MACHINE LEARNING FOR DISRUPTION WARNING ON ALCATOR C-MOD, DIII-D, AND EAST TOKAMAKS

R.S. GRANETZ, C. REA, K.J. MONTES, R.A. TINGUELY
Massachusetts Institute of Technology, Plasma Science and Fusion Center
Cambridge, MA, USA
Email: granetz@mit.edu

N. EIDIETIS, O.M. MENEGHINI
General Atomics
San Diego, CA, USA

D.L. CHEN, B. SHEN, B.J. XIAO
Institute of Plasma Physics, Chinese Academy of Sciences
Hefei, Anhui, China

K. ERICKSON, M.D. BOYER
Princeton Plasma Physics Laboratory
Princeton, NJ, USA

Abstract

The paper reports on disruption prediction using machine learning (ML), trained on large databases containing only plasma parameters that are available in real time on C-Mod, DIII-D, and EAST. It is found that the prediction algorithms can differ substantially in performance among the three machines on a timeslice-by-timeslice basis, but can have similar disruption detection rates 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 database for each tokamak contains parameters sampled at ~10^6 times throughout ~10^4 discharges, disruptive and non-disruptive, over the last 3-4 years of operation. It is found that a number of parameters (e.g. \( P_{\text{rad}}/P_{\text{input}} \), \( l_i/n_i \), \( B_{n=1}/B_0 \)) exhibit changes 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.

A shallow ML method known as Random Forests is applied to a binary classification scheme. Two classes are defined as “close to a disruption” and “far from a disruption or from a non-disruptive shot”. The threshold time that divides “close” from “far” is determined by optimising the classification prediction accuracy for each machine.

1. INTRODUCTION

Application of artificial intelligence using machine learning methods for generating real time warnings of impending disruptions in tokamaks is being developed because approaches based on first-principles plasma physics are too complex to be of practical use, particularly in real time. Our ML work described here uses a 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, proven to work on multiple present-day devices, can be developed, it may resolve this conundrum. 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 are 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 the next section we will describe the databases in some detail. In Section 3 we compare and contrast between machines the behavior of several plasma parameters. Section 4 describes the development of the Random Forests (RF) ML algorithms, and compares their disruption prediction performance on the three machines. In Section 5 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, followed by a summary and conclusions in Section 6.

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. 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 those used by [Gerhardt2013], and include diagnostic measurements such as \( I_p \) error \( = I_p - I_p \) (programmed), radiated power fraction \( = P_{rad}/P_{inp} \), \( n/n_{Greenwald} \), \( Z \) error \( = Z(centroid) - Z_{programmed} \), \( B_{pol} \), amplitude, as well as a number of equilibrium parameters derived from EFIT reconstructions \( (q_{pol}, l, \text{elongation}, \text{etcetera}) \). In order to facilitate multi-machine studies, many of these physics parameters are normalised to machine size or B-field where appropriate. 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 of such application is reported in section 5). 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, and additional sampling is done every 1 ms during the 20 ms period before each disruption. For DIII-D, all shots are sampled every 25 ms, and additional sampling is done every 2 ms for the 100 ms period before each disruption. And for EAST, all shots 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 timescales in each machine. 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. New parameters continue to be added from time to time. Many of the plasma parameters are derived from EFIT [Lao1985] reconstructions. In order to avoid excessive interpolation, as well as unacceptable non-causal smoothing, we have run our own EFITs on all the discharges, at the times we desire for our databases, and using only causal smoothing, only 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. Although disruptions certainly occur during rampup and rampdown, in ITER and future reactors (as well as many EAST discharges) the rampup and rampdown will be a negligible fraction of the discharge duration, and the plasma stored energy will also be relatively small. 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 \( (P_{rad}/P_{inp}) \), \( l/(\text{current profile peakedness}) \), Greenwald fraction \( (n/n_{\text{Greenwald}}) \), \( n=1 \) component of B, \( T_e \) 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 Figs. 1 and 2. In Fig. 1 the evolution of the \( n=1 \) toroidal Fourier harmonic of the magnetic field, \( B_{pol} \), is shown as a flattop disruption is approached, for thousands of disruptions on each machine. On C-Mod, although \( B_{pol} \) tends to increase on a notable fraction of 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, \( B_{pol} \) tends to increase slowly before disruptions, starting roughly a half-second before the disruption time. And on EAST, \( B_{pol} \) does not show any change of behavior as disruptions are approached. Another example is given in Fig. 2, 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 on C-Mod and DIII-D is less pronounced, and with much less warning time, particularly on C-Mod. Similar contrasting behavior between machines is also seen for the radiated power fraction, the normalised \( I_p \) error, and others. Our observations that the same plasma parameters generally show markedly different evolution leading up to disruptions could complicate the successful development of a universal disruption predictor.
The Machine Learning model adopted to develop the 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 via Random Forests. The methodological details of the Random Forests algorithm can be found in the original paper from Breiman [Breiman2001] and in previous publications from the authors [Rea2018a, Rea2018b]. It is a supervised algorithm, and therefore class labels need to be assigned to each sample in the available datasets. 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, the univariate analysis on the aforementioned plasma signals (see section 3), has motivated the selection for thresholds in time, specific for each device, used to define the sample class labels. 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 or non-disruptive”. 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 process.

Briefly summarising the Random Forests methodology, the forests are developed by growing a large number of independent, de-correlated decision trees, thus collecting a parallel set of predictions. Each tree is 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 normalisation 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 stabilisation. 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.

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 [Windsor2005, Cannas2007, Vega2013, Gerhardt2013]. 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).

**TABLE 1. LIST OF SIGNALS USED FOR DEVELOPING DPRF ON DIFFERENT TOKAMAKS**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Signal description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_equal_1_normalised</td>
<td>$B_{n=1}$ perturbed radial field of nonrotating modes, normalised to $B_{tor}$</td>
</tr>
<tr>
<td>q95</td>
<td>Safety factor at the 95% flux surface</td>
</tr>
<tr>
<td>n/nG</td>
<td>Greenwald density fraction</td>
</tr>
<tr>
<td>ip_error_frac</td>
<td>Fractional error between measured and programmed plasma current</td>
</tr>
<tr>
<td>li</td>
<td>Normalised internal inductance</td>
</tr>
<tr>
<td>betap</td>
<td>Poloidal beta</td>
</tr>
<tr>
<td>Vloop</td>
<td>Loop voltage [V]</td>
</tr>
<tr>
<td>Wmhd</td>
<td>Stored plasma energy [J]</td>
</tr>
<tr>
<td>*Te_width_normalized</td>
<td>Electron temperature profile width, normalised to plasma minor radius</td>
</tr>
<tr>
<td>radiated_fraction</td>
<td>Total radiated power divided by total input power</td>
</tr>
</tbody>
</table>

*For the C-Mod DPRF, the Te profile width had to be dropped from the list of input features because those data were missing from a significant number of discharges.

### 4.1 Time-slice performances

Before going into the details of a shot-by-shot analysis, we report DPRF performances in terms of a confusion matrix for each device (Fig. 3). DPRF was trained using a different threshold for the class label separation, $t_{class}$, 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, and then corroborated through a cross-validation procedure described in the next subsection. In contrast, the threshold time for C-Mod was set at a minimum value that is practically useful for disruption warning purposes. The performances reported in Fig. 3 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 it is far from perfection, ranging from ~60% for DIII-D, to just ~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.

![Fig. 3](image_url)

**FIG. 3 – DPRF performances on time slice basis are summarised in a confusion matrix for each tokamak. Several different figures of merit can be discerned from these, as described in the text. 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 or non-disruptive” time slice. The fraction of correctly predicted disruption samples varies considerably, ranging from ~60% for DIII-D, to just ~22% for EAST. Note that the performances are very different when considered on a shot-by-shot analysis.**
4.2 Shot-by-shot performances

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 disruptivity value for a single time slice. In order to provide a more accurate warning of an impending disruption, we desire to evaluate the performance of DPRF on a shot-by-shot basis. We do this by using a hysteresis window 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 triggered. 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). Since the alarm trigger is a function of the operational parameters (i.e. the chosen disruptivity thresholds, alarm window, and $\tau_{\text{class}}$), 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 recall $TP/(TP + FN)$. For each machine, we have optimised these rates over the operational space on the corresponding training set via a K-fold cross-validation procedure by maximising a binary classification metric called the $F_\beta$-score, given by

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

where we have chosen $\beta = 2$ to weight the recall. Upon splitting the training set into $K = 5$ subsets, random forests were trained on each combination of $K-1$ subsets and tested on the corresponding held out subset for each point in the operational parameter grid. We then calculate the mean $F_2$-score of the $K$ trainings splits at each operational point and record the $\tau_{\text{class}}$, high threshold, low threshold, and alarm window combination that corresponds to the maximum $F_2$ value. These ‘best’ parameters are then used to train a DPRF model on the entire training set and apply it on the unseen test set. The results of this generalisation and ‘best’ points for each machine are shown in Figure 4.

![Cumulative fraction of test set disruptions detected with at least the given warning time (blue) for Alcator C-Mod, DIII-D, and EAST; the legend shows the class label time and high disruptivity threshold for the ‘best’ operational point, as well as the true and false positive rates](image)

The cross-validation results reveal that smaller alarm windows tend to have higher performance metrics - the ‘best’ point for each machine is found at a window size $\sim 1$ ms, below the sampling period in our databases. Although windows of this size are helpful for interpreting a noisy, densely sampled real-time disruptivity signal (see Section 5), the low threshold is ineffective for offline analysis of our database in this operational space. Therefore, we only included the optimal $\tau_{\text{class}}$ and high threshold values for each machine in Fig. 4. Note that the optimised $\tau_{\text{class}}$ values on DIII-D and EAST are near the time of the distribution shift of parameters like the $B_{n=1}$ and $V_{\text{loop}}$ (see Figs 1 and 2). This is correlated with a higher fraction of predicted disruptions when compared to C-Mod, for which $\tau_{\text{class}}$ falls well outside the region of any observable change in the physics parameter distributions. Since the disruptivity threshold for the alarm trigger is less than 50% for each tokamak, we also see that the random forest output should not be thought of as an injective mapping to disruption probability. Rather, the signal must be calibrated separately for each machine in order to yield an optimised performance.
4.3 Predictions interpretation via feature contribution analysis

One of the most disconcerting aspects of Machine Learning algorithms is their difficulty to provide explainable predictions. Most of the state-of-the-art models are indeed regarded as black box, even though it is now considered cutting-edge AI research to provide insights on why specific predictions were made (see techniques for deep learning such as layerwise relevance propagation or axiomatic attribution methods). Random Forests are powerful ML algorithms because they not only provide information on the training dataset through importance values for input features, but also can guarantee explainable predictions by accessing the information contained in each of the trees decision paths [Palczewska2014].

Trees interpretation means that each prediction can be linearly decomposed into the contribution (positive or negative) coming from the \( k \)-th splitting feature. Joint contributions can also be obtained. If we consider a linear regression approach [Saabas], then the prediction cast by one tree can be interpreted as:

\[
prediction_j = bias + \sum_{k=1}^K contrib(k).
\]

For the full forest it becomes:

\[
prediction = \frac{1}{J} \sum_j^{\frac{1}{J}} bias_j + \frac{1}{J} \sum_j^{\frac{1}{J}} \left( \frac{1}{J} \sum_j^{\frac{1}{J}} contrib(k) \right),
\]

where \( J \) is the total number of trees in the forest and \( k \) represents the \( k \)-th feature. The numerical values for the feature contributions add together algebraically to give the Random Forest output value for each time slice. A negative feature contribution for a given time slice means that the feature's measured value pushes you towards a region of feature space that is in the “far from disruption or non-disruptive” class. Examples of breaking down the disruptivity into its feature contributions are shown for DIII-D in Figures 6 and 7, and for EAST in Figure 8.

5. REAL TIME MACHINE LEARNING-BASED ALGORITHMS ON DIII-D AND EAST

5.1 DIII-D PCS algorithm

A DPRF routine to run in the DIII-D PCS in real time was developed by training on the same signals that are furnished to the PCS in real time, including those from real time EFITs. The training process used scikit-learn [Pedregosa2012], the open-source Python library, through the OMFIT framework [Meneghini2015]. To integrate DPRF in the real time DIII-D environment, the trained forest was translated [Morawiec] into C, the PCS-compatible language. DPRF has continuously and safely run in DIII-D PCS for more than 4 months of operations, gathering data on about 850 discharges, 66% of which were non-disruptive, 6% disrupted during the flattop, and the remaining 28% disrupted during rampup or rampdown. We only train on the flattop portion of discharges (both disruptive and non-disruptive).

In Fig. 5, we show an example of a non-disruptive discharge: the disruptivity predictions are shown in the second panel of Fig. 5, together with the average computing time for DPRF predictions, which ranges around 250-300 \( \mu s \). In Fig. 6 we show the same non-disruptive DIII-D discharge, together with the breakdown of the disruptivity predictions into the features contributions. It is possible to see that no particular feature contributes much to the disruptivity during the flattop phase; the disruptivity itself ranges around 15% throughout the flattop.

In contrast, Fig. 7 shows an example of a flattop disruption: it is interesting to note that, for this particular case, the disruptivity signal increases above 60% before the impending disruption (at \( \sim 4.6 \) s) with more than 150 ms warning time. \( n_{equal_1 normalised, q95} \) and \( n/nG \) deeply affect the disruptivity predictions.
5.2 EAST VDE’s experiments

The EAST DPRF algorithm has been recently tested between shots during experiments performed to purposely trigger VDE’s. 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 (kappa), and the current centroid information (zcur_lmsz), plus the error between the programmed current centroid position and the actual reconstructed one (z_error_lmsz). A representative discharge is shown in Fig. 8: in the first panel, the plasma signal (black) and the disruptivity (blue; causally smoothed with a 10 ms window) are reported, while in the second and third panels it is possible to see the contributions from each input feature (just for visualisation purposes, the feature contributions are split in two panels). From Fig. 8 it is possible to see that the disruptivity signal is strongly affected by the elongation and the current centroid signals, thus reflecting the actual changes in the physics of the discharge.

6. SUMMARY AND CONCLUSIONS

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. 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, an encouraging result that we attribute to the extremely low rate of false positives. Apparently, the correct recognition of just a few positive time slices far outweighs lots of false negative 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.

Thanks to the Random Forests 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 (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 is achieved, and the most appropriate actuators are identified for possibly avoiding impending disruptions.

Future work includes taking an
algorithm that is trained on data from one of the machines, and testing it on the other machines, as well as training an algorithm using data from all machines together.

FIG. 8 – EAST DPRF disruptivity prediction for discharge 81317. In the first panel, the plasma current (black) and the disruptivity (blue) are reported. The second and third panels show the breakdown of disruptivity in terms of feature contributions. 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 VDE’s were not in the training set.

ACKNOWLEDGEMENTS

This work supported by the U.S. Department of Energy, Office of Science, and Office of Fusion Energy Sciences under awards DE-FC02-99ER54512, DE-SC0014264, DE-SC0010720, DE-SC0010492, DE-FC02-04ER54698. Part of the data analysis was performed using the OMFIT integrated modeling framework [Meneghini2015]. 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