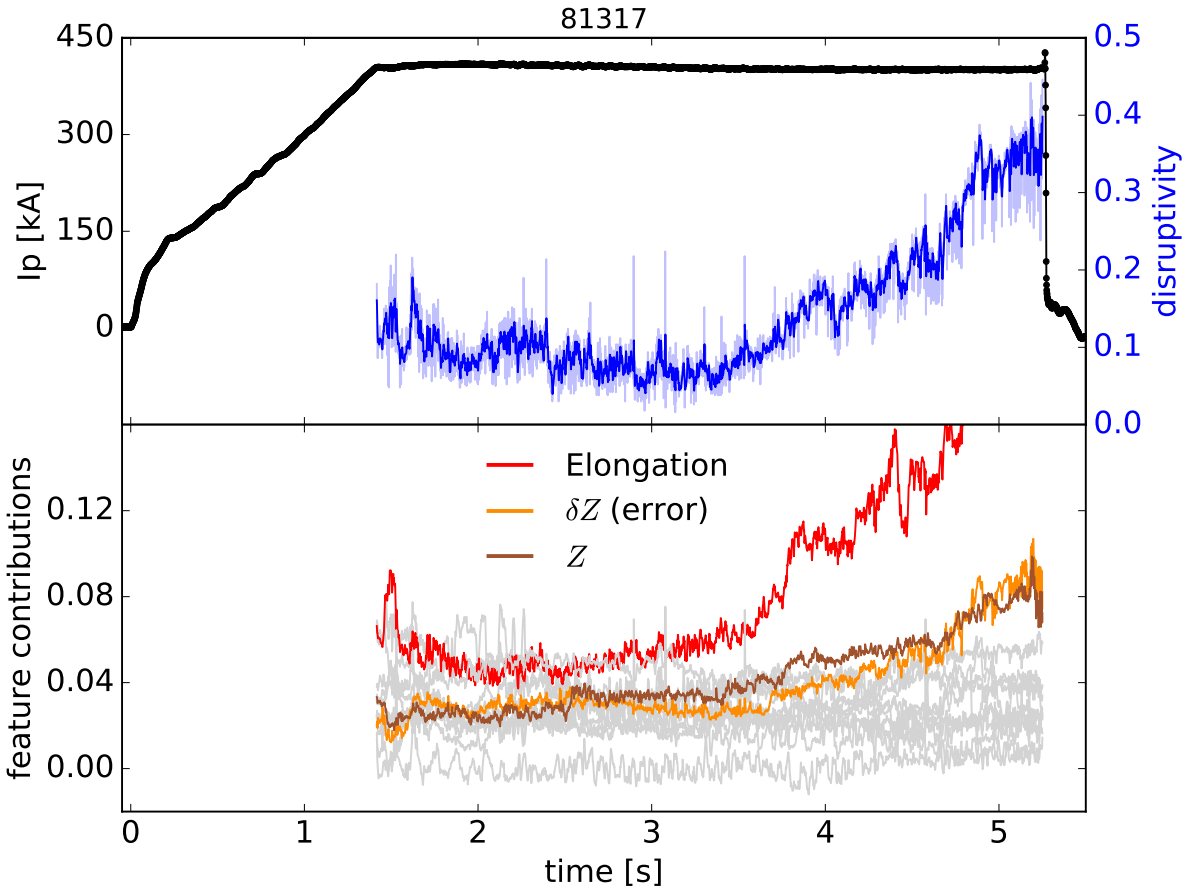


Figure 9: EAST DPRF disruptivity prediction for discharge 81317. In the first panel, the plasma current (black) and the disruptivity (blue) are reported. The disruptivity is causally smoothed with a 10 ms window and shown only during the flattop phase of the plasma current. The second panel shows the breakdown of disruptivity in terms of its 13 feature contributions, only the three most highly contributing features are shown in color. It is seen that the predictor determined that elongation and current centroid information reflect changes in the physics evolution prior to the VDE, even though these types of VDEs were not tailored when training the algorithm.

achieved very good performances with databases of more limited size or curated to exclude specific disruption dynamics [8].

On an individual time slice basis, the prediction accuracies vary considerably, with the true positive rate on C-Mod and EAST being particularly low, i.e. many missed disruption time slices. However, we find that optimised predictors do much better on a shot-by-shot basis on all three machines ($\sim$ 80-90% success rates for each machine), an encouraging result that we attribute to the extremely low rate of false positive time slices. This could mean that simultaneously running a suite of predictors, each trained on a different type of disruption, or a different region of operational space, may be a way

to realise machine-independent disruption prediction. Noting from Table 3 that these true positive rates come at a cost of false alarms $\sim 10\%$, further improvement of DPRF performance requires an elimination of false alarms. Investigation of the early warning behavior discussed at the end of Section 4.3 may allow a substantial improvement. The cause of this behavior should be identified and isolated by the addition of one or multiple features correlating with DPRF early warnings, or further curation of the region of validity for training.

DPRF provides a predictive output correlated with the onset of disruptions, i.e. a disruptivity signal, now incorporated in the DIII-D PCS. Thanks to the white box features of Random Forests, DPRF also provides a way to interpret the prediction (e.g. which signals contributed the most to triggering an alarm). By identifying the causes underlying the disruption events, a better understanding of disruption dynamics can be achieved, and the most appropriate actuators can be identified in the future for possibly avoiding impending disruptions. We find that the most important disruption-relevant physics parameters on C-Mod, DIII-D, and EAST are different on each machine, which likely reflects the fact that their operational spaces are not identical, and that different types of disruptions are more prevalent on each machine.

Work in the near future will include a study of the attributes of early warnings in each dataset and how these affect the optimisation procedure. This problem will likely be most fruitful to study on EAST due to the long flattop duration for most shots on this device. To advance understanding of how disruption dynamics scale from machine to machine, future work must also involve studies in domain adaptation. This includes training an algorithm on data from one tokamak or physics regime and testing it on another, as well as training an algorithm using data from all machines together. This effort can be further advanced by populating the database with additional dimensionless parameters that are both relevant to disruptions and available in real-time to the plasma control system.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Awards DE-FC02-99ER54512, DE-SC0014264, DE-SC0010720, DE-SC0010492, DE-FC02-04ER54698. Additionally, this work is supported by the National MCF Energy R&D Program of China, Grant No. 2018YFE0302100. Part of the data analysis was performed using the OMFIT integrated modeling framework [23]. 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

References

[1] Murari A, Vega J, Rattá G, Vagliasindi G, Johnson M and Hong S 2009 *Nuclear Fusion* **49** 055028 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/49/i=5/a=055028?key=crossref.3f9cea5ab70840b502a07d305d450b7f

[2] Zheng W, Hu F, Zhang M, Chen Z, Zhao X, Wang X, Shi P, Zhang X, Zhang X, Zhou Y, Wei Y and Pan Y 2018 *Nuclear Fusion* **58** 056016 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/58/i=5/a=056016?key=crossref.573f7d15f6f06cbe6ee98536b02bcefb

[3] Rattá G, Vega J, Murari A, Vagliasindi G, Johnson M and de Vries P 2010 *Nuclear Fusion* **50** 025005 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/50/i=2/a=025005?key=crossref.6240d82066621bd84181169f23e7fdb9

[4] Dormido-Canto S, Vega J, Ramírez J, Murari A, Moreno R, López J and Pereira A 2013 *Nuclear Fusion* **53** 113001 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/53/i=11/a=113001?key=crossref.6543d1976a1debf63c3b0a331717c4d4

[5] Murari A, Lungaroni M, Peluso E, Gaudio P, Vega J, Dormido-Canto S, Baruzzo M and Gelfusa M 2018 *Nuclear Fusion* **58** 056002 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/58/i=5/a=056002?key=crossref.ddcfcc865f68a96ff567c36fb8cbe7a9

[6] Pau A, Fanni A, Cannas B, Carcangiu S, Pisano G, Sias G, Sparapani P, Baruzzo M, Murari A, Rimini F, Tsalas M and de Vries P C 2018 *IEEE Transactions on Plasma Science* **46** 2691–2698 ISSN 0093-3813 URL https://ieeexplore.ieee.org/document/8383700/

[7] Cannas B, de Vries P C, Fanni A, Murari A, Pau A and Sias G 2015 *Plasma Physics and Controlled Fusion* **57** 125003 ISSN 0741-3335 URL http://stacks.iop.org/0741-3335/57/i=12/a=125003?key=crossref.08aa124780e86432523c5b910ad780d4

[8] Aledda R, Cannas B, Fanni A, Pau A and Sias G 2015 *Fusion Engineering and Design* **96-97** 698–702 ISSN 09203796 URL https://linkinghub.elsevier.com/retrieve/pii/S0920379615002148

[9] Pautasso G, Tichmann C, Egorov S, Zehetbauer T, Gruber O, Maraschek M, Mast K F, Mertens V, Perchermeier I, Raupp G, Treutterer W, Windsor C and Team A U 2002 *Nuclear Fusion* **42** 100–108 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/42/i=1/a=314?key=crossref.9f5aa4f9f94faec26101b46fe6452846

[10] Wroblewski D, Jahns G and Leuer J 1997 *Nuclear Fusion* **37** 725–741 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/37/i=6/a=I02?key=crossref.ecdebb1473fb9ace04907429034f6cce

[11] Rattá G A, Vega J and Murari A 2018 *Fusion Science and Technology* **74** 13–22 ISSN 1536-

1055 URL https://doi.org/10.1080/15361055.2017.1390390https://www.tandfonline.com/doi/full/10.1080/15361055.2017.1390390

[12] Windsor C, Pautasso G, Tichmann C, Buttery R, Hender T, Contributors J E and Team t A U 2005 *Nuclear Fusion* **45** 337–350 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/45/i=5/a=004?key=crossref.170e4cfeab7836eaf142634f3e851578

[13] Parsons M S 2017 *Plasma Physics and Controlled Fusion* **59** 085001 ISSN 0741-3335 URL http://stacks.iop.org/0741-3335/59/i=8/a=085001?key=crossref.196e629f51178b35020667662ffc221f

[14] Gerhardt S, Darrow D, Bell R, LeBlanc B, Menard J, Mueller D, Roquemore A, Sabbagh S and Yuh H 2013 *Nuclear Fusion* **53** 063021 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/53/i=6/a=063021?key=crossref.4e9fe9246f128e88fc1b70d28ade6ead

[15] Rea C, Montes K J, Erickson K G, Granetz R S and Tinguely R A 2019 A Real-Time Machine Learning-Based Disruption Predictor on DIII-D (submitted to Nuclear Fusion) Tech. rep. MIT Plasma Science and Fusion Center URL http://library.psfc.mit.edu/catalog/reports/2010/19ja/19ja004/abstract.php

[16] Lao L, St John H, Stambaugh R, Kellman A and Pfeiffer W 1985 *Nuclear Fusion* **25** 1611–1622 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/25/i=11/a=007?key=crossref.382b4e7e430c8741af0f7248e9a56c09

[17] Rea C, Granetz R S, Montes K, Tinguely R A, Eidietis N, Hanson J M and Sammuli B 2018 *Plasma Physics and Controlled Fusion* **60** 084004 ISSN 0741-3335 URL http://stacks.iop.org/0741-3335/60/i=8/a=084004?key=crossref.28bdf1a3b6818b4ea295d05b7923f328

[18] Breiman L E O 2001 *Machine Learning* **45** 5–32 URL https://doi.org/10.1023/A:1010933404324

[19] Rea C and Granetz R S 2018 *Fusion Science and Technology* **74** 89–100 ISSN 1536-1055 URL https://doi.org/10.1080/15361055.2017.1407206https://www.tandfonline.com/doi/full/10.1080/15361055.2017.1407206

[20] Breiman L, Friedman J, Olshen R and Stone C 1984 *Classification and Regression Trees* (Chapmann & Hall) ISBN 978-0-412-04841-8

[21] Cannas B, Fanni A, Sonato P and Zedda M 2007 *Nuclear Fusion* **47** 1559–1569 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/47/i=11/a=018?key=crossref.93dedeb238489ae80a9d0b39ef426378

[22] Vega J, Dormido-Canto S, López J M, Murari A, Ramírez J M, Moreno R, Ruiz M, Alves D and Felton R 2013 *Fusion Engineering and Design* **88** 1228–1231 ISSN 09203796 URL https://linkinghub.elsevier.com/retrieve/pii/S0920379613002974

[23] Meneghini O, Smith S, Lao L, Izacard O, Ren Q, Park J, Candy J, Wang Z, Luna C, Izzo V, Grierson B, Snyder P, Holland C, Penna J, Lu G, Raum P, McCubbin A, Orlov D, Belli E, Ferraro N, Prater R, Osborne T, Turnbull A and Staebler G 2015 *Nuclear Fusion* **55** 083008 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/55/i=8/a=083008?key=crossref.5f4846d96e96da6689b641740716977c

[24] Hollmann E M, Aleynikov P B, Fülöp T, Humphreys D A, Izzo V A, Lehnen M, Lukash V E, Papp G, Pautasso G, Saint-Laurent F and Snipes J A 2015 *Physics of Plasmas* **22** 021802 ISSN 1070-664X URL http://aip.scitation.org/doi/10.1063/1.4901251

[25] Palczewska A, Palczewski J, Marchese Robinson R and Neagu D 2014 Interpreting Random Forest Classification Models Using a Feature Contribution Method *Advances in Intelligent Systems and Computing* (*Advances in Intelligent Systems and Computing* vol 263) ed Bouabana-Tebibel T and Rubin S H (Cham: Springer International Publishing) pp 193–218 ISBN 978-3-319-04716-4 URL http://link.springer.com/10.1007/978-3-319-04717-1{_}9

[26] Saabas A 2015 treeinterpreter URL https://github.com/andosa/treeinterpreter

[27] Csiszar C P 2018 waterfall URL https://github.com/chrispaulca/waterfall

[28] Vega J, Murari A, Dormido-Canto S, Moreno R, Pereira A and Acero A 2014 *Nuclear Fusion* **54** 123001 ISSN 0029-5515 URL http://stacks.iop.org/0029-5515/54/i=12/a=123001?key=

crossref.f4de2feb15eb8c8c84411fa26bba5154