Risk-Aware Framework Development for Disruption Prediction: Alcator C-Mod and DIII-D Survival Analysis

Citation: Keith, Z., Nagpal, C., Rea, C. et al. Risk-Aware Framework Development for Disruption Prediction: Alcator C-Mod and DIII-D Survival Analysis. J Fusion Energ 43, 21 (2024).

As Published: 10.1007/s10894-024-00413-y

Publisher: Springer Science and Business Media LLC

Persistent URL: https://hdl.handle.net/1721.1/155279

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Journal of Fusion Energy (2024) 43:21
https://doi.org/10.1007/s10894-024-00413-y

RESEARCH

Risk-Aware Framework Development for Disruption Prediction: Alcator C-Mod and DIII-D Survival Analysis

Zander Keith[1] · Chirag Nagpal[2] · Cristina Rea[1] · R. Alex Tinguely[1]

Accepted: 12 May 2024

Abstract

Survival regression models can achieve longer warning times at similar receiver operating characteristic performance than previously investigated models. Survival regression models are also shown to predict the time until a disruption will occur with lower error than other predictors. Time-to-event predictions from time-series data can be obtained with a survival analysis statistical framework, and there have been many tools developed for this task which we aim to apply to disruption prediction. Using the open-source Auton-Survival package we have implemented disruption predictors with the survival regression models Cox Proportional Hazards, Deep Cox Proportional Hazards, and Deep Survival Machines. To compare with previous work, we also include predictors using a Random Forest binary classifier, and a conditional Kaplan-Meier formalism. We benchmarked the performance of these five predictors using experimental data from the Alcator C-Mod and DIII-D tokamaks by simulating alarms on each individual shot. We find that developing machine-relevant metrics to evaluate models is an important area for future work. While this study finds cases where disruptive conditions are not predicted, there are instances where the desired outcome is produced. Giving the plasma control system the expected time-to-disruption will allow it to determine the optimal actuator response in real time to minimize risk of damage to the device.

Keywords Tokamak plasma · Disruption prediction · Machine learning · Survival analysis

Introduction

The tokamak's simple torus-shaped design and good confinement properties have made it the most constructed and studied magnetic confinement fusion concept in the world. From this successful history, understanding of the physics basis, and familiarity with the engineering requirements, there are presently several startup companies such as Commonwealth Fusion Systems [1], as well as the international collaboration on the ITER project [2] pursuing tokamaks as a pathway to the first fusion pilot plant. However, plasma confinement in tokamaks is prone to instabilities, which may lead to disruptions [3]. A disruption is the sudden and complete loss of plasma confinement, inflicting large thermal and structural loads on plasma facing components. In present day experimental devices, disruptions are relatively harmless, but in a future tokamak power plant the stored thermal and magnetic energy will be high enough to cause significant damage. Over the course of a discharge (shot), the plasma control system (PCS) must be able to predict disruption onset with sufficient warning time to take action for avoiding the disruption or minimizing damage.

The best choice of action to address an oncoming disruption depends on the associated risks of damage. Triggering disruption mitigation systems (DMS) such as massive gas injection can be done on short timescales (10's of ms); however, this speed comes at a price. DMS essentially creates a disruption with reduced thermal and structural loads [4]. For example, the experimental tokamak SPARC [5] is designed to withstand the mechanical strains

✉ Zander Keith
zkeith@mit.edu

Chirag Nagpal
chiragnagpal@google.com

Cristina Rea
crea@mit.psfc.edu

R. Alex Tinguely
tinguely@psfc.mit.edu

[1] Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, MA, USA

[2] Google Research, Google, San Francisco, CA, USA

Published online: 10 June 2024

from at least 300 unmitigated or 1800 mitigated disruptions [6]. Terminating the shot early with a safe ramp-down of the plasma current is less hazardous, but this requires a long warning time (100's of ms). Ideally, the PCS should be able to make an informed decision on which action to take considering the length of warning time before a disruption occurs. Determining this warning time is challenging, as there are many potential causes of disruptions with a wide range of timescales. However, decades of experimental data from a diverse set of tokamaks in a variety of operating regimes provides a robust foundation for exploring a data-driven approach to disruption prediction and avoidance [7].

A non-exhaustive list of data-driven approaches include those deployed on the ASDEX-Upgrade [8], DIII-D [9], JET [10], and EAST [11] tokamaks. There have also been studies using experimental data for cross-machine modeling [12]. The models used in these systems predict disruption onset with reasonable performance, and are typically able to correctly identify 90% of disruptive states (true positive rate–TPR) while only labeling 5% of stable states as disruptive (false positive rate–FPR). ITER expects to require a TPR of at least 95% [13], which would lead to a significantly higher FPR with the present models. This is unacceptable in situations where false positives come with a high risk of damage. Even if the damage caused by DMS is negligible, significant downtime due to continuous premature plasma termination would prove detrimental to the economics of a tokamak power plant [14].

These data-driven models also only give a binary output: whether or not a disruption will occur in the future. While some implementations such as the random forest described in [9] offer insight in a postmortem analysis, the information given to the PCS in real time is limited. The task of identifying the least hazardous actions to avoid disruptions is similar to the time-to-event predictions common in healthcare for selecting treatments based on mortality risk. This statistical framework is called survival analysis, and is well-established in healthcare and other fields [15, 16]. In the context of tokamak operation, survival analysis should allow the prediction of both if a disruption will occur in the future and the window of time when it is most likely to take place. Previous work has applied survival analysis to disruption prediction in tokamaks [17, 18]; however, these studies did not compare performance with other machine learning methods.

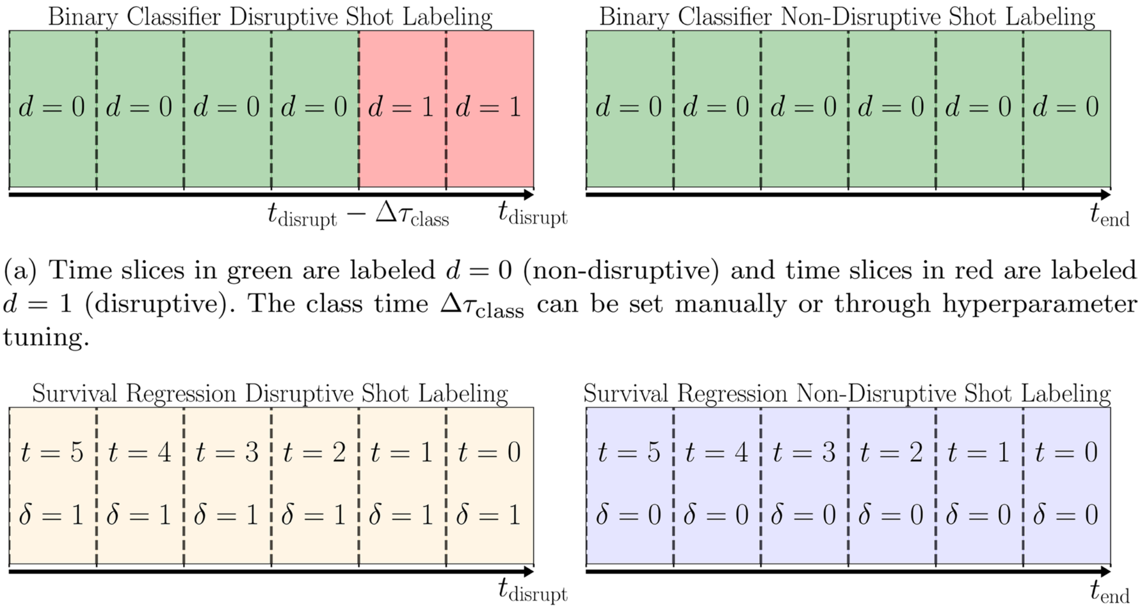
Using the open-source Auton-Survival package [19] along with data from the Alcator C-Mod [20] and DIII-D [21] tokamaks, we have benchmarked the performance of the binary classifier model Random Forest (RF) [9, 22] against the conditional Kaplan-Meier formalism (KM) [17, 23], as well as the survival regression models Cox Proportional Hazards (CPH) [24], Deep Cox Proportional Hazards (DCPH) [15], and Deep Survival Machines (DSM) [25]. The remainder of this paper is organized as follows: The "Models" section describes the differences between the types of machine learning algorithms investigated in this study, and a full list of models and their acronyms is given in Table 1. The "Methods for Comparing Disruption Predictor Performance" section covers the methodology used to compare model performance based on various metrics. The "Model Training and Bootstrap Results" section provides the calculated metrics for each model in various scenarios when applied to data from the Alcator C-Mod and DIII-D tokamaks, and discusses the importance of various diagnostics. The "Estimating Time-to-Disruption" section investigates using the models to estimate time-to-disruption in Alcator C-Mod and DIII-D, and provides calculated output for selected disruptive and non-disruptive shots on Alcator C-Mod. Lastly, the "Summary" section reviews the analysis and covers areas for future work.

Models

Binary Classifiers

The typical method for predicting disruption onset with machine learning is to use binary classification. There are many types of binary classifiers, and for this study we will employ a RF implementation as used in [9]. We assume there exists a transition from a stable plasma state to a disruptive plasma state at a point in time before the disruption takes place, and that this transition can be detected using plasma stability measurements of various parameters such as electron density, plasma current, shaping, etc. Over the course of a shot, real-time diagnostics sample at some finite temporal resolution. The signals from each shot are then a series of time slices.

To train a binary classifier, the time-series data are represented as a collection of N tuples $\{\vec{x}_i, d_i\}_{i=1}^{N}$, where $\vec{x}_i$ is a vector of features (plasma measurements at a given time) and d_i is the label of the features so each time slice is either disruptive ($d = 1$) or non-disruptive ($d = 0$). For shots where no disruption is observed, all time slices are labeled as non-disruptive. For shots where a disruption is observed, time slices within a specified time window before the disruption are labeled as disruptive, while time slices outside this window are labeled as non-disruptive. This time window used for labeling is called the class time $\Delta\tau_{\mathrm{class}}$, and can be determined using plasma physics intuition for certain disruptive instabilities. For example, on DIII-D the amplitudes of magnetic field perturbations are observed to start growing around 50 ms before locked mode disruptions take place [26]. Alternatively, precursors for disruptions caused by impurity buildup may be expected to evolve on the energy confinement timescale (100's of ms). The class


Binary Classifier Disruptive Shot Labeling
$d=0$
$d=0$
$d=0$
$d=0$
$d=1$
$d=1$
$t_{\text{disrupt}} - \Delta\tau_{\text{class}}$
t_{disrupt}
Binary Classifier Non-Disruptive Shot Labeling
$d=0$
$d=0$
$d=0$
$d=0$
$d=0$
$d=0$
t_{end}
Survival Regression Disruptive Shot Labeling
$t=5$ $t=4$ $t=3$ $t=2$ $t=1$ $t=0$
$\delta=1$ $\delta=1$ $\delta=1$ $\delta=1$ $\delta=1$ $\delta=1$
t_{disrupt}
Survival Regression Non-Disruptive Shot Labeling
$t=5$ $t=4$ $t=3$ $t=2$ $t=1$ $t=0$
$\delta=0$ $\delta=0$ $\delta=0$ $\delta=0$ $\delta=0$ $\delta=0$
t_{end}

(a) Time slices in green are labeled $d = 0$ (non-disruptive) and time slices in red are labeled $d = 1$ (disruptive). The class time $\Delta\tau_{\text{class}}$ can be set manually or through hyperparameter tuning.

(b) For the time slices of the disruptive shot, $\delta_i = 1$ since a disruption is observed. For the time slices of the non-disruptive shot, $\delta_i = 0$ since the shot ends (data is censored) before a disruption is observed.

Fig. 1 Comparison between labeling of time-series data for a disruptive shot (left) and a non-disruptive shot (right) for binary classification (**a**) and survival regression (**b**)

time could also be left as a value included in hyperparameter tuning when optimizing for performance of a particular metric. A diagram of labeling data is shown in Fig. 1a. Once trained, the binary classifier takes as input a new feature vector $\vec{x}$ and outputs a value between 0 and 1, which is the classifier's prediction that $\vec{x}$ should be labeled as disruptive. The output of a binary classifier can be interpreted as the probability that the present time slice is disruptive, or $P_D(\vec{x})$.

Conditional Kaplan–Meier Formalism

An extension of a binary classifier using the survival analysis framework is the conditional Kaplan–Meier formalism (KM) disruption predictor as described in [17]. In this implementation, the outputs of a binary classifier are extrapolated and used to predict the risk of a disruption occurring at a future point in time.

First, a moving window of duration Δt_{fit} is used to calculate the line of best fit for several previous predictions made by the binary classifier. One can then extrapolate from the present time slice $\vec{x}$ at time t the probability that a future time slice $\vec{x}_f$ at time $t + \Delta t_{\text{horizon}}$ is in the disruptive class using the equation

$$P_D(\vec{x}_f) \approx P_D(\vec{x}) + \frac{dP_D}{dt}\Delta t_{\text{horizon}} \tag{1}$$

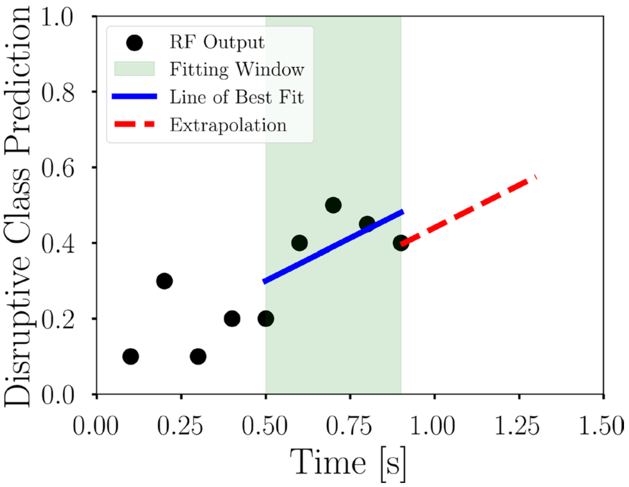


Fig. 2 An illustration of the KM disruption predictor implemented in [17]. The output of a binary classifier (black dots) within the window Δt_{fit} (green) is used to calculate a line of best fit (blue). The slope of this line is then extrapolated into the future (red) to determine the confidence that a time slice at $t + \Delta t_{\text{horizon}}$ should be labeled as disruptive

where $P_D(\vec{x}_f)$ is restricted to the interval [0, 1], $\frac{dP_D}{dt}$ is obtained from the line of best fit, and $\Delta t_{\text{horizon}}$ is how far into the future the prediction is being made [17]. Both Δt_{fit}

Table 1 Acronyms for models

Acronym	Model	Training data labels	Deep learning
RF	Random forest [22]	Binary classification	No
KM	Conditional Kaplan–Meier [23]	Binary classification	No
CPH	Cox proportional hazards [24]	Survival regression	No
DCPH	Deep cox proportional hazards [27]	Survival regression	Yes
DSM	Deep survival machines [25]	Survival regression	Yes

and $\Delta t_{\text{horizon}}$ can be set manually or left to hyperparameter tuning. An illustration of this is shown in Fig. 2.

Censoring-Informed Survival Regression Models

We will investigate three types of survival regression models: Cox Proportional Hazards (CPH), Deep Cox Proportional Hazards (DCPH), and Deep Survival Machines (DSM). These models are available in the open-source Auton-Survival package [19]. The training data for a survival regression model is given as $\{(\vec{x}_i, t_i, \delta_i)\}_{i=1}^N$ [25], where $\vec{x}_i$ is the feature vector, t_i is time until the final measurement, and δ_i is whether or not an event was observed at the final measurement. A comparison between the labeling of disruptive and non-disruptive shots for the survival regression models is shown in Fig. 1b.

Once trained, the survival regression model takes as input a new feature vector $\vec{x}$ and a horizon time $\Delta t_{\text{horizon}}$. The output is then the probability that an event will be observed within the time interval $[t, t + \Delta t_{\text{horizon}}]$. This framework has several advantages which make it more readily applicable to disruption prediction compared to a binary classifier. First, the survival regression models take into account censoring which may otherwise be absent from binary classifiers [25]. For instance, if a plasma is in a disruptive state but the shot ends before a disruption occurs, this would lead to incorrect labeling for the final time slices of the data and negatively impact the binary classifier's training.

The output of the survival regression models is also more interpretable than a binary classifier. As stated previously, the class time in a binary classifier determines how the output must be interpreted. One would need to re-train a new RF with varying class times to be able to differentiate between various horizons. With the survival regression models, the input includes the features to be evaluated and a horizon time in the future. One can use multiple horizons to calculate the probability of an event happening within some arbitrary future time window using a single survival regression model.

Methods for Comparing Disruption Predictor Performance

Our goal is to compare the performance of the above models in a way that ensures the development of a risk-aware framework for disruption handling relevant to tokamak operation. We will start by evaluating the survival analysis models as disruption predictors when trained and tested on the same device. This requires a large existing dataset, which may be infeasible to obtain for future tokamaks if disruptions cause significant damage. It has been shown that when simply using normalized signals for the input features, the predictive performance of data-driven techniques suffers when being applied to a device whose data is not included in training [28]. There are methods that aim to solve this problem by using a large amount of data from existing devices combined with limited data from a new device. These approaches include augmenting training data by applying surrogate techniques to synthetically increase the number of disruptive shots [29], and dynamically adapting the training set to include the most relevant data as it becomes available [30, 31]. It has been demonstrated that these techniques can greatly improve the performance of machine learning models on new devices with limited data [12, 32]. Similar methods could be applied to survival analysis models, though this is an area for future work and we will not do so in this study. As such, any results presented here will be optimistic when compared to potential performance in a new tokamak. Despite this limitation, we will still be able to determine the efficacy of survival analysis models compared to previously investigated models when used in the same way, and illustrate the difficulties that arise when attempting to compare performance of different machine learning algorithms.

While binary classifiers and survival regression models may be trained to reduce misclassifications for each individual time slice of the dataset, this value is not immediately applicable to machine operation. What a tokamak operator requires is the model's prediction performance on a shot-by-shot basis. To calculate this performance, we will use similar methodology as presented in [9]. We calculate the model's output risk values over the course of a shot and find the first time when an alarm

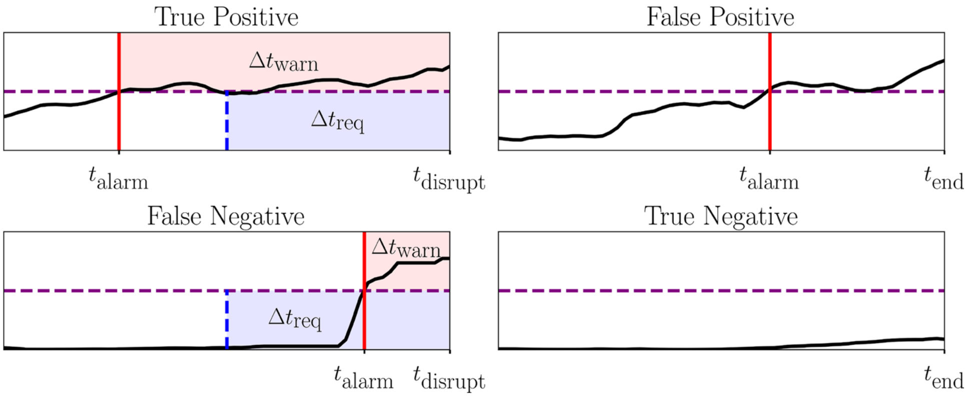


Fig. 3 Illustration of how alarms, true positives, false positives, and warning times are defined in our benchmarking methodology. If the predicted risk value (black) exceeds some threshold (purple dashed) an alarm is triggered. For disruptive shots (left), the warning time Δt_{warn} (red) is compared to the required warning time Δt_{req} (blue). If $\Delta t_{warn} > \Delta t_{\mathrm{req}}$ the alarm is recorded as a true positive (upper left). Otherwise, the alarm is recorded as a false negative (lower left). For non-disruptive shots (right), if an alarm is ever triggered it is recorded as a false positive (upper right). Otherwise it is recorded as a true negative (lower right)

is triggered. For non-disruptive shots, if an alarm is ever triggered this is recorded as a false positive. For disruptive shots, an alarm is only recorded as a true positive if the warning time is long enough for the PCS to execute some response. Warning time is defined as the time before a disruption that an alarm is triggered, $\Delta t_{\mathrm{warn}} = t_{\mathrm{disrupt}} - t_{\mathrm{alarm}}$ (if no alarm is triggered, $\Delta t_{\mathrm{warn}} = 0$). Then for some required warning time Δt_{req} we enforce the condition $\Delta t_{\mathrm{warn}} > \Delta t_{\mathrm{req}}$. A sketch of this is shown in Fig. 3. The exact value of Δt_{req} depends on the actuators in a future device. In ITER, it is anticipated that the response time of DMS will be around 30 ms [33], while in SPARC it is expected to be 20 ms. In this study we will compare performance of each model with a required warning time of 10 ms, 50 ms, and 100 ms.

Performance Metrics

There are numerous metrics which can be used to evaluate the performance of a predictive model. The first we will consider is the Receiver Operating Characteristic (ROC) curve, which details the tradeoff between TPR and FPR. The area under this curve can be used as a metric of overall model performance (AUROC), where higher is better. A perfect AUROC score is 1, and in a situation with even class balance randomly guessing the labels yields an AUROC score of 0.5. While this metric is widely used, it is challenging to meaningfully interpret AUROC values in the context of disruption prediction and avoidance.

The primary drawback is AUROC assumes equal misclassification costs, in that false negatives and false positives have the same weighting. A predictive model deployed in a PCS should never be expected to operate in all regions of the ROC curve simultaneously. In future devices such as ITER or SPARC, alarm sensitivity must be optimized for the costs of false negatives, false positives, and disruption frequency in the scenario that an operator will run [33]. While all possible operating points are included in the ROC curve, the total area is irrelevant when determining a model's performance at the optimal TPR/FPR point. We will still include AUROC results to compare with previous studies that reported this metric.

Another metric which is specifically used in the context of evaluating survival analysis models is the Activity Monitoring Operating Characteristic (AMOC) curve [34], which describes the tradeoff between time to detect that a particular condition is met and FPR. For the purposes of disruption prediction where we are instead looking at time before an event, we will look at a curve describing

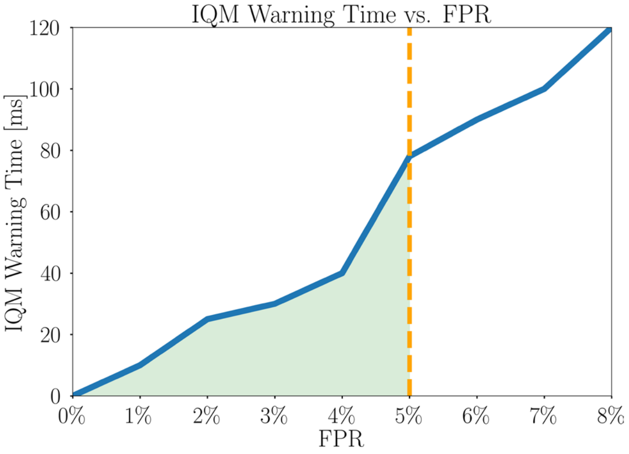


Fig. 4 Depiction of a warning time characteristic (WTC) curve. Obtained by sweeping through many alarm configurations and finding the interquartile mean (IQM) of warning times corresponding to each unique FPR. Area under this curve (AUWTC) is only counted for the FPRs under 5%, shaded in green

the tradeoff between typical warning time and FPR, or a Warning Time Characteristic (WTC) curve.

Similar to an ROC curve, the area under this WTC curve (AUWTC) yields a metric on how well the model performs, where higher is better. Unlike an ROC curve, we are unable to include the entire area of the curve in this calculation, since long warning times could easily be achieved with high FPR. This is not a meaningful result, so we will limit this area calculation to only include FPRs below some cutoff value. For this study we arbitrarily chose a FPR of 5%, but this could easily be varied within the presented framework. A plot of this metric is shown in Fig. 4. Note that this area has units of ms, and in the calculation the actual warning times are multiplied by small values from the FPR.

For AUROC and AUWTC, we must determine the relationship between FPR, TPR, and typical warning times for a model. The most straightforward approach is using a simple threshold alarm, as shown in Fig. 3. In this alarm type, a threshold value is set and once the model's predicted risk value exceeds it an alarm is triggered. To obtain ROC and WTC curves with this alarm type, the FPR, TPR, and warning times are calculated for many threshold values [12]. In this study, the threshold values were determined by every unique risk calculated by a model over the entire dataset. This ensures each threshold has a unique FPR, TPR, and set of warning times, but there could be multiple TPRs and warning times corresponding to a single FPR and vice versa. This is due to the fact that the FPR is determined by the threshold values in the non-disruptive shots, which is independent from the TPR and warning times that are determined by the threshold values in the disruptive shots.

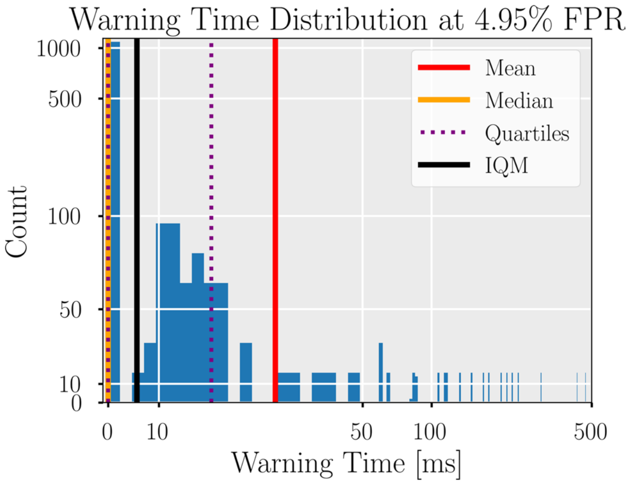


Fig. 5 Distribution of warning times from a CPH predictor on Alcator C-Mod data at 4.95% FPR, plotted on semi-log scales. The mean (red) is heavily skewed by outliers, while the median (orange) is zero from the low TPR. The IQM (black) provides a nonzero value while being resistant to the influence of outliers

To resolve this, we chose to group TPRs and warning times by FPR. For each unique FPR, we found all the alarm thresholds which produced that FPR and then grouped the TPRs and warning times for those thresholds together. The reported TPR is then the mean of all the TPRs corresponding to a single FPR, while the reported warning time is the interquartile mean (IQM) of all the warning times corresponding to a single FPR. The IQM was chosen due to the distribution of warning times in the low FPR region. An example of this is shown in Fig. 5. The warning times tend to be heavily skewed right; thus, the arithmetic mean is impacted by outliers. The low FPR region also tends to have a low TPR, leading to the majority of the warning times being 0 and therefore having a median value of 0. The IQM is both resistant to outliers while yielding a nonzero value that can be used to compare the models' performance.

Dataset Description

The datasets used in this study are composed of experimental data from the 2012–2016 campaigns of Alcator C-Mod and the 2014–2018 campaigns of DIII-D [12]. The 16 signals used are in Table 2. These signals were chosen because they are available for both C-Mod and DIII-D. In addition, all of these signals are expected to be available in real time to the PCS with sampling frequency of at least 200 Hz on future experimental devices like SPARC.

For this study, we are only considering disruptions during the steady state phase of the shot with the highest plasma

Table 2 Signals in Dataset

Signal	Description	Source
I_p	Plasma current	Magnetics
$I_p - I_{pprog}$	Difference between plasma current and programmed plasma current	Magnetics
V_{loop}	Loop voltage	Magnetics
$B^{n=1}/B_{\text{tor}}$	Normalized magnitude of $n = 1$ mode	Magnetics
Upper gap	Distance between plasma and upper divertor	Magnetics
Lower gap	Distance between plasma and lower divertor	Magnetics
ℓ_i	Normalized internal inductance	Equilibrium fit
κ	Plasma elongation	Equilibrium fit
β_p	Poloidal beta	Equilibrium fit
β_N	Normalized beta	Equilibrium fit
q_{95}	Safety factor at 95% normalized flux surface	Equilibrium fit
W_{mhd}	Total stored energy	Equilibrium fit
P_{rad}	Radiated power	Bolometry
$P_{\text{rad}}/P_{\text{in}}$	Radiated power/input power	Bolometry
n_e	Electron density	Interferometry
n/n_G	Greenwald density fraction	Interferometry

current (flattop). Only data from this flattop region was included in this study. The flattop region was determined by intervals of time where the programmed plasma current had a magnitude of at least 100 kA and the time derivative had an absolute value less than 60 kA / s. Typical flattop durations for the shots in the dataset are 1 s for C-Mod and 4 s for DIII-D. Shots with a flattop duration of less than 150 ms, disrupted after the flattop phase, or encountered hardware failures were discarded. In both datasets there is a flag that indicates if a shot was intentionally disrupted by the operators; however, depending on the type of study being done this flag may not have been set. We discarded all shots that were labeled as intentionally disrupted, though there could be intentionally disruptive shots still included.

After filtering, the C-Mod dataset had 5682 shots, where 5052 were stable and 630 were disruptive, while the DIII-D dataset had 7417 shots, where 6869 were stable and 548 were disruptive. For model training, hyperparameter tuning, and calculating performance metrics, the datasets were split into training, validation, and testing categories in ratios of 60%, 20%, 20% respectively, ensuring each category had the same ratio of stable and disruptive shots. The makeup of each dataset for C-Mod and DIII-D is shown in Table 3.

The signals were also put on a new timebase, since the original datasets had non-uniform sample rates between devices and had higher sampling rates in the region just before a disruption [35]. For C-Mod the original dataset had a sampling frequency of 50 Hz that increased to 1 kHz for the 20 ms before a disruption. For DIII-D the original dataset had a sampling frequency of 40 Hz that increased to 500 Hz for the 100 ms before a disruption. The authors note that this structure was chosen to reduce the size of the dataset while including fast dynamics close to a disruption [35]. For this study, all of the signals for both devices were linearly interpolated to a uniform 200 Hz, so each time slice has a duration of 5 ms. As mentioned previously, all signals used in this study should be available in real time at a temporal resolution of at least 200 Hz in future high performance tokamaks. There are some signals which could potentially be acquired with greater frequency, such as those measured with magnetic probes. The results shown may improve if the training data were provided with these signals at a higher temporal resolution. However, many of the included signals are global physical quantities that should be expected to evolve on resistive or energy confinement timescales considerably longer than this 5 ms time slice duration.

Before training, the signals were transformed using the preprocessing functions provided in Auton-Survival [19], where each measurement is normalized by subtracting the mean and dividing by standard deviation. The feature vectors were then created by concatenating 10 time slices of data together. That is, the feature vector for a given time is composed of the present and the previous 9 measurements, or $\vec{x}_i = \{x_{i-9}, x_{i-8}, ..., x_{i-1}, x_i\}$. For the first several time slices where there are not enough previous measurements to make a full vector, the missing values are filled with zeroes. With a 5 ms time slice duration this means each feature vector contains 50 ms of temporal information, similar to [36]. Note that this treatment is strictly causal, as only past and present measurements are included.

Model Training and Bootstrap Results

To compare the performance of the models as disruption predictors in a variety of scenarios, AUROC and AUWTC metrics have been calculated for each dataset using several required warning times. One instance of each model was hyperparameter tuned to maximize AUROC evaluated at a particular required warning time, with an additional model tuned to maximize AUWTC. This was done because the required warning time may significantly impact the optimal hyperparameters for a model. In a future system where the required warning time is determined by the DMS hardware, the model should be hyperparameter tuned and have its performance evaluated on the same value of Δt_{req}. In addition, if a model was going to be selected by its AUWTC score, it should be hyperparameter tuned to maximize AUWTC. This leads to 40 models total that we investigated, one for each combination of model type (RF, KM, CPH, DCPH, DSM), evaluation metric (AUROC at 10 ms, 50 ms, 100 ms, or AUWTC), and dataset (C-Mod, DIII-D).

The hyperparameter tuning was accomplished using the Optuna library [37] with a Bayesian inference search scheme. Models were trained using only data from the training set, and the metrics of interest calculated for the validation set on a shot-by-shot basis. Hyperparameter tuning is a very computationally intensive task, requiring re-training a model for dozens or hundreds of iterations. For this study where many unique models were investigated, we were not able to do a dense sweep of the entire parameter space for each model. Bayesian inference allows for an

Table 3 Training, validation, and testing datasets

	Training shots		Validation shots		Testing shots	
Device	Stable	Disruptive	Stable	Disruptive	Stable	Disruptive
C-Mod	3031	378	1011	126	1010	126
DIII-D	4122	328	1373	110	1374	110

efficient search, but in the end each model was limited to 300 or fewer hyperparameter tuning runs. It is possible that a better set of hyperparameters could have been found for each model, be it class time, horizon, learning rate, number of trees, etc. Despite this limitation, if a trend is seen to persist across several models we can reasonably determine that it was not due to inadequate hyperparameter tuning.

Bootstrap Results

Once the models were hyperparameter tuned, the metrics of interest were calculated for the test set using bootstrapping. Shots in the test set were randomly sampled with replacement for 50 iterations to obtain the median and interquartile range for the distribution of calculated metrics. This sampling was seeded so each model's bootstrapped metrics were calculated on the same random selections of shots. Bootstrap results of the AUROC scores are shown in Fig. 6 and results for AUWTC scores are shown in Fig. 7. Note that while AUWTC is not impacted by required warning time, we still calculated AUROC for all three required warning times on the models hyperparameter tuned for AUWTC. A full set of ROC and WTC curves are in appendices A and B for C-Mod and DIII-D respectively.

The most significant trend is that AUROC typically decreases as Δt_{req} increases. This is an expected result, as it should be more difficult to anticipate a disruption further in the future. In Fig. 6 there is one particular case where the AUROC score actually reverses this trend and increases with larger Δt_{req} in the RF tuned for AUROC on DIII-D, though this could be from the 50 ms AUROC model getting a bad set of hyperparameters.

A result from both AUROC (Fig. 6) and AUWTC (Fig. 7) is that predicting disruptions is far easier on DIII-D than on Alcator C-Mod. This is in agreement with previous studies [36], and may be due to a variety of factors. 25% of disruptions on C-Mod are thought to have been caused by sudden injections of molybdenum into the plasma (UFOs) [38]. UFO disruptions can happen in under 5 ms, which does not allow enough time to deploy DMS and is much shorter than the required warning times investigated here. On DIII-D, UFO disruptions are less frequent because it experiences lower heat flux to plasma facing components and has graphite as a wall material, which radiates less energy when introduced to the plasma

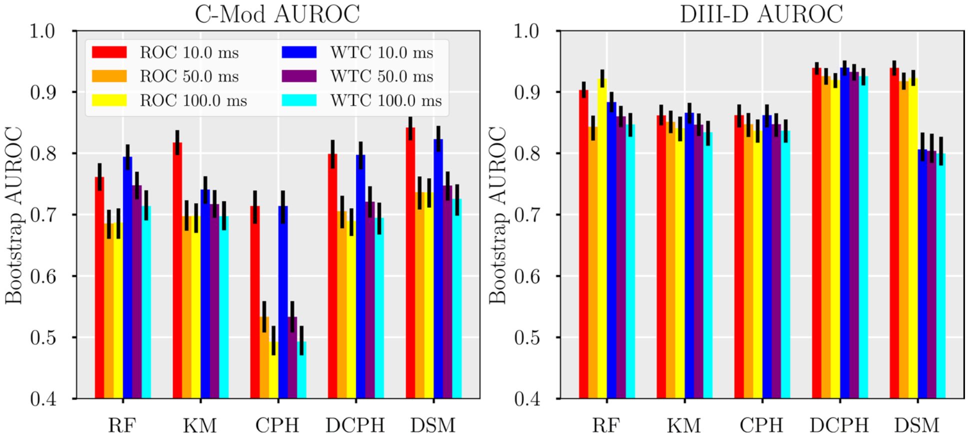


Fig. 6 Median and interquartile range of bootstrap AUROC metrics calculated on Alcator C-Mod (left) and DIII-D (right). Models are either hyperparameter tuned to maximize AUROC with Δt_{req} of 10 ms (red), 50 ms (orange), or 100 ms (yellow), or to maximize AUWTC (blue, purple, cyan). The AUROC metrics are then calculated for Δt_{req} of 10 ms (red, blue), 50 ms (orange, purple), or 100 ms (yellow, cyan)

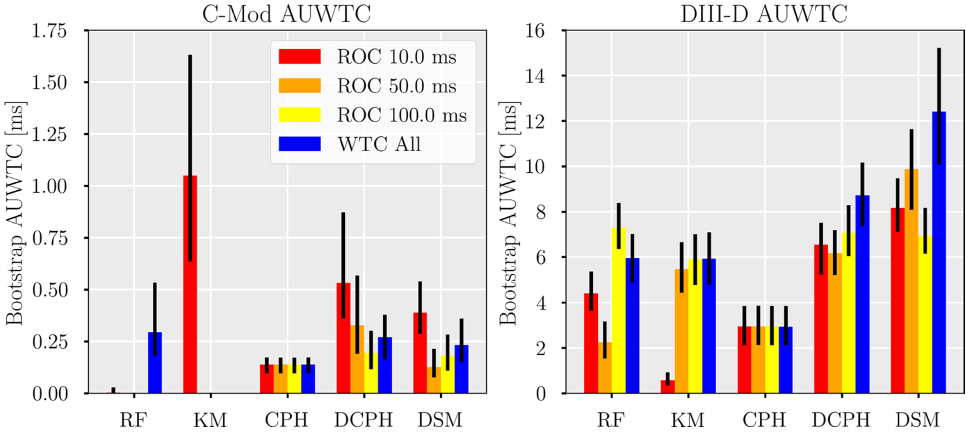


Fig. 7 Median and interquartile range of bootstrap AUWTC metrics calculated on Alcator C-Mod (left) and DIII-D (right). Models are either hyperparameter tuned to maximize AUROC with Δt_{req} of 10 ms (red), 50 ms (orange), or 100 ms (yellow), or to maximize AUWTC (blue). The AUWTC metrics are then calculated for each model

[7]. The difference in performance could also be due to the small size of C-Mod compared to DIII-D (0.7 m vs 1.7 m major radius) leading to faster plasma dynamics, making predictions with longer required warning times challenging. This can be seen in the typically larger difference in AUROC score as Δt_{req} is increased on C-Mod compared to DIII-D (Fig. 6).

Comparing AUROC results between models, it can be seen that RF, KM, DCPH, and DSM typically have similar scores. CPH on the other hand is generally the lowest performing model for both devices and metrics, with the median AUROC dropping below 0.5 in Alcator C-Mod for the longer required warning times (Fig. 6).

Comparing AUWTC results on DIII-D, RF and KM have similar values, while DCPH and DSM are generally higher. The DSM model specifically tuned to maximize AUWTC is by far the best, doubling the IQM warning time of RF.

Further looking at the results on DIII-D, we can see the pivotal role of metrics in designing disruption prediction algorithms. The chosen metric to hyperparameter tune the models for can greatly impact how well the model performs on that metric. This is an expected outcome, and it reinforces the need to identify metrics which are relevant to tokamak operations. This is most evident with the DSM model, where the models tuned for AUROC greatly outperform those tuned for AUWTC (Fig. 6) and vice versa (Fig. 7). Additionally on DIII-D, DCPH does not follow exactly the same pattern, and the instance of this model tuned for AUWTC has similar AUROC scores as those tuned for AUROC. However, the DCPH model tuned for AUWTC achieves higher AUWTC scores.

For the AUWTC metric, an outlier is how much KM tuned for 10 ms Δt_{req} outperforms the other models on C-Mod, which is an especially interesting result given that KM tuned for 10 ms Δt_{req} performs the lowest of all models on DIII-D (Fig. 7). The reason for this discrepancy is unclear; however, given the AUWTC values on C-Mod are significantly lower than on DIII-D combined with the large error bars in this case, this could simply be due to random variation.

Another interesting result is that the CPH models have nearly identical performance for both AUROC and AUWTC metrics regardless of which goal they were being tuned for. This is likely due to the CPH models only have two hyperparameters, learning rate and $\Delta t_{\mathrm{horizon}}$, so a similar optimal combination was found for each model.

Diagnostic Importance

Ideally, the signals from many diagnostics should be included in the training of data-driven models, though this may not always be feasible due to cost or technological complexity. To prioritize installation of diagnostics we must determine which are most relevant to making a prediction. However calculating input feature importance from a model that is already trained can be challenging because the output of a neural network is nonlinear with respect to the inputs. Even with simple linear models, some signals may only be useful when combined with others. For instance, one would need to know both the plasma current and the electron density to obtain the proximity to the Greenwald density limit. As such, to determine importance of diagnostics we chose to look at a single model re-trained on datasets where only subsets of all the signals were included.

The signals and corresponding diagnostics included in this study are shown in Table 2. Of these, the signals from magnetic probes are likely to always be available in real time on future tokamaks as they serve important functionalities such as controlling the plasma current and providing feedback for maintaining vertical stability. In addition, low-n modes are among the most prevalent disruptive precursors [39] and their onset can be detected with an array of magnetic probes. Equilibrium fitting utilizes data entirely from magnetic probes so it is conceivable that any device which has magnetics could include these signals as well, though not all devices may have the capability to perform fitting and relay it to the PCS in real time.

With these considerations, we investigated five cases. First with only signals from magnetic probes, as this should be the bare minimum. Second, with the addition of equilibrium fitting. The third and fourth cases then included either bolometry or interferometry, as these are separate diagnostic systems. Finally, we compared performance to the original instance of having signals from all diagnostics. The results of re-training the DSM model on Alcator C-Mod with the signals available from various sets of real-time diagnostics are shown in Fig. 8.

The results show that signals from magnetic probes account for almost all of the AUROC performance at short timescales. For longer timescales, much of the remaining performance is recovered with signals from equilibrium fitting, as there is a significant increase in the AUROC score for 100 ms Δt_{req}. For short timescales, it appears that there is a minor increase in performance when including bolometry, which could indicate measuring radiated power is necessary to detect fast impurity injections in C-Mod. An interesting result is that including interferometry does not appear to noticeably increase performance. This could mean the relevant information is already encapsulated in other signals or not many disruptions on Alcator C-Mod were caused by exceeding a density limit.

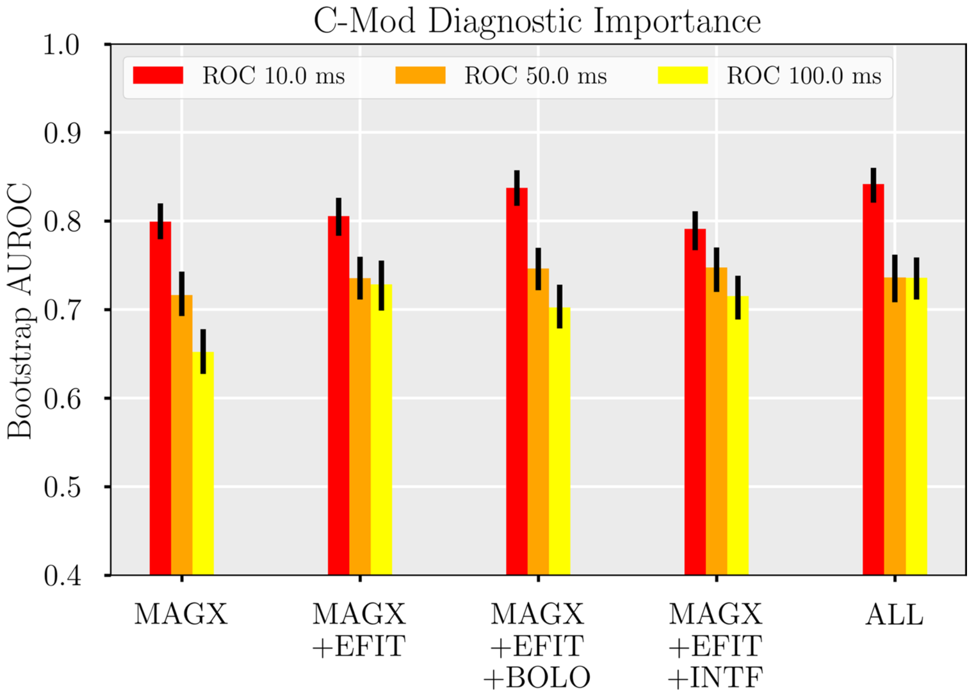


Fig. 8 Median and interquartile range of bootstrap AUROC metrics calculated on Alcator C-Mod when including various sets of diagnostics: magnetics (MAGX), equilibrium fitting (EFIT), bolometry (BOLO), interferometry (INTF), and all signals. Models are hyperparameter tuned to maximize AUROC with Δt_{req} of 10 ms (red), 50 ms (orange), or 100 ms (yellow)

Estimating Time-to-Disruption

The above sections discuss the usage of machine learning models as simple disruption predictors. That is, an alarm is triggered whenever the risk exceeds some threshold. As such, they have been hyperparameter tuned to use a single class time and/or horizon time. While triggering alarms in this way is useful, it lacks the information required to determine the best action to take to minimize damage from a disruption. The value we are most interested in is the expected time-to-disruption. Rigorously calculating this value would require obtaining the risks for an infinite number of future times. However, we are not interested in exceedingly long horizons, just those long enough for the PCS to actuate a response. Therefore, we can estimate the expected time-to-disruption for any feature vector $\vec{x}$ using a restricted mean survival time (RMST) [40] calculation

$$RMST(\vec{x}) = \int_0^T S(t|\vec{x})dt \tag{2}$$

where T is the maximum time horizon and $S(t|\vec{x}) \in [0, 1]$ is the survival function, or probability that no disruption occurs until after time t given some observed feature $\vec{x}$. This is essentially the expected value that no disruption will occur over the course of many time intervals in the domain $t \in [0, T]$ seconds. As a consequence, the longest predicted time-to-disruption is T seconds. This ignores the long horizons of the distribution, but if T large enough their contributions to the total expected value should be negligible. This maximum time is arbitrary and could be changed for future devices. In this study we chose $T = 1$ s since this is a very long timescale relative to PCS actions.

Calculating this survival function $S(t|\vec{x})$ depends on the type of model. Survival regression models include this function as a direct output where it may change nonlinearly in time. Binary classifiers and the conditional Kaplan-Meier formalism require making some assumptions on the evolution of risk in time, which will be discussed below.

As mentioned previously, the output of a binary classifier can be interpreted as the probability that some feature vector $\vec{x}$ is within the time window $\Delta\tau_{\text{class}}$ of a disruption. If we make the strong assumption that the distribution of disruptions is uniform within the class time, the probability of a disruption occurring over some arbitrary time interval $\Delta t < \Delta\tau_{\text{class}}$ is $P_D(\vec{x})\frac{\Delta t}{\Delta\tau_{\text{class}}}$, and similarly the probability that no disruption occurs over that same time interval is $(1 - P_D(\vec{x})\frac{\Delta t}{\Delta\tau_{\text{class}}})$. If we make another strong assumption that this survival function remains constant in time, we can take the product to find the probability that there is no disruption for n consecutive time steps. The survival function for an RF model is then

$$S(n\Delta t|\vec{x}) = \prod_{i=1}^{n} \left(1 - P_D(\vec{x})\frac{\Delta t}{\Delta\tau_{\text{class}}}\right) \tag{3}$$

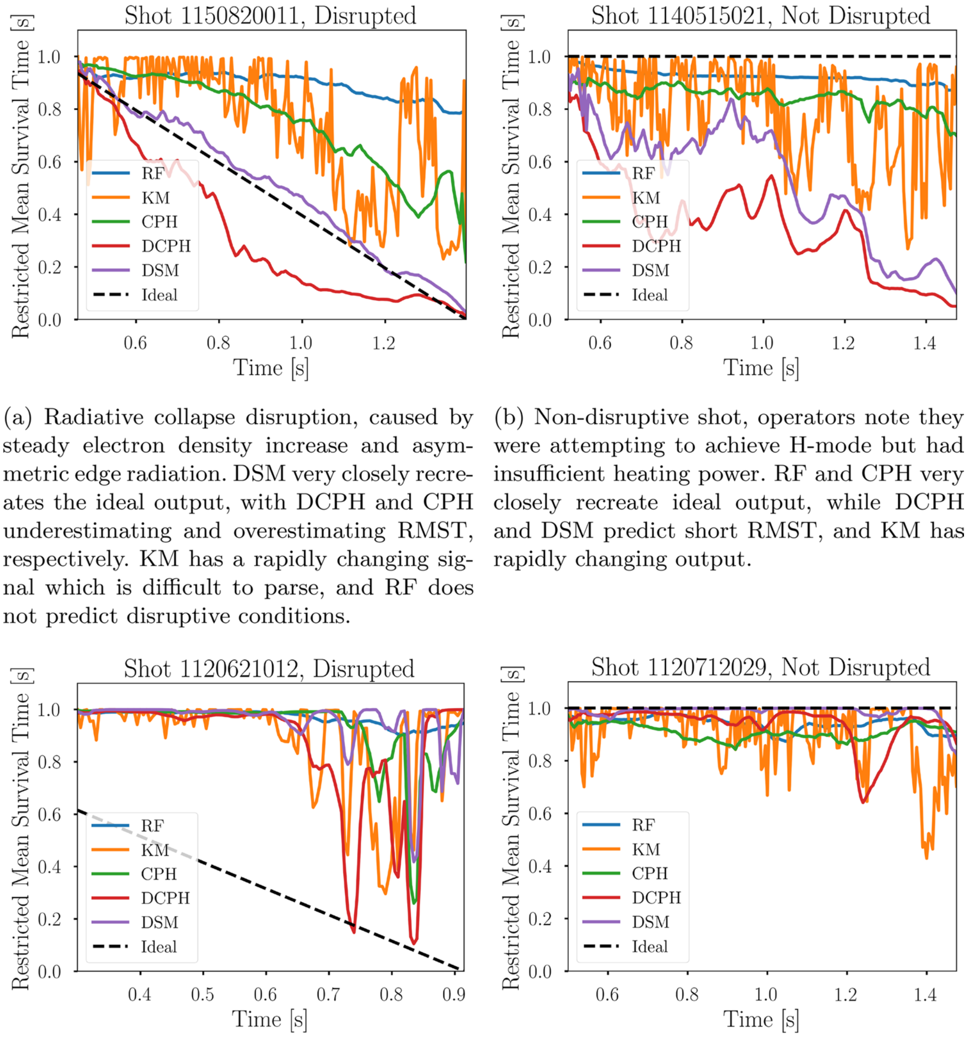
For the conditional Kaplan-Meier formalism, the RMST is calculated similarly [17]. The only difference being that instead of a constant value, the linear extrapolation allows the survival probability to change over time based on the calculated slope from the linear fit as shown in Eq. 1. The survival function for a KM model is then

$$S(n\Delta t|\vec{x}) = \prod_{i=1}^{n} \left[1 - \left(P_D(\vec{x}) + \frac{dP_D}{dt} i\Delta t \right) \frac{\Delta t}{\Delta \tau_{\text{class}}} \right] \tag{4}$$

where $(P_D(\vec{x}) + \frac{dP_D}{dt} i\Delta t)$ is limited to the interval [0, 1].

The RMST calculated using five disruption predictor models for four shots of Alcator C-Mod is shown below. A line of ideal prediction is also shown, which for non-disruptive shots is the maximum time T, and for disruptive shots is held at T until T seconds before a disruption takes place, at which point it decreases with a slope of -1 s/s. The particular models making these calculations have been hyperparameter tuned using a simplified metric which will be discussed shortly.

Fig. 9 RMST for two disruptive shots (**a**, **c**) and two stable shots (**b**, **d**) on Alcator C-Mod. An ideal output line is shown for reference (dotted black). The models shown are RF (blue), KM (yellow), CPH (green), DCPH (red) and DSM (purple) that have all been hyperparameter tuned to minimize the squared difference between calculated output and the ideal output for the last 1.5 s of flattop duration



(a) Radiative collapse disruption, caused by steady electron density increase and asymmetric edge radiation. DSM very closely recreates the ideal output, with DCPH and CPH underestimating and overestimating RMST, respectively. KM has a rapidly changing signal which is difficult to parse, and RF does not predict disruptive conditions.

(b) Non-disruptive shot, operators note they were attempting to achieve H-mode but had insufficient heating power. RF and CPH very closely recreate ideal output, while DCPH and DSM predict short RMST, and KM has rapidly changing output.

(c) Operators note that this shot encountered gap control issues starting at around 0.7 seconds, likely leading to the plasma contacting the wall and several impurity injections. KM, CPH, DCPH, and DSM all show rapid spikes towards the ideal line around 200 to 100 ms before disruption; however, they return to not predicting disruptive conditions when the disruption actually occurs. RF shows no significant change, always predicting a long RMST.

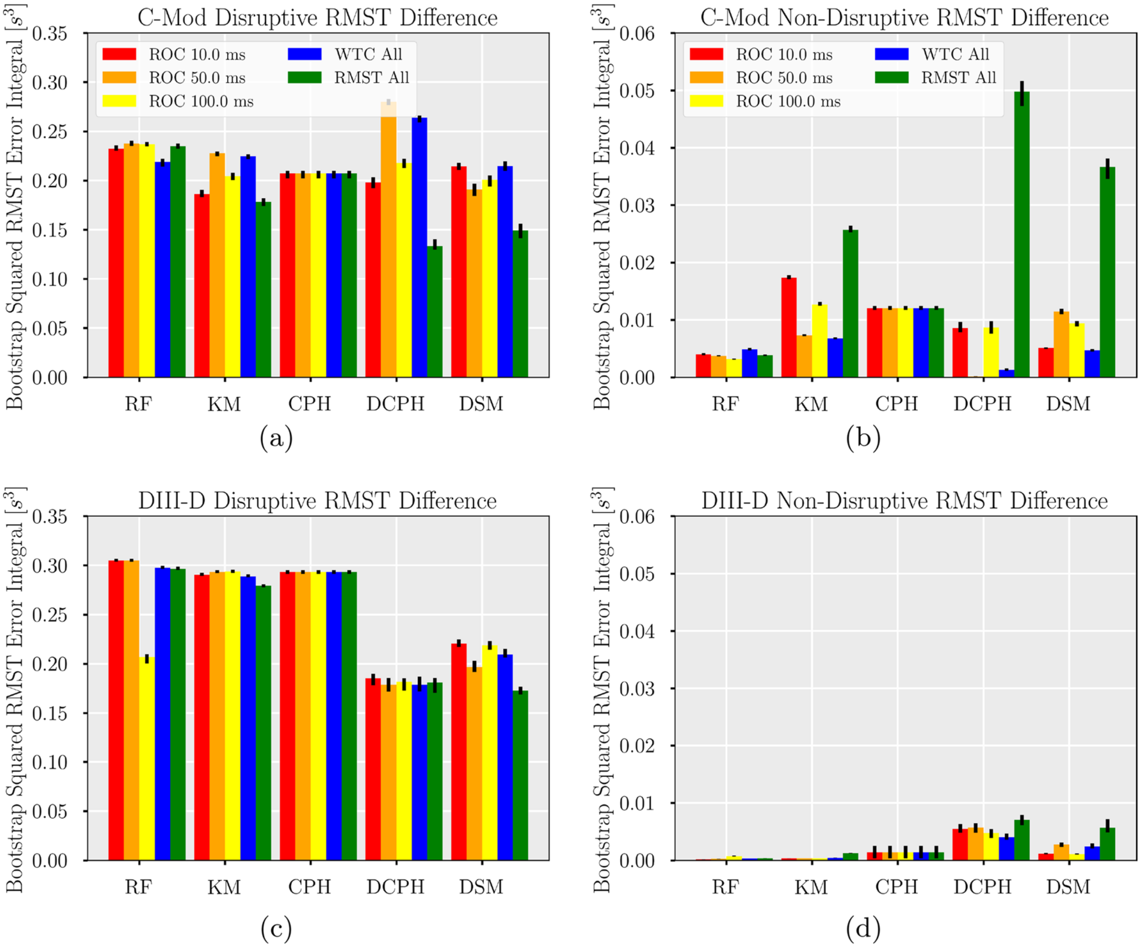
(d) Non-disruptive shot. Operators note that there was an impurity injection at 1.17-1.25 seconds, and DCPH model clearly decreases in RMST at that time. However, all models return to predicting long RMST when the flattop ends normally.

Encapsulating the quality of an RMST plot in a machine-relevant metric that can be used to evaluate models is challenging. Using the models to calculate an RMST value is considerably more computationally expensive than using the models as disruption predictors, since the survival function must be evaluated at many time horizons instead of just one. Another consideration is that the RMST output by a model should be close to the ideal value while not changing too quickly, as rapid fluctuations in the output would greatly interfere with the PCS being able to plan actuator responses. This behavior is most obvious with the KM model, but it is also the case with CPH and DCPH shown in Fig. 9c.

For this study, we have defined a simplified metric where the squared difference between the ideal value and the output RMST is calculated for each time slice and integrated over the final 1.5 s of the shot. This time domain restriction was done to reduce the computation required to calculate the metric. Note that this metric does not penalize the output for changing rapidly; however, it is unclear what the tradeoff between line smoothness and closeness to the ideal value should be. Future work should aim to design a computationally tractable metric that describes the quality of these expected time-to-disruption plots that would be useful for a tokamak operator or real time in the PCS.

Bootstrapped results of this simple metric for disruptive and non-disruptive shots in the test set are shown in Fig. 10. Models were hyperparameter tuned to minimize the simplified metric output for only disruptive shots. This choice was again to reduce the computation required, as there are significantly fewer disruptive shots than non-disruptive shots in both datasets.

For non-disruptive shots, the RMST calculated by all models typically matches the ideal value fairly well. While neglecting the non-disruptive shots appears to increase the error of the KM, DCPH, and DSM models on Alcator C-Mod as shown in Fig. 10b, the errors are very small



(a)

(b)

(c)

(d)

Fig. 10 Median and interquartile range of bootstrap RMST difference integrals for Alcator C-Mod (**a**, **b**) and DIII-D (**c**, **d**) across all disruptive shots (**a**, **c**) and non-disruptive shots (**b**, **d**) of the test set. Models are either hyperparameter tuned to maximize AUROC with Δt_{req} of 10 ms (red), 50 ms (orange), or 100 ms (yellow), to maximize AUWTC (blue), or to minimize the squared difference between calculated output and the ideal output for the last 1.5 s of flattop duration (green)

compared to the disruptive shots as shown in Fig. 10a. However, there are still instances like Fig. 9b where the model's calculated RMST suggests a disruption is oncoming even when none eventually occurred.

For disruptive shots, the calculated RMST is at times almost perfect, as with the DSM shown in Fig. 9a. However, there are many instances where the RMST remains unchanged even up to the time of disruption. In addition, the RMST output may fluctuate wildly. This mismatch in predicted RMST versus ideal RMST for disruptive shots is also evident in the test set results shown in Fig. 10a and c. On C-Mod, the DCPH and DSM models trained to minimize RMST error have the best performance; however, in all cases this is worse than for non-disruptive shots. On DIII-D, the models also perform worse than on non-disruptive shots. Also on DIII-D, there is an interesting result where the RF hyperparameter tuned for 100 ms AUROC performed far better than the rest of the RF models. Otherwise, the DCPH and DSM models tuned for all metrics showed greater performance than RF, KM, and CPH, with DSM hyperparameter tuned to minimize RMST error performing the best.

Summary

We have evaluated the performance of several data-driven algorithms in their usage as disruption predictors on experimental data from Alcator C-Mod and DIII-D. Our methods require a large amount of existing data and do not take into account transferring models to new devices. This is an important consideration for future work, but in the present study we can compare performance between survival analysis and previously studied models. Our findings indicate that on Alcator C-Mod, the AUROC scores of RF, KM, DCPH, and DSM are similar, and CPH scores lower. A major component of the resulting AUROC scores for C-Mod is the required warning time, where making predictions further in the future is challenging. We find that the AUROC scores of all models on DIII-D are higher than on C-Mod, and there is little difference between the models trained for each required warning time investigated. Also on DIII-D, we show that DCPH and DSM have improved ability to predict disruptions with extended time horizons in the low FPR regime. While DCPH shows this behavior when being hyperparameter tuned for either metric, the AUROC score of DSM decreases when hyperparameter tuned for AUWTC. We determine that identifying machine-relevant metrics is a major component of designing disruption predictors.

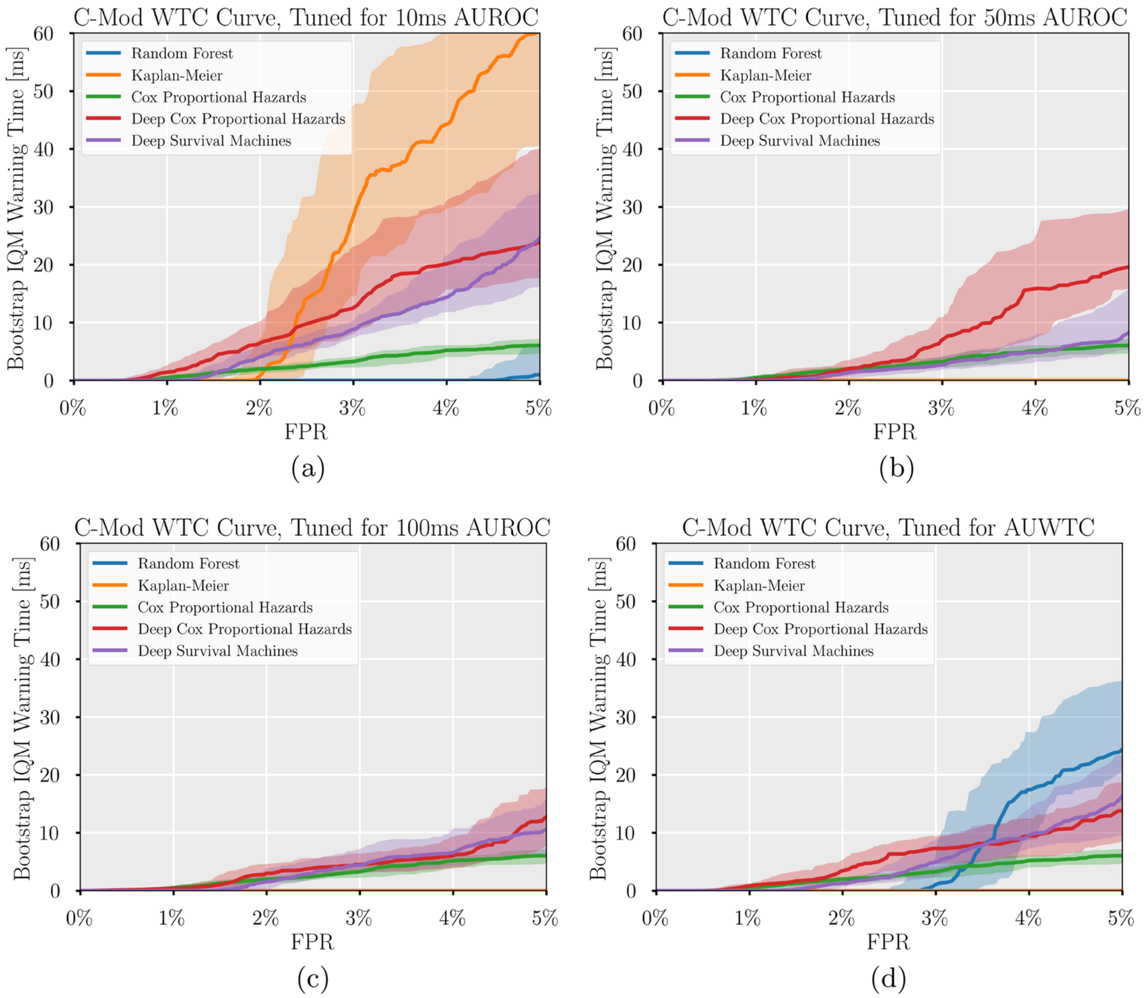
We also attempted to predict the time-to-disruption using RMST, and provided the results of this calculation using a simple difference metric over the test sets for both devices, as well as an excerpt of the produced values for four shots on Alcator C-Mod. We demonstrate that while this calculation is generally close to ideal in non-disruptive shots, there are potentially rapid fluctuations in RMST output. In addition, there are many instances where the calculated RMST output is not a good indicator of disruptive conditions. Despite these challenges, there are cases where the desired RMST output is produced for disruptive shots. We determine that developing a metric which evaluates the quality of expected time-to-disruption predictions relevant to tokamak operation should be an essential goal of future work in this area.

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.

Appendix A Alcator C-Mod Results

See Figs. 11 and 12.



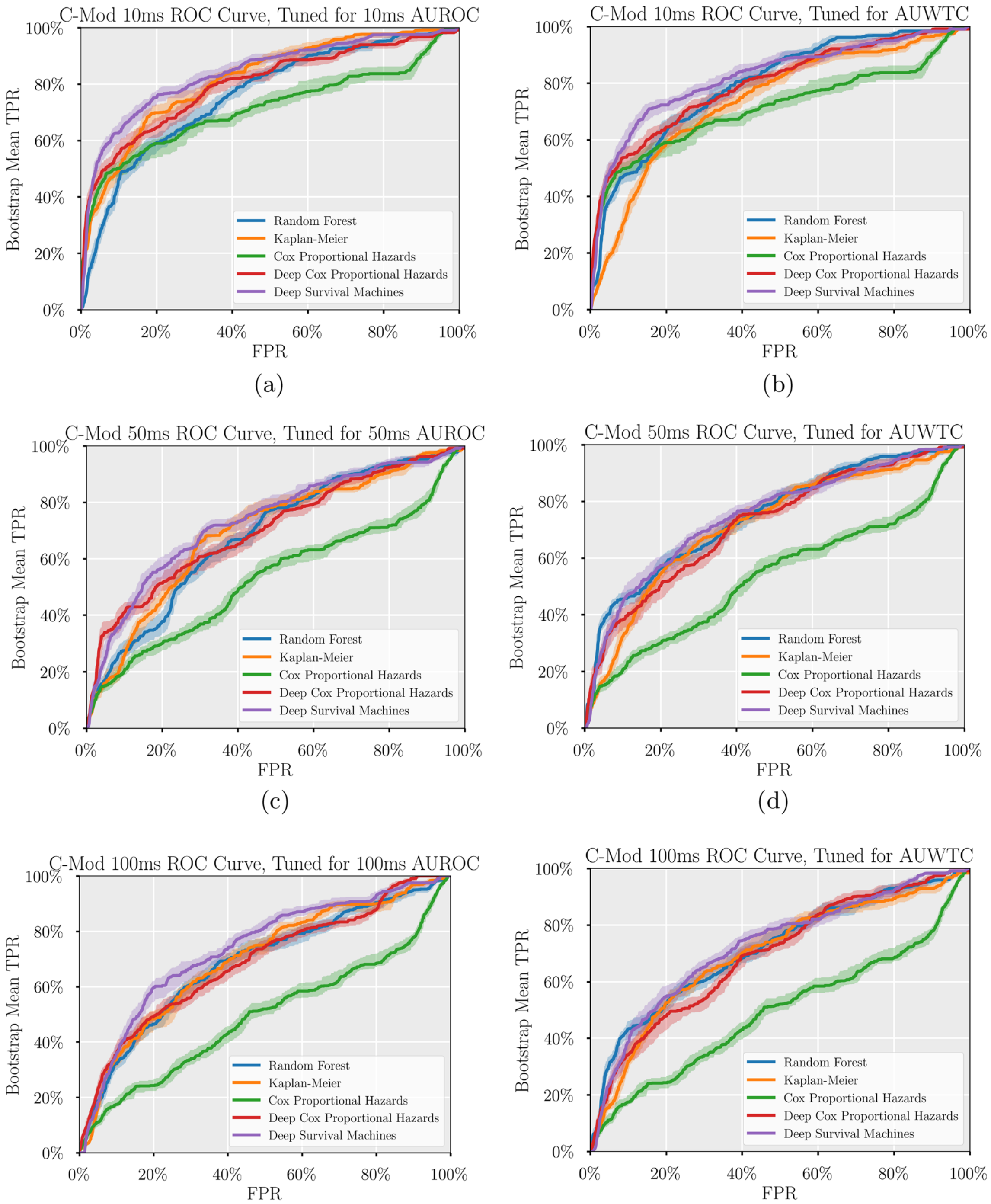
(a)

(b)

(c)

(d)

Fig. 11 Alcator C-Mod bootstrapped WTC curves for models hyperparameter tuned to maximize AUROC with a particular required warning time (**a**, **b**, **c**), or to maximize AUWTC (**d**). The median and interquartile range are represented as a solid line and shaded region, respectively



(a) (b) (c) (d) (e) (f)

Fig. 12 Alcator C-Mod bootstrapped ROC curves for models hyperparameter tuned to maximize AUROC with a particular required warning time (**a**, **c**, **e**), or to maximize AUWTC (**b**, **d**, **f**). The median and interquartile range are represented as a solid line and shaded region, respectively

Appendix B DIII-D Results

See Figs. 13 and 14

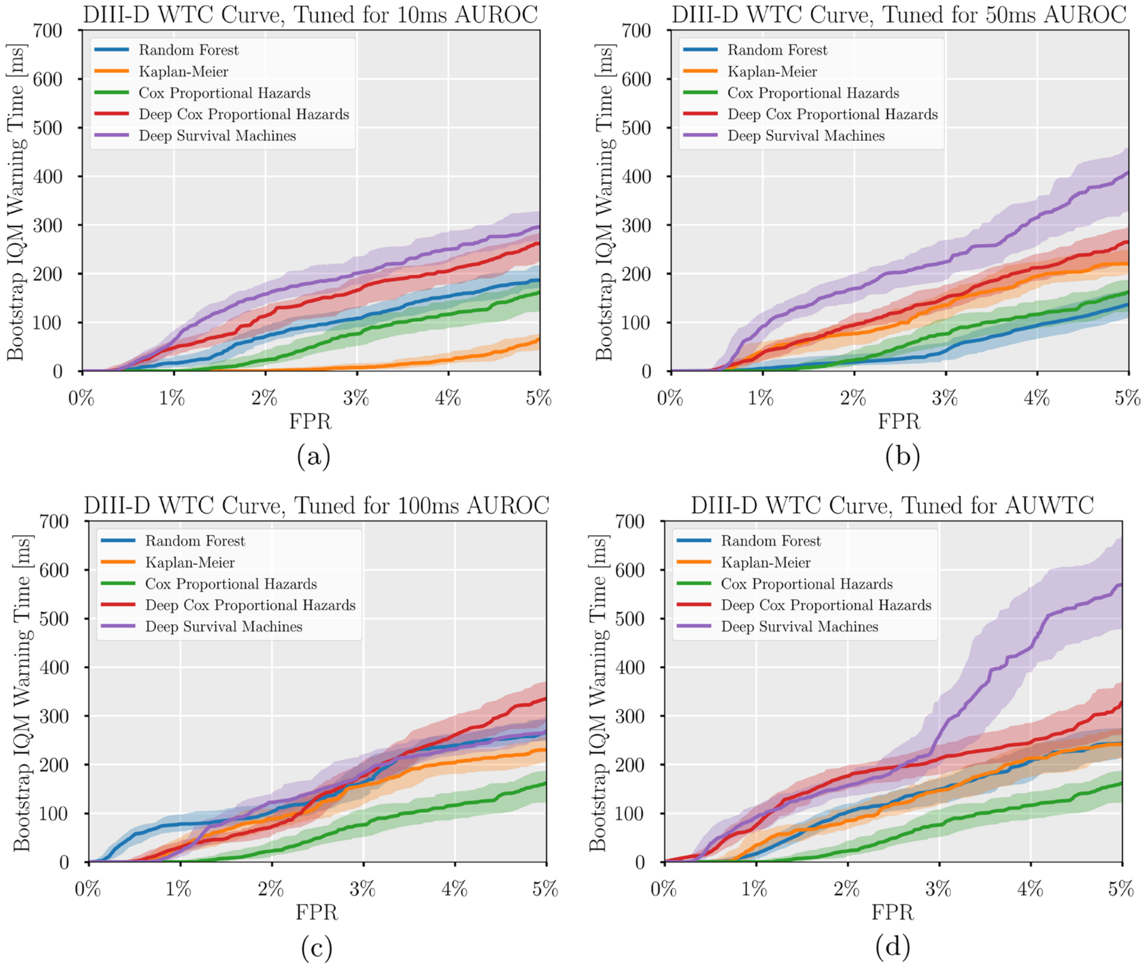
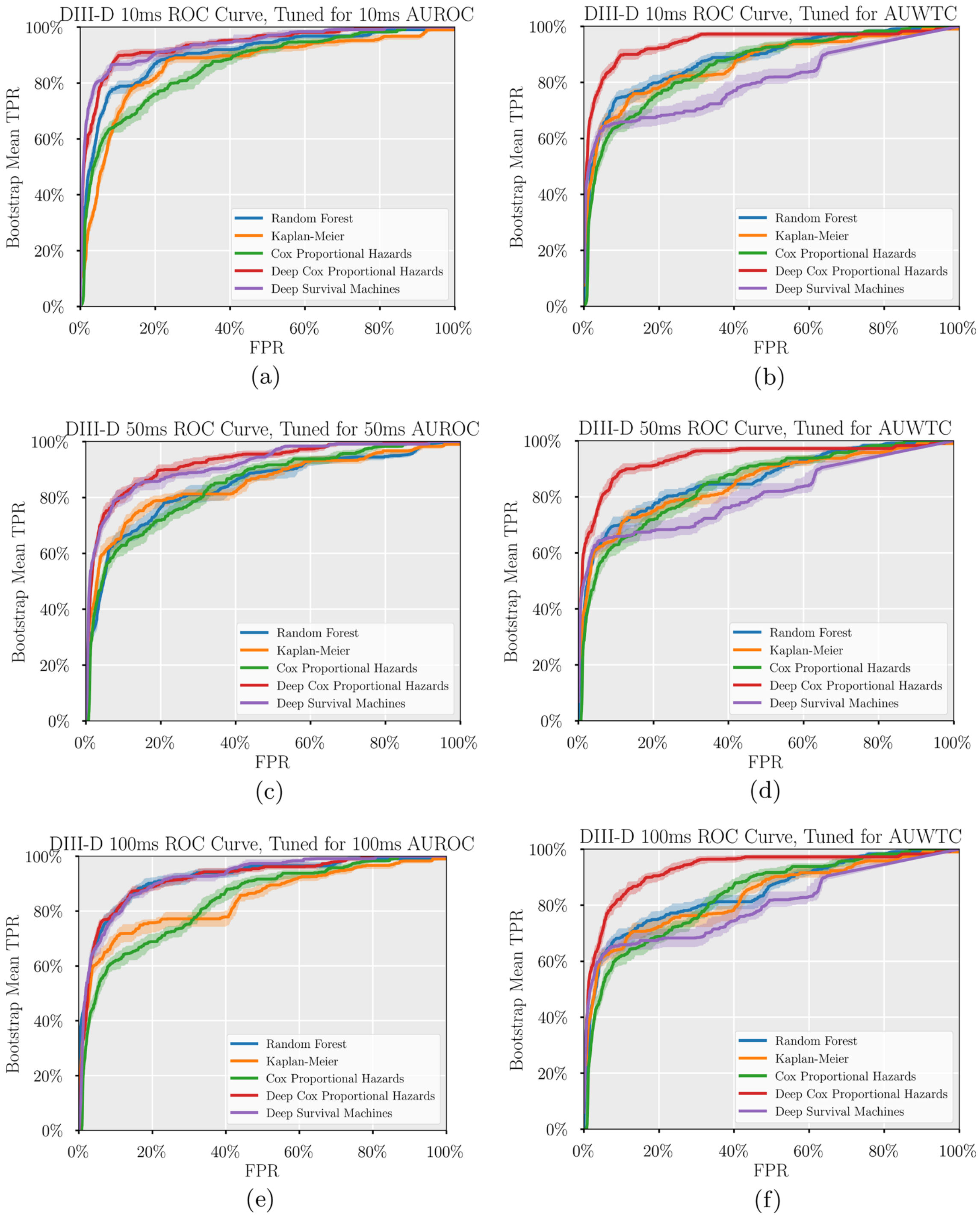


Fig. 13 DIII-D bootstrapped WTC curves for models hyperparameter tuned to maximize AUROC with a particular required warning time (**a**, **b**, **c**), or to maximize AUWTC (**d**). The median and interquartile range are represented as a solid line and shaded region, respectively



DIII-D 10ms ROC Curve, Tuned for 10ms AUROC

(a)

DIII-D 10ms ROC Curve, Tuned for AUWTC

(b)

DIII-D 50ms ROC Curve, Tuned for 50ms AUROC

(c)

DIII-D 50ms ROC Curve, Tuned for AUWTC

(d)

DIII-D 100ms ROC Curve, Tuned for 100ms AUROC

(e)

DIII-D 100ms ROC Curve, Tuned for AUWTC

(f)

Fig. 14 DIII-D bootstrapped ROC curves for models hyperparameter tuned to maximize AUROC with a particular required warning time (**a**, **c**, **e**), or to maximize AUWTC (**b**, **d**, **f**). The median and interquartile range are represented as a solid line and shaded region, respectively

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10894-024-00413-y.

Acknowledgements Framework development and computational resources at MIT were supported by Commonwealth Fusion Systems.

Data curation for Alcator C-Mod and DIII-D was supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, under Award DE-SC0014264.

Acquisition of DIII-D data was 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 Award(s) DE-FC02-04ER54698.

Funding 'Open Access funding provided by the MIT Libraries'.

References

1. A.J. Creely et al., SPARC as a platform to advance tokamak science. Physics of Plasmas **30**(9), 090601 (2023). https://doi.org/10.1063/5.0162457
2. M. Shimada et al., Chapter 1: Overview and summary. *Nuclear Fus.***47**(6), S1 (2007). https://doi.org/10.1088/0029-5515/47/6/S01
3. F.C. Schuller, Disruptions in tokamaks. Plasma Phys. Controlled Fus. **37**(11A), A135 (1995). https://doi.org/10.1088/0741-3335/37/11A/009
4. D.G. Whyte, D.A. Humphreys, A.G. Kellman, Disruption physics and mitigation on DIII-D. *Fus. Sci. Technol.***48**(2), 954–967 (2005). https://doi.org/10.13182/FST05-A1051
5. A.J. Creely et al., Overview of the SPARC tokamak. J. Plasma Phys. **86**(5), 865860502 (2020). https://doi.org/10.1017/S0022377820001257
6. V. Riccardo, *Disruption Loads in SPARC*. en. Invited Oral. ITER Headquarters (2022)
7. C. Rea et al., Disruption prediction investigations using machine learning tools on DIII-D and Alcator C-Mod. Plasma Phys. Controll. Fus. **60**(8), 084004 (2018). https://doi.org/10.1088/1361-6587/aac7fe
8. B. Cannas et al., Disruption prediction with adaptive neural networks for ASDEX upgrade. In *Fusion Engineering and Design. Proceedings of the 26th Symposium of Fusion Technology (SOFT-26)***86**(6), 1039–1044 (2011). ISSN: 0920-3796. https://doi.org/10.1016/j.fusengdes.2011.01.069. https://www.sciencedirect.com/science/article/pii/S0920379611000810
9. C. Rea et al., A real-time machine learning-based disruption predictor in DIII-D. Nuclear Fus. **59**(9), 096016 (2019). https://doi.org/10.1088/1741-4326/ab28bf
10. E. Aymerich et al., Disruption prediction at JET through deep convolutional neural networks using spatiotemporal information from plasma profiles. Nuclear Fus. **62**(6), 066005 (2022). https://doi.org/10.1088/1741-4326/ac525e
11. B. H. Guo et al., "Disruption prediction on EAST tokamak using a deep learning algorithm". *Plasma Physi. Controll. Fus.***63**(11) (2021). Publisher: IOP Publishing, p. 115007. ISSN: 0741-3335. https://doi.org/10.1088/1361-6587/ac228b
12. J. X. Zhu et al., "Hybrid deep-learning architecture for general disruption prediction across multiple tokamaks". en. In *Nuclear Fus.***61**(2) (2020). p. 026007. https://doi.org/10.1088/1741-4326/abc664
13. E.J. Strait et al., Progress in disruption prevention for ITER. Nuclear Fus. **59**(11), 112012 (2019). https://doi.org/10.1088/1741-4326/ab15de
14. Andrew D. Maris et al., The impact of disruptions on the economics of a tokamak power plant. Fus. Sci. Technol. (2023). https://doi.org/10.1080/15361055.2023.2229675
15. X. Zhu, J. Yao, J. Huang, Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Shenzhen, China: IEEE (2016), pp. 544–547. ISBN: 978-1-5090-1611-2. https://doi.org/10.1109/BIBM.2016.7822579. http://ieeexplore.ieee.org/document/7822579/
16. J. B. Sabuhoro, Y. Gervais, Factors determining the success or failure of Canadian establishments on foreign markets: a survival analysis approach. *Int. Trade J.***20** (2004). https://doi.org/10.1080/08853900500467974
17. R.A. Tinguely et al., An application of survival analysis to disruption prediction via Random Forests. Plasma Phys. Controlled Fusion **61**(9), 095009 (2019). https://doi.org/10.1088/1361-6587/ab32fc
18. K.E.J. Olofsson, D.A. Humphreys, R.J. La Haye, "Event hazard function learning and survival analysis for tearing mode onset characterization. Plasma Phys. Controlled Fusion **60**(8), 084002 (2018). https://doi.org/10.1088/1361-6587/aac662
19. C. Nagpal, W. Potosnak, A. Dubrawski, Auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data. *Proc. Mach. Learn. Res.***182** (2022)
20. M. Greenwald et al., 20 years of research on the Alcator C-Mod tokamak). Phys. Plasmas **21**(11), 110501 (2014). https://doi.org/10.1063/1.4901920
21. M.E. Fenstermacher et al., DIII-D research advancing the physics basis for optimizing the tokamak approach to fusion energy. Nuclear Fusion **62**(4), 042024 (2022). https://doi.org/10.1088/1741-4326/ac2ff2
22. L. Breiman, Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
23. E. L. Kaplan, P. Meier, Nonparametric estimation from incomplete observations". *J. Am. Stat. Assoc.***53**(282) (June 1958). Publisher: Taylor & Francis _eprint:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1958.10501452, pp. 457–481. ISSN: 0162-1459. https://doi.org/10.1080/01621459.1958.10501452. https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452
24. D.R. Cox, Regression models and life-tables''. J. Royal Stat. Soc. Seri. B (Methodol.) **34**(2), 187–202 (1972). https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
25. C. Nagpal, X. R. Li, A. Dubrawski, Deep survival machines: fully parametric survival regression and representation learning for censored data with competing risks (2021). arXiv:2003.01176 [cs, stat]. https://doi.org/10.48550/arXiv.2003.01176
26. R. Sweeney et al., Relationship between locked modes and thermal quenches in DIII-D. Nuclear Fus. **58**(5), 056022 (2018). https://doi.org/10.1088/1741-4326/aaaf0a
27. D. Faraggi, R. Simon, A neural network model for survival data. Stat. Med. **14**(1), 73–82 (1995). https://doi.org/10.1002/sim.4780140108
28. C.G. Windsor et al., A cross-tokamak neural network disruption predictor for the JET and ASDEX Upgrade tokamaks. Nuclear

Fus. **45**(5), 337 (2005). https://doi.org/10.1088/0029-5515/45/5/004

29. K. Rath et al., Data augmentation for disruption prediction via robust surrogate models. J. Plasma Phys. **88**(5), 895880502 (2022). https://doi.org/10.1017/S0022377822000769
30. B. Cannas et al., An adaptive real-time disruption predictor for ASDEX Upgrade. Nuclear Fus. **50**(7), 075004 (2010). https://doi.org/10.1088/0029-5515/50/7/075004
31. A. Murari et al., Adaptive learning for disruption prediction in non-stationary conditions. Nuclear Fus. **59**(8), 086037 (2019). https://doi.org/10.1088/1741-4326/ab1ecc
32. A. Murari et al., On the transfer of adaptive predictors between different devices for both mitigation and prevention of disruptions. Nuclear Fus. **60**(5), 056003 (2020). https://doi.org/10.1088/1741-4326/ab77a6
33. P. C. de Vries et al., Requirements for triggering the ITER disruption mitigation system. *Fus. Sci. Technol.***69**(2) (2016), pp. 471–48. https://doi.org/10.13182/FST15-1764
34. X. Jiang, G. F. Cooper, D. B. Neill, Generalized AMOC curves for evaluation and improvement of event surveillance. In: AMIA ... Annual Symposium proceedings. AMIA Symposium 2009, 281–285 (2009). ISSN: 1942-597X
35. K.J. Montes et al., Machine learning for disruption warnings on Alcator C-Mod, DIII-D, and EAST. Nuclear Fus. **59**(9), 096015 (2019). https://doi.org/10.1088/1741-4326/ab1df4
36. J. Zhu, Data-driven study of major disruption prediction and plasma instabilities across multiple tokamaks. PhD thesis. Massachusetts Institute of Technology (2022)
37. T. Akiba et al., Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, pp. 2623–2631 (2019). ISBN: 978-1-4503-6201-6. https://doi.org/10.1145/3292500.3330701
38. R. Granetz et al., Tungsten and disruptions in SPARC. In APS Division of Plasma Physics Meeting Abstracts ADS Bibcode: 2021APS..DPPJO7006G. (Jan. 2021). https://ui.adsabs.harvard.edu/abs/2021APS..DPPJO7006G
39. P.C. de Vries et al., Survey of disruption causes at JET. Nuclear Fus. **51**(5), 053018 (2011). https://doi.org/10.1088/0029-5515/51/5/053018
40. K. Han, I. Jung, Restricted mean survival time for survival analysis: a quick guide for clinical researchers. Korean J. Radiol. **23**(5), 495–499 (2022). https://doi.org/10.3348/kjr.2022.0061