Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Horizon-Aware Early Event Prediction for Tokamak Disruption Alarms

Takeshi Koshizuka

takeshikoshizuka938444@gmail.com

RIKEN

Takaharu Yaguchi

Kyushu University

Research Article

Keywords: tokamak disruption prediction, early event prediction, survival analysis, temporal label smoothing, alarm evaluation

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11399031/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Horizon-Aware Early Event Prediction for Tokamak Disruption Alarms

Takeshi Koshizuka[1,2*] and Takaharu Yaguchi[1,2]

[1*]RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan.
[2]Institute of Mathematics for Industry (IMI), Kyushu University, Fukuoka, Japan.

*Corresponding author(s). E-mail(s): takeshi.koshizuka@riken.jp;
Contributing authors: yaguchi@imi.kyushu-u.ac.jp;

Abstract

Disruption predictors must warn the plasma control system early enough for the intended response, and the required warning time differs between mitigation and avoidance. We view a warning system as a predictor that outputs a disruption risk and an alarm policy acting on it, and ask which risk the predictor should learn. Binary classifiers do not specify when a disruption occurs, and survival analysis models the full distribution of the time to disruption, whereas an alarm depends only on the risk within a finite horizon. We introduce two horizon-focused targets into tokamak disruption prediction: a horizon-truncated survival objective and fixed-horizon early event prediction (EEP) with temporal label smoothing (TLS). We compare them with binary classification and survival analysis on the DIII-D, Alcator C-Mod, and EAST databases of DisruptionBench, evaluating the risk by its threshold-free ranking at a 40 ms deadline and the alarm policy by thresholds calibrated on validation data. On every device, TLS outperforms the deep survival models in both ranking and alarm performance; TLS performs best on DIII-D and EAST, and a random forest trained on a hard horizon target performs best on C-Mod. TLS also outperforms hard targets placed at either end of its smoothing range in alarm performance on all devices, isolating the benefit of label smoothing. Horizon truncation alone did not consistently improve survival analysis. Alarm performance remains limited on C-Mod and DIII-D and varies strongly across training seeds. Ablations show that the target should extend to the required warning time.

Keywords: tokamak disruption prediction, early event prediction, survival analysis, temporal label smoothing, alarm evaluation

1 Introduction

Disruptions remain one of the main threats to the safe operation of future tokamaks. A disruption can impose severe thermal and electromagnetic loads on plasma-facing and structural components, and the plasma control system (PCS) must therefore be warned early enough to select and execute a response [1, 2]. The required warning time depends on that response: triggering the disruption mitigation system requires only a short lead time, whereas avoidance actions such as a controlled ramp-down require substantially more [2].

A data-driven disruption warning system can be viewed as two modules. A predictor maps the measurements available at each time to a

disruption risk, a quantity with a defined meaning, and an alarm policy decides from this risk when to raise an alarm, for example by thresholding it with hysteresis [1, 3]. An alarm policy can be designed and calibrated for an intended response only if the risk is defined in the terms that this response requires. The central question is therefore which quantity the predictor should be trained to output.

Most data-driven disruption predictors are binary classifiers trained to recognize a pre-disruptive state, defined by the outcome of the discharge or by a class time before the disruption [1, 3–5]. Their output indicates whether the plasma appears to be heading towards a disruption, but not when the disruption will occur. The class time is usually chosen to separate stable and pre-disruptive phases rather than from the lead time of a control action, and a single class time does not match the precursor timescales of all disruption mechanisms [2]. To select and time a response, the PCS instead needs the risk that a disruption occurs within the time required by the intended action. Survival analysis models this risk directly: by modeling the conditional distribution of the time to disruption while accounting for right censoring [6], it provides the disruption risk within any horizon from a single model [2, 7]. Keith et al. [2] formulated risk-aware disruption prediction in this framework and applied deep survival models, including deep Cox proportional hazards (DCPH) [8] and Deep Survival Machines (DSM) [9], to Alcator C-Mod and DIII-D.

Whether such a risk can be learned reliably from data is a separate question. A survival model fits the distribution of the residual time over its full range. In long discharges, most landmarks lie far from the disruption, where precursor signals are weak, so much of the likelihood concerns event times well beyond the horizon relevant to an alarm, although the alarm decision depends only on the risk within that horizon. One remedy keeps the survival model but truncates its objective at the horizon, as in the horizon-truncated dynamic survival analysis of Yèche et al. [10], which we refer to as survTLS. A further step gives up modeling the time to disruption and learns only the risk within the horizon. This fixed-horizon early event prediction (EEP), developed for clinical early-warning systems, estimates at every time step whether the event will occur within a prescribed horizon [11–13]. Its hard target, however, switches from negative to positive at a single residual time, although precursor signals do not change abruptly there, and only the last part of each disruptive discharge is positive, so positive examples are scarce. Temporal label smoothing (TLS) [13] addresses both issues by replacing the abrupt boundary with a target that decreases smoothly with the time to the event. To our knowledge, neither formulation has been examined for tokamak disruption prediction.

Which of these risks best serves an alarm policy has not been established for disruption prediction. In this work, we therefore compare predictors trained on binary classification (BC), full-distribution survival analysis (SA), horizon-truncated survival analysis, and fixed-horizon EEP with hard and smoothed targets, on data from three tokamaks.

The two-module view also determines how the predictors should be evaluated. A risk is useful only if it separates discharges that will disrupt from those that will not early enough for the intended action, and this property does not depend on the threshold that a particular policy applies. We therefore evaluate the risk by its threshold-free ranking of disruptive and non-disruptive discharges using only the information available before a required deadline (Deadline AUPRC). We use the area under the precision–recall curve because disruptive discharges are a minority, so that the area under the receiver operating characteristic curve can be uninformative [14], and we keep detection and warning time separate rather than combining them into a single quantity such as the area under the warning-time characteristic curve [2]. In operation, however, the alarm policy must fix its threshold in advance, and what matters is whether disruptions are detected in time while false alarms during normal operation remain rare. We therefore also evaluate alarm policies whose thresholds are calibrated on validation data, by the Deadline F1 and the false-alarm rate per 24 hours of normal operation on unseen test discharges. Separating the two evaluations reveals whether a difference in alarm performance stems from the risk itself or from the calibration of its threshold.

The main contributions of this work are as follows:

1. We introduce two horizon-focused prediction targets, horizon-truncated survival analysis and fixed-horizon EEP with TLS, into tokamak disruption prediction, and we compare them systematically with BC and SA under a common protocol on three tokamaks.
2. We isolate the effect of temporal label smoothing by comparing TLS with hard fixed-horizon targets placed at either end of its smoothing range, with the training data, encoder, and optimization fixed.
3. We propose an evaluation protocol that separates the risk, through its threshold-free ranking at an operational deadline, from the alarm policy, through the performance of validation-calibrated thresholds.

2 Related Work

2.1 Machine Learning for Disruption Detection

Machine-learning-based disruption prediction has evolved from device-specific classifiers built on manually selected plasma quantities to deep sequence models and, more recently, standardized multi-machine evaluation. Early studies demonstrated that disruption precursors could be inferred from diagnostic signals using cross-tokamak neural networks, support vector machines, fuzzy-logic and regression-tree systems, and unsupervised or manifold-based representations [15–19]. Despite their architectural diversity, most of these methods formulated disruption prediction as fixed-window binary classification.

Most of these predictors label each time slice of a disruptive shot as positive when it lies within a prescribed class time Δ_{class} of the disruption, and raise an alarm when the predicted probability satisfies a threshold or hysteresis rule. This formulation imposes an abrupt, user-defined boundary between non-disruptive and pre-disruptive states. However, the relevant precursor timescale depends on the underlying disruption mechanism. For example, locked-mode precursors may emerge over tens of milliseconds, whereas impurity accumulation can evolve over hundreds of milliseconds. Consequently, a single Δ_{class} cannot capture the temporal structure of all disruption processes, nor does it necessarily coincide with their physical onset [2]. We relate this formulation to the prediction targets compared in this study in section 3.2.

Random-forest predictors trained on large DIII-D and Alcator C-Mod databases were subsequently developed and deployed in real-time operation [1, 20]. Later research largely replaced manually engineered features with recurrent neural networks (RNNs), temporal convolutional networks (TCNs), spatiotemporal convolutional models, and Transformers that learn temporal representations directly from diagnostic signals [4, 5, 21, 22]. DisruptionBench compared a random forest, the hybrid deep learner, a Transformer, and a continuous convolutional neural network (CCNN) [23] across Alcator C-Mod, DIII-D, and EAST under common data splits and alarm logic [3]. Long-context studies have shown that predictive information is not limited to the immediate pre-disruption interval but is also present much earlier in the plasma evolution, motivating the use of sequence encoders capable of preserving long-range dependencies [3].

More recently, disruption prediction has increasingly been formulated within the framework of survival analysis [2, 7, 24]. A key advantage of this framework is its principled treatment of censored observations and its ability to explicitly model the time remaining until disruption. Olofsson et al. [24] modeled the hazard function for tearing-mode onset, while Tinguely et al. [7] combined random-forest predictions with conditional Kaplan–Meier estimation to obtain horizon-dependent disruption risks. More recently, Keith et al. [2] compared conditional Kaplan–Meier estimation with Cox proportional hazards, Deep Cox proportional hazards, and DSM.

These survival-based studies model the time to disruption over its full range. In parallel, clinical early-warning research has developed survival and fixed-horizon formulations that focus on the time window relevant to an alarm, as reviewed below; their potential for disruption prediction has not yet been assessed.

2.2 EEP

Survival analysis models the time to an event while accounting for right censoring. Classical approaches include the Kaplan–Meier estimator and Cox proportional hazards, and later work introduced nonlinear, discrete-time, and fully

parametric deep survival models [8, 9, 25–30]. Dynamic prediction updates these estimates as longitudinal measurements accumulate, either by landmarking [31] or with recurrent deep models [32].

EEP developed largely in clinical early-warning systems, where multivariate time series from electronic health records are used to predict adverse events, such as acute kidney injury, circulatory failure, or decompensation, before they occur [11, 12, 33, 34]. These systems typically predict, at every time step, whether the event will occur within a fixed horizon, so that the prediction is aligned with the decision to raise an alarm, and they evaluate the resulting alarms at the event level rather than per time step.

Subsequent work examined the training target itself. Yèche et al. [13] related fixed-horizon EEP to dynamic survival analysis and proposed TLS, which replaces the abrupt fixed-horizon label with a target that varies smoothly with the time to the event. Yèche et al. [10] adapted dynamic survival analysis to EEP by restricting the survival objective to the horizon of interest and localizing risk within it, and they used this within-horizon information to prioritize alarms.

The survival-based disruption predictors discussed above, in contrast, model the residual time over its full range. The targets of binary classification, full-distribution and horizon-truncated survival analysis, and fixed-horizon EEP compared in this study are defined in section 3.

3 Methods

We first fix the notation (section 3.1) and contrast the three prediction targets used for disruption prediction (section 3.2). We then follow the progression from modeling the full time to disruption to modeling only the risk within the horizon: survival analysis (section 3.3), the horizon-truncated survival objective (section 3.4), and fixed-horizon EEP with TLS (section 3.5). Finally, we describe the causal encoder shared by the neural predictors (section 3.6).

3.1 Problem Setup

Consider shot $i \in \{1, \dots, N\}$, represented by a multivariate diagnostic sequence $\mathcal{X}_i = (\mathbf{x}_{i,0}, \mathbf{x}_{i,\Delta_x}, \dots, \mathbf{x}_{i,L_i\Delta_x})$ with $\mathbf{x}_{i,t} \in \mathbb{R}^d$, where L_i is the index of the final valid diagnostic sample of shot i, so that the sequence ends at physical time $L_i\Delta_x$. The sequence is sampled on the diagnostic input grid $\Delta_x = 5\,\text{ms}$. We use t for physical landmark time, so $t \in \{0, \Delta_x, \dots, L_i\Delta_x\}$. Let T_i be the physical disruption time and C_i the physical censoring time, both measured in milliseconds. The observed endpoint and event indicator are $Y_i = \min(T_i, C_i)$ and $\delta_i = \mathbb{I}\{T_i \le C_i\}$. All temporal quantities in the objectives below, including t, Y_i, $R_{i,t}$, and the training horizon H, are physical times rather than numbers of diagnostic input steps.

At each causal prediction origin, or landmark, $t < Y_i$, the observed residual follow-up time is $R_{i,t} = Y_i - t$. If $\delta_i = 1$, $R_{i,t}$ is the observed time remaining until disruption. If $\delta_i = 0$, the discharge ends without a disruption after $R_{i,t}$. Survival analysis treats this end as right censoring, which implies only that the disruption time would exceed $R_{i,t}$ if the discharge continued. Because a discharge that has ended cannot disrupt afterwards, the end of a non-disruptive shot can instead be treated as a terminal event that precludes disruption. The SA models of section 3.3 use the first view, and the EEP targets and survTLS (sections 3.4 and 3.5) use the second.

At landmark t, the measurements available to a causal predictor are the full observed prefix $\mathcal{X}_{i,0:t} = (\mathbf{x}_{i,0}, \dots, \mathbf{x}_{i,t})$, which excludes all future measurements $\mathbf{x}_{i,t+\Delta_x:L_i\Delta_x}$. We denote by $\mathcal{I}_i$ the set of valid landmarks of shot i used for training (section B.2).

3.2 Three Prediction Targets

Disruption predictors differ in the risk they are trained to output. Figure 1 contrasts the training targets of binary classification (BC), survival analysis (SA), and fixed-horizon EEP for a disruptive shot. The three targets answer different questions, and they differ in how much they say about when the disruption occurs.

Binary classification: is this discharge disruptive?

BC labels whether a discharge, or a state within it, is disruptive. In its simplest form, every landmark of shot i receives the shot outcome $d_{i,t} = \delta_i$, and the model estimates $\Pr_\theta(d_{i,t} = 1 \mid \mathcal{X}_{i,0:t})$. Alternatively, a pre-disruptive state can be annotated

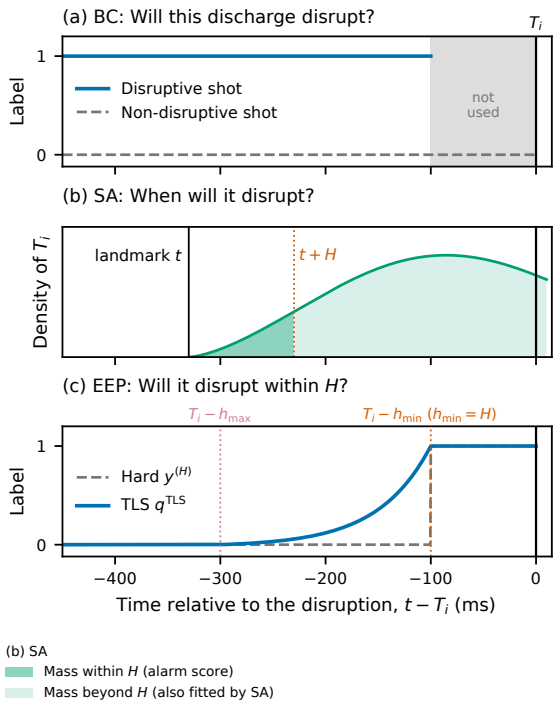


Fig. 1 Training targets for a disruptive shot. (a) BC assigns one label to the whole discharge. (b) SA models the distribution of the disruption time at every landmark, including the mass beyond $t+H$. (c) EEP labels every landmark by whether the disruption occurs within the horizon H; TLS replaces the hard boundary at $T_i - H$ by a target that decreases from one at $T_i - h_{\min}$, with $h_{\min} = H$, to zero at $T_i - h_{\max}$. The values of H and of the smoothing are illustrative.

from the physical onset of a precursor. BC is a simple and widely used problem, but its label does not specify when the disruption occurs relative to the landmark. A shot-level label also asks the model to flag early parts of a disruptive discharge that may not yet contain any precursor, and an onset-based annotation depends on the disruption mechanism [2].

Survival analysis: when will the discharge disrupt?

SA models the conditional distribution of the disruption time while accounting for right censoring [6]. It provides the time to disruption directly and the risk at any horizon from a single model [2, 7].

Fixed-horizon EEP: will the discharge disrupt within H?

EEP labels every landmark from its residual time and an operational prediction horizon H: the label is positive when the disruption occurs within the next H (eq. (5)), and the model estimates the probability of this event. No annotation of a pre-disruptive state is required, and learning concentrates on the horizon that matters for the control action. Conventional class-time classifiers [2] label the last Δ_{class} of a disruptive shot as positive; when Δ_{class} is chosen as an operational prediction horizon, this target coincides with the hard EEP target, and we classify such models as EEP.

Two axes.

The targets thus differ along two axes: whether they model the residual time (survival modeling), and whether learning is restricted to the operational horizon (horizon focus). SA models the residual time over its full range without horizon focus. The horizon-truncated survival objective of section 3.4 keeps survival modeling but restricts it to the horizon, and EEP focuses on the horizon without modeling the residual time.

3.3 Survival Analysis

At landmark t, an SA model with parameters Θ defines a conditional density $f_\Theta(\tau \mid \mathbf{z}_{i,t})$ of the residual time and the survival function $S_\Theta(\tau \mid \mathbf{z}_{i,t}) = \int_\tau^\infty f_\Theta(u \mid \mathbf{z}_{i,t})\,du$, where $\mathbf{z}_{i,t}$ is the encoded prefix (section 3.6). Under conditional independent censoring, the censored likelihood over causal landmarks gives the objective [30]

$$\mathcal{L}_{\text{SA}}(\Theta) = -\sum_i \sum_{t<Y_i} \Big[\delta_i \log f_\Theta\left(R_{i,t} \mid \mathbf{z}_{i,t}\right) + (1-\delta_i) \log S_\Theta\left(R_{i,t} \mid \mathbf{z}_{i,t}\right)\Big]. \quad (1)$$

The alarm score for horizon H is the predicted probability of a disruption within H, $1 - S_\Theta(H \mid \mathbf{z}_{i,t})$. We use two SA models evaluated in previous disruption studies [2]. DSM [9] parameterizes f_Θ as a mixture of M Weibull distributions whose weights and parameters are produced from $\mathbf{z}_{i,t}$, and optimizes a lower bound of eq. (1). DCPH [8] is a deep Cox proportional-hazards model whose hazard is a baseline hazard multiplied by

$\exp(g_\Theta(\mathbf{z}_{i,t}))$. It is trained with the Cox partial likelihood, and its survival function uses a baseline cumulative hazard Λ_0 estimated on the training data. Its horizon risk $1 - \exp[-\Lambda_0(H)e^{g_\Theta(\mathbf{z}_{i,t})}]$ is a strictly increasing function of $g_\Theta(\mathbf{z}_{i,t})$ whenever $\Lambda_0(H) > 0$, so the ranking of landmarks does not depend on H.

3.4 Horizon-Truncated Survival Objective (survTLS)

Learning the complete event-time distribution with $\mathcal{L}_{\text{SA}}$ requires fitting residual event times over the full observed range, including events far beyond the horizon relevant to the alarm. On long and finely sampled sequences, these distant and weakly predictable events can make optimization unnecessarily difficult. A first way to focus learning on the operational horizon keeps the survival model but truncates its objective at a physical training horizon H. survTLS [10] estimates a sequence of discrete hazards over future-time bins and fits them only within the horizon, so that it still localizes the risk in time within H.

Let δ_h be the output-bin width and $T_{\max} \geq H$ the time range of the outputs, so that the head has $K = T_{\max}/\delta_h$ outputs. The number of bins within the training horizon is $K_H = \lceil H/\delta_h \rceil$. Bin $k \in \{1, \ldots, K\}$ is centered at $k\delta_h$ and covers $B_k = \big((k - \frac{1}{2})\delta_h, (k + \frac{1}{2})\delta_h\big]$. The hazard head predicts $\lambda_\theta(k \mid \mathbf{z}_{i,t}) = \sigma(b_k(\mathbf{z}_{i,t}))$, the conditional probability that the event occurs in B_k given that it has not occurred in an earlier bin, where $b = (b_1, \ldots, b_K)$ is a neural prediction head.

The one-hot residual-time target is replaced by a smoothed event-time probability mass $\widetilde{f}_{i,t}(k)$, obtained by integrating over each bin a Gaussian centered at $R_{i,t}$ with standard deviation $R_{i,t}/\ell$, where ℓ is a dimensionless length scale; distant events thus receive less certain temporal localization. The mass below the first bin, including negative times, is assigned to $k = 1$, and the mass above $T_{\max}$ is assigned to $k = K$, so that $\sum_{k=1}^{K} \widetilde{f}_{i,t}(k) = 1$; the precise expressions are given in section B.5. Normal termination is treated as a terminal event (section 3.1): for landmarks of non-disruptive shots, $\widetilde{f}_{i,t}(k) = 0$ for every k, so that every bin within the horizon is negative even if the discharge ends earlier. Define the soft at-risk weight and the target hazard by

$$u_{i,t,k} = 1 - \sum_{j=1}^{k-1} \widetilde{f}_{i,t}(j), \qquad \widetilde{\lambda}_{i,t}(k) = \frac{\widetilde{f}_{i,t}(k)}{u_{i,t,k}}, \quad (2)$$

so that non-disruptive landmarks receive zero target hazards with unit weight. Because $u_{i,t,k}\widetilde{\lambda}_{i,t}(k) = \widetilde{f}_{i,t}(k)$ and $u_{i,t,k}(1 - \widetilde{\lambda}_{i,t}(k)) = 1 - \sum_{j \leq k} \widetilde{f}_{i,t}(j)$, the weighted terms of eq. (4) summed over all K bins give the expected discrete-time log-likelihood under the smoothed event-time distribution,

$$\sum_{k=1}^{K} \Big[\widetilde{f}_{i,t}(k) \log \lambda_{\theta,k} + \Big(1 - \sum_{j \leq k} \widetilde{f}_{i,t}(j)\Big) \log\left(1 - \lambda_{\theta,k}\right) \Big], \quad (3)$$

where $\lambda_{\theta,k} = \lambda_\theta(k \mid \mathbf{z}_{i,t})$. A sharp event therefore keeps its full weight in its own bin. The horizon-truncated objective uses only the first K_H bins and is averaged over the valid landmarks of each shot and then over the N training shots:

$$\mathcal{L}_{\text{surv}}^{(H)} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{I}_i|} \sum_{t \in \mathcal{I}_i} \sum_{k=1}^{K_H} u_{i,t,k} \times \Big[\widetilde{\lambda}_{i,t}(k) \log \lambda_\theta\left(k \mid \mathbf{z}_{i,t}\right) + \Big(1 - \widetilde{\lambda}_{i,t}(k)\Big) \log\left(1 - \lambda_\theta\left(k \mid \mathbf{z}_{i,t}\right)\right) \Big]. \quad (4)$$

The targets are computed over all K bins and are not renormalized after the truncation to K_H bins.

Restricting the objective to $k \leq K_H$ avoids fitting event times beyond the training horizon, and the ordered hazard profile provides a fine-grained representation of event imminence within H. At inference time, the hazards are aggregated into an alarm score that gives greater weight to shorter future offsets (section C.1).

3.5 Fixed-Horizon EEP with TLS

survTLS still models when the disruption occurs within the horizon. Fixed-horizon EEP goes one step further: it gives up modeling the residual time and directly learns the probability of a disruption within the next H, which aligns the objective with the early-warning task [13].

The hard fixed-horizon target is

$$y_{i,t}^{(H)} = \mathbb{I}\{\delta_i = 1 \ \wedge \ 0 < R_{i,t} \leq H\}. \tag{5}$$

Normal termination is treated as a terminal event: every landmark of a non-disruptive shot is negative for every H, including landmarks within H of the end of the discharge, because no disruption can occur after the end. The label is therefore observed at every valid landmark $t \in \mathcal{I}_i$, and the losses below use all valid landmarks without a censoring mask. A scalar prediction head outputs $p_\theta^{(H)}(\mathbf{z}_{i,t}) \in (0,1)$. When it is trained on the hard target, it estimates the probability that a disruption occurs within the next H milliseconds and before the discharge ends. This quantity differs from the horizon risk of the SA models, which treat the end of a non-disruptive shot as censoring and thus describe the disruption time of a discharge that would continue.

The hard target has two weaknesses. It switches abruptly at $T_i - H$, although the predictive signal does not change abruptly at this boundary, and only the last H of each disruptive shot is positive, which causes severe label imbalance. TLS [13] replaces the step with a target that decreases smoothly with the residual time. It reduces confidence where the signal is weak or ambiguous and concentrates learning on the comparatively rare event-proximal landmarks.

Let $h_{\min} < h_{\max}$ delimit the smoothing range in milliseconds, $\gamma > 0$, in ms^{-1}, control its curvature, and $D_{\mathrm{TLS}} = h_{\max} - h_{\min}$. Because γ multiplies a time, its value depends on the time unit; we use $\gamma = 0.025\,\mathrm{ms}^{-1}$ $(25\,\mathrm{s}^{-1})$. The exponential TLS smoothing profile is

$$\rho_\gamma(\tau) = \begin{cases} 1, & \tau \leq h_{\min}, \\ 0, & \tau \geq h_{\max}, \\ \dfrac{e^{-\gamma(\tau - h_{\min})} - e^{-\gamma D_{\mathrm{TLS}}}}{1 - e^{-\gamma D_{\mathrm{TLS}}}}, & \text{otherwise.} \end{cases} \tag{6}$$

The smoothed target at landmark t is $q_{i,t}^{\mathrm{TLS}} = \delta_i \rho_\gamma(R_{i,t})$. Because ρ_γ decreases from one to zero between $h_{\min}$ and $h_{\max}$, the TLS target is bounded by the hard targets of eq. (5) at these two horizons:

$$y_{i,t}^{(h_{\min})} \leq q_{i,t}^{\mathrm{TLS}} \leq y_{i,t}^{(h_{\max})}. \tag{7}$$

The three targets coincide outside $(h_{\min}, h_{\max})$, so Hard EEP at $H = h_{\min}$ or $H = h_{\max}$ differs from TLS only in the values assigned within the smoothing range.

The TLS objective is the binary cross-entropy averaged over the valid landmarks of each shot and then over the N training shots:

$$\mathcal{L}_{\mathrm{TLS}}^{(H)} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{I}_i|}\sum_{t\in\mathcal{I}_i}\Big[q_{i,t}^{\mathrm{TLS}}\log p_\theta^{(H)}(\mathbf{z}_{i,t}) + \big(1 - q_{i,t}^{\mathrm{TLS}}\big)\log\Big(1 - p_\theta^{(H)}(\mathbf{z}_{i,t})\Big)\Big]. \tag{8}$$

Replacing $q_{i,t}^{\mathrm{TLS}}$ in eq. (8) by the hard target $y_{i,t}^{(H)}$ gives the hard-label EEP objective, which we refer to as Hard EEP. When trained on the TLS target instead, the output estimates the conditional expectation of $q_{i,t}^{\mathrm{TLS}}$ given the prefix. Because $q_{i,t}^{\mathrm{TLS}}$ takes intermediate values for residual times between $h_{\min}$ and $h_{\max}$, this quantity is not the probability of a disruption within a fixed horizon. By eq. (7), however, it lies between the probabilities of a disruption within $h_{\min}$ and within $h_{\max}$, so its meaning as a risk is fixed by the two bounds. We therefore treat the TLS output as a smoothed, horizon-aware risk score and use it to rank landmarks and to trigger alarms. At inference time, the model monitors $p_\theta^{(H)}(\mathbf{z}_{i,t})$, and an alarm is raised when this score exceeds a threshold calibrated on validation data.

3.6 Shared Causal Encoder

All neural predictors except CCNN are conditioned on a common causal feature representation. The encoder separates a recurrent history branch from a recent-window branch that directly retains high-resolution measurements.

At landmark t, the history branch has processed $\mathcal{X}_{i,0:t}$ causally and maintains the recurrent representation $\mathbf{z}_{i,t}^{\mathrm{hist}} = g_{\mathrm{hist}}(\mathcal{X}_{i,0:t})$. The recurrent state is propagated through the complete causal prefix, while gradients are truncated to the most recent 256 input steps during backpropagation through time (BPTT). The forward state can therefore carry information from before the truncation boundary, but gradient-based credit assignment does not extend through the complete prefix.

The recent-window branch directly retains the ordered measurements in the most recent L_{win} diagnostic input samples. Defining $\mathbf{x}_{i,t'} = \mathbf{0}$ for $t' < 0$, the causal recent window and its representation are $\mathbf{w}_{i,t} = [\mathbf{x}_{i,t-(L_{\text{win}}-1)\Delta_x}; \ldots; \mathbf{x}_{i,t}] \in \mathbb{R}^{L_{\text{win}}d}$ and $\mathbf{z}^{\text{win}}_{i,t} = g_{\text{win}}(\mathbf{w}_{i,t})$. Unlike the history representation, the recent-window representation bypasses recurrent compression of the recent measurements and therefore preserves short-timescale precursor dynamics explicitly.

The two representations are concatenated as $\mathbf{z}_{i,t} = [\mathbf{z}^{\text{hist}}_{i,t}; \mathbf{z}^{\text{win}}_{i,t}]$. A single-branch model uses only the corresponding component of $\mathbf{z}_{i,t}$. In the reported experiments, g_{hist} is instantiated as a GRU and g_{win} as an MLP applied to the flattened recent window. These are implementation choices for the two information-preservation mechanisms; the prediction objectives below are independent of the selected encoder architecture.

Use of large language models.

Large language model (LLM)-based coding assistants (Claude, Anthropic; Codex, OpenAI) were used mainly to implement and test the experiment and evaluation code, and to assist in discussing the interpretation of the results. All code, results, and their interpretation were checked by the authors, who take full responsibility for the content of this article.

4 Experiments

4.1 Experimental Design

We conduct three studies on DIII-D, Alcator C-Mod, and EAST, treating each tokamak as a separate within-device task.

1. *Predictor comparison.* We compare predictors that differ in the quantity they are trained to predict: binary classification (BC), full-distribution survival analysis (SA), and fixed-horizon EEP (section 4.3). All neural sequence models except CCNN share the causal encoder of section 3.6; CCNN and RF use their own feature extraction.
2. *Training horizon and evaluation deadline.* Each method selects its training horizon H, or for TLS its smoothing bounds $(h_{\min}, h_{\max})$, on validation data (section 4.4). Hard EEP is trained at $H = h_{\min}$ of the selected TLS bounds, so that it differs from TLS only in the smoothing, and Hard EEP at $H = h_{\max}$ is reported as a second reference (section 4.6.2). Every selected model is evaluated at the deadlines $w \in \{10, 40, 100, 300\}$ ms. For TLS, we additionally fix each upper bound $h_{\max} \in \{20, 40, 80, 120, 240, 360\}$ ms, select its lower bound on the monitor split, and train it with three seeds, which gives an $h_{\max} \times w$ grid.
3. *Encoder ablation.* For TLS, we fix the selected bounds and training cohort and remove either the history branch or the recent-window branch of the encoder. The conclusions of this ablation therefore concern the encoder used with the TLS objective.

Within each study, all settings other than the studied factor are fixed.

Table 1 DisruptionBench database composition before resampling; parentheses give disruptive-shot fractions.

Device	Shots	Disruptions	Native interval
C-Mod	4,510	975 (21.6%)	5 ms
DIII-D	8,608	1,244 (14.5%)	25 ms
EAST	14,347	4,508 (31.4%)	100 ms

4.2 Datasets and Preprocessing

We use the public DIII-D, Alcator C-Mod, and EAST databases from DisruptionBench [3]. Each shot comprises a multivariate diagnostic sequence and a disruptive or non-disruptive endpoint. For DSM and DCPH, non-disruptive shots are right-censored at their last valid observation; for the EEP models and survTLS, their landmarks are negative (section 3.5).

Nine plasma quantities are common to the three databases. They describe magnetohydrodynamic and equilibrium state, density-limit proximity, plasma shaping, current tracking, and loop-voltage evolution; symbols and units are listed in section B.1. The DIII-D and EAST databases each provide three additional device-specific signals, which the neural sequence models use as additional input channels. The C-Mod input contains three machine-indicator channels instead, which are constant within the within-device task and therefore carry no information.

Table 2 Compared predictors, grouped by prediction target (section 3.2). "Shared" denotes the causal encoder of section 3.6.

Method	Target	Label or objective	Model
CCNN	BC	Shot label $d_{i,t}$	Conv. net
DSM	SA	$\mathcal{L}_{\mathrm{SA}}$ (Weibull mix.)	Shared
DCPH	SA	Cox partial lik.	Shared
survTLS	SA ($\leq H$)	$\mathcal{L}_{\mathrm{surv}}^{(H)}$	Shared
RF	EEP	Hard $y_{i,t}^{(H)}$	Forest
Hard EEP	EEP	Hard $y_{i,t}^{(h_{\min})}$	Shared
TLS	EEP	Smoothed $q_{i,t}^{\mathrm{TLS}}$	Shared

Signals are placed on a causal 5 ms grid by forward-filling only previously observed values; this common diagnostic input-grid interval is denoted by $\Delta_x = 5$ ms. We use shots whose prepared sequences contain between 15 and 2,048 landmarks (75 ms to about 10 s), and we disable absolute-time and short-time Fourier transform (STFT) features. Each channel is standardized with statistics estimated on the training partition. Shot-level training, calibration, monitoring, and test fractions are 60%, 10%, 10%, and 20%, assigned once with a fixed seed and stratified by shot outcome; partitioning precedes prefix generation. The disruption time T_i is taken from the DisruptionBench databases and is resolved on the 5 ms grid. The resulting partition sizes, the observation time of non-disruptive shots, the eligibility rules, and the temporal resolution of T_i are given in section B.2.

4.3 Compared Models

Table 2 lists the compared predictors, grouped by the prediction targets of section 3.2. All neural predictors except CCNN use the shared encoder of section 3.6; implementation details and method-specific hyperparameters are listed in sections B.3 and B.4.

BC.

CCNN [23], the continuous convolutional network evaluated in DisruptionBench [3], is trained as a sequence-to-label classifier on the shot label $d_{i,t} = \delta_i$, with each disruptive training sequence truncated at $T_i - H$; its score at landmark t is its prediction for the prefix ending at t. Because its training settings and evaluation protocol differ from the original ones (section B.4), the results are not directly comparable with the published intra-machine C-Mod AUC of approximately 0.97.

SA.

DSM uses Auton-Survival,[1] the implementation used in previous risk-aware disruption-prediction research [2], and is trained on every valid landmark (section B.4). For DCPH, each risk set of the partial likelihood is formed from all valid landmarks in a mini-batch of shots. survTLS follows the horizon-truncated objective and risk-localization construction of Yèche et al. [10].

EEP.

TLS follows the authors' official target construction with the exponential smoothing profile of eq. (6).[2] Hard EEP is trained on the hard target with $H = h_{\min}$ of the selected TLS bounds and shares the training cohort, encoder, and optimization of TLS, so that the two differ only in the smoothing of the target. RF is a random-forest classifier, the model family used in earlier real-time disruption predictors [1, 20], trained on the hard target $y_{i,t}^{(H)}$ with 63 causal window features of the nine common signals (section B.4); it therefore differs from Hard EEP in both its input and its model class.

4.4 Optimization and Model Selection

All neural models share the optimizer, schedule, and stopping rule (sections B.3 and B.4), and for each candidate configuration we retain the checkpoint with the largest monitor-split Deadline AUROC at $w = 40$ ms. CCNN, DSM, and DCPH search $H \in \{20, 40, 80, 120\}$ ms, and RF and survTLS search $H \in \{20, 40, 80, 120, 240, 360\}$ ms. DSM additionally searches the number of mixture components $M \in \{1, 3\}$, and survTLS the output-bin width and learning rate. TLS searches the 22 smoothing intervals $(h_{\min}, h_{\max})$ of table 11. For DCPH, H changes only which short disruptive shots are eligible for training, not the ranking produced by a given model (section B.4). After

[1] Auton-Survival: https://github.com/autonlab/auton-survival

[2] TLS: https://github.com/ratschlab/tls

threshold calibration (section 4.5), the configuration with the largest monitor-split Deadline F1 of seed 0 is selected and frozen for seeds 1 and 2 (section C.6). Test data are not used for any selection.

4.5 Evaluation Protocol

Following section 1, we evaluate separately the risk output by each predictor and the alarm policy built on it. A useful alarm must be raised sufficiently far in advance of a disruption to allow control or mitigation measures to be implemented. We denote this required lead time by w and use $w = 40\,\text{ms}$ as the primary deadline, following the DisruptionBench setting for the ITER disruption-mitigation response [3]. Because the appropriate deadline depends on the available control action [2], the horizon study additionally evaluates $w \in \{10, 100, 300\}\,\text{ms}$. H is the physical training horizon used to construct the training target, whereas w is the required alarm lead time used only to define the evaluation deadline; the two quantities need not be equal.

Ranking metric.

Deadline AUPRC evaluates the risk: it measures whether disruptive shots receive higher scores than non-disruptive shots, using only information available by the deadline. Keith et al. [2] used AUROC for this purpose; because disruptive shots are a minority, we use AUPRC, as in EEP evaluation [13, 14], and report Deadline AUROC in section D.

Alarm metrics.

The alarm policy is evaluated at the event level, following EEP evaluation [10, 13], Deadline Recall is the proportion of disruptions for which at least one alarm is raised no later than w before the disruption, and Alarm Precision is the proportion of emitted alarms that are useful. At most one true alarm is credited per disruptive shot; all normal-shot, late, and duplicate alarm episodes are counted as false alarms. Deadline F1 is the harmonic mean of the two, and FAR/24h is the number of alarm events in non-disruptive shots normalized to 24 hours of normal operation. Definitions are given in section C.

Alarm policy.

Each method's score sequence (section C.1) is post-processed with the fixed settings of section B.6 and thresholded. The threshold is selected on the calibration split so that the normal-shot false-positive rate (FPR), the proportion of non-disruptive shots with at least one alarm, is approximately 1%. Together with the checkpoint and configuration selected on the monitor split, this defines a validation-selected fixed policy that is applied unchanged to the test split. To separate calibration from ranking, we also sweep the threshold on the test data and trace Deadline Recall against the realized FAR/24h. These Recall–FAR curves use test labels and are descriptive only.

Uncertainty.

We report the mean and sample standard deviation (ddof = 1) over three training seeds on one fixed split, and a paired shot-level bootstrap with 2,000 replicates. Each replicate resamples the test shots with replacement, separately within disruptive and non-disruptive shots, from the test shots evaluated by every compared method, and the same resampled shots are used for all methods and seeds. Each metric is averaged over the seeds within a replicate, and we report 95% percentile intervals for the difference between two methods. These intervals are conditional on the fixed split and training seeds and do not describe variation across data splits or devices; we therefore also show the value of every training seed. The sensitivity of the fixed policy to the calibration split is analyzed by resampling the calibration shots 1,000 times with the checkpoints and test data fixed (section D).

4.6 Results

4.6.1 Comparison of Training Targets

Figure 2 compares the methods at the primary deadline $w = 40\,\text{ms}$. Configurations are selected on the monitor split with seed 0 and frozen for seeds 1 and 2. Because the fixed-policy alarm metrics vary strongly across training seeds, the figure shows every seed together with the mean. Numerical values and paired differences are reported in tables 12 and 13.

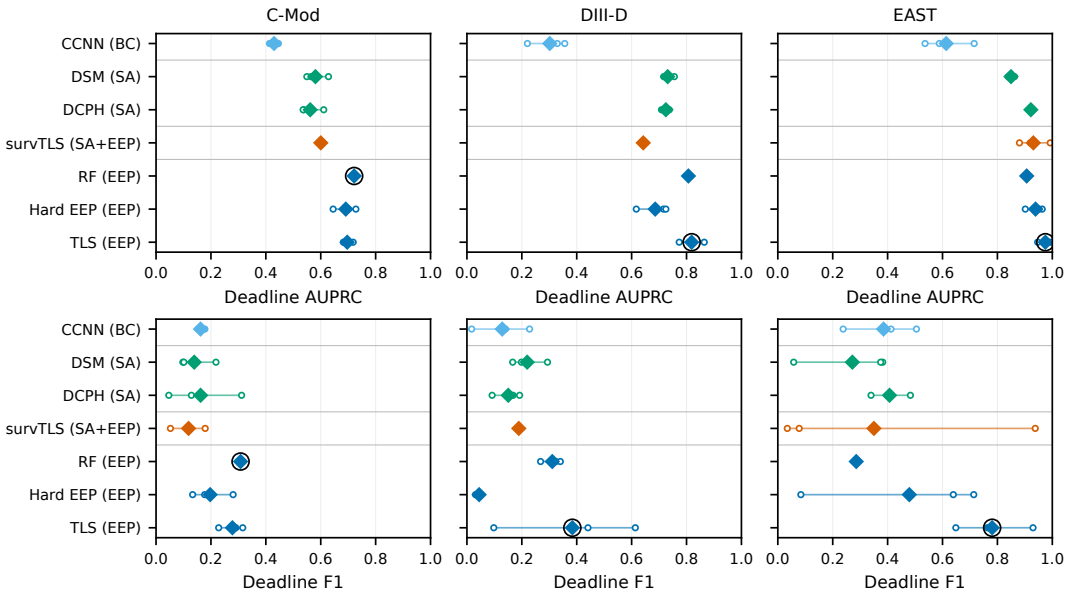


Fig. 2 Deadline AUPRC (top) and validation-selected fixed-policy Deadline F1 (bottom) at $w = 40$ ms. Open circles show the three training seeds, diamonds their mean, and lines the range across seeds. Colors indicate the training target; the circled point marks the largest mean on each device. Hard EEP is trained at $H = h_{\min}$ of the selected TLS bounds.

Table 3 Label-smoothing ablation at $w = 40$ ms: TLS with the selected bounds $(h_{\min}, h_{\max})$ and Hard EEP with $H = h_{\min}$ and $H = h_{\max}$, trained on the same cohort with the same encoder and optimization. Mean ± sample standard deviation over three training seeds, and TLS minus Hard EEP with 95% paired-bootstrap intervals; an asterisk marks intervals that exclude zero.

Device	Target (ms)	Deadline AUPRC	Deadline F1	TLS − Hard: Δ AUPRC [95% CI]	TLS − Hard: Δ F1 [95% CI]
C-Mod	Hard $H = h_{\min} = 240$	0.691 ± 0.042	0.197 ± 0.076	$+0.005$ $[-0.013, +0.024]$	$+0.081$ $[+0.044, +0.118]^*$
	Hard $H = h_{\max} = 360$	0.717 ± 0.016	0.209 ± 0.061	-0.020 $[-0.042, +0.002]$	$+0.069$ $[+0.036, +0.104]^*$
	TLS (240, 360)	0.696 ± 0.019	0.278 ± 0.045		
DIII-D	Hard $H = h_{\min} = 40$	0.686 ± 0.060	0.044 ± 0.011	$+0.133$ $[+0.098, +0.162]^*$	$+0.340$ $[+0.301, +0.377]^*$
	Hard $H = h_{\max} = 120$	0.734 ± 0.044	0.132 ± 0.097	$+0.085$ $[+0.049, +0.118]^*$	$+0.252$ $[+0.213, +0.292]^*$
	TLS (40, 120)	0.819 ± 0.046	0.384 ± 0.262		
EAST	Hard $H = h_{\min} = 120$	0.939 ± 0.033	0.479 ± 0.344	$+0.036$ $[+0.028, +0.044]^*$	$+0.302$ $[+0.283, +0.320]^*$
	Hard $H = h_{\max} = 240$	0.947 ± 0.036	0.624 ± 0.154	$+0.028$ $[+0.021, +0.035]^*$	$+0.157$ $[+0.136, +0.177]^*$
	TLS (120, 240)	0.975 ± 0.025	0.781 ± 0.141		

EEP versus survival analysis.

On every device, the best EEP model exceeds both DSM and DCPH in Deadline AUPRC and Deadline F1 (table 13). The best EEP model is TLS on DIII-D (Deadline AUPRC 0.819, Deadline F1 0.384) and EAST (0.975, 0.781), and RF on C-Mod (0.722, 0.308). TLS exceeds RF on DIII-D in Deadline F1 by +0.073 ([+0.025, +0.123]) without a resolved difference in Deadline AUPRC, and on EAST in both metrics. On C-Mod, RF is slightly ahead of TLS in both metrics, but neither difference is resolved.

survTLS, which restricts the survival objective to the horizon, does not consistently outperform the full-distribution survival models: its mean Deadline AUPRC is higher than those of DSM and DCPH on C-Mod and EAST but lower on DIII-D, and its mean Deadline F1 exceeds both on none of the devices. Because survTLS and DSM also differ in the parameterization of the residual time, the target smoothing, and the alarm score (section C.1), this comparison does not isolate horizon truncation, but it shows little benefit of truncation relative to DSM.

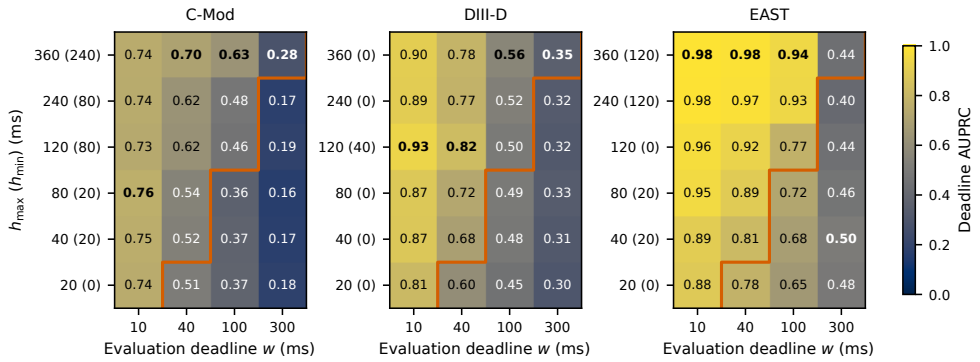


Fig. 3 Deadline AUPRC of TLS for each upper bound $h_{\max}$ of the target and evaluation deadline w, averaged over three training seeds. Row labels give $h_{\max}$ and, in parentheses, the lower bound $h_{\min}$ selected for it. The orange line separates cells with $w \leq h_{\max}$ (left) from cells with $w > h_{\max}$ (right); bold values mark the largest value in each column.

Binary classification.

The successful alarms of CCNN are raised much earlier than those of the other methods, with a median warning time above 1 s on DIII-D and EAST. A shot-level target does not tell the model when the disruption occurs, so its scores often rise early in a discharge; such early excursions also occur in non-disruptive shots, where they count as false alarms.

Device dependence.

Alarm performance remains limited on C-Mod and DIII-D for every method: the largest mean Deadline F1 is 0.308 on C-Mod and 0.384 on DIII-D, compared with 0.781 on EAST. C-Mod is also the most difficult device in ranking. When the deadline is shortened to $w = 10$ ms, however, every TLS configuration on C-Mod reaches a Deadline AUPRC of 0.73–0.76 (fig. 3), which is consistent with precursors that develop shortly before the disruption.

Ranking versus calibrated alarms.

The fixed-policy alarm metrics vary much more across seeds than the ranking metric. The Deadline F1 of TLS on DIII-D is 0.098, 0.441, and 0.613 for the three seeds, whereas its Deadline AUPRC varies only from 0.773 to 0.865; Hard EEP and survTLS on EAST show the same pattern. The shot-level bootstrap intervals are conditional on the trained seeds and do not include this variation. Threshold selection contributes to it, because each seed uses a calibration split with fewer than 1,200 shots (table 9). Resampling the calibration shots with each checkpoint fixed moves the Deadline F1 of TLS on DIII-D within [0.288, 0.526] and that of survTLS on EAST within [0.343, 0.551] (seed-averaged 2.5th and 97.5th percentiles; table 14). The Recall–FAR curves (fig. 4) show that the ranking supports the fixed-policy results at the primary budgets: on DIII-D and EAST, TLS reaches the highest Deadline Recall near the primary FAR budget, whereas on C-Mod the curves of TLS, RF, and Hard EEP overlap. Fixed-policy alarm metrics should therefore be interpreted together with the threshold-free ranking and with their variation across seeds and calibration data.

4.6.2 Ablation: Label Smoothing

The hard targets at $H = h_{\min}$ and $H = h_{\max}$ bound the TLS target (eq. (7)), so comparing TLS with Hard EEP at both bounds, with the cohort, encoder, and optimization fixed, isolates the effect of the smoothing (table 3).

TLS exceeds Hard EEP at both bounds in Deadline F1 on all three devices, and every paired interval excludes zero: by +0.340 and +0.252 on DIII-D, +0.081 and +0.069 on C-Mod, and +0.302 and +0.157 on EAST, for $H = h_{\min}$ and $H = h_{\max}$, respectively. Hard EEP at $H = h_{\max}$ has a higher mean Deadline F1 than at $H = h_{\min}$ on every device: extending the positive region already helps the hard target, but the graded target helps more.

Table 4 TLS encoder ablation: Deadline AUPRC and Deadline F1 at $w = 40$ ms. Bold entries mark the largest mean within each device.

Device	Encoder	AUPRC	F1
C-Mod	GRU+MLP	**0.696** ± 0.019	**0.278** ± 0.045
	GRU-only	0.670 ± 0.030	0.215 ± 0.065
	MLP-only	0.618 ± 0.014	0.061 ± 0.020
DIII-D	GRU+MLP	**0.819** ± 0.046	**0.384** ± 0.262
	GRU-only	0.745 ± 0.041	0.066 ± 0.019
	MLP-only	0.678 ± 0.029	0.025 ± 0.044
EAST	GRU+MLP	0.975 ± 0.025	0.781 ± 0.141
	GRU-only	**0.994** ± 0.004	**0.941** ± 0.017
	MLP-only	0.960 ± 0.005	0.256 ± 0.016

4.6.3 Ablation: Label Range and Evaluation Deadline

For each upper bound $h_{\max}$, the lower bound $h_{\min}$ is selected on the monitor split at $w = 40$ ms with seed 0 and frozen for all deadlines and seeds. Figure 3 reports the Deadline AUPRC of these configurations for every evaluation deadline w. The rows differ in $h_{\min}$ and, for some intervals taken from earlier runs, in the training cohort (table 11), so the comparison across rows is descriptive.

The Deadline AUPRC decreases as the evaluation deadline w grows, and an upper bound $h_{\max}$ that covers the deadline mitigates this decrease. This behavior is expected: an earlier warning is a harder task, and $h_{\max}$ sets the range of residual times over which the target carries information about the risk. At $w = 300$ ms, which exceeds every upper bound except the largest, the Deadline AUPRC is at most 0.50 on all devices. The target should therefore extend to the lead time required by the intended control action.

4.6.4 Ablation: Encoder Architecture

We fix the TLS objective, data partitions, training cohort, and selected bounds, and remove either the recent-window branch (GRU-only) or the history branch (MLP-only) (table 4).

The history branch is essential on every device: MLP-only yields the lowest Deadline AUPRC and Deadline F1 on all three devices. The contribution of the recent-window branch is device dependent: GRU+MLP achieves the best Deadline AUPRC and Deadline F1 on DIII-D and C-Mod, whereas GRU-only is best on EAST. A gating mechanism that can attenuate the recent-window features when they are not useful may therefore be beneficial.

5 Conclusion and Limitations

This study asked which risk a disruption predictor should provide to an alarm policy, and compared BC, full-distribution and horizon-truncated SA, and fixed-horizon EEP on the DIII-D, Alcator C-Mod, and EAST databases of DisruptionBench under a common causal protocol. On every device, TLS outperformed DSM and DCPH in both Deadline AUPRC and Deadline F1 at a 40 ms deadline. TLS performed best on DIII-D and EAST, and a random forest trained on a hard horizon target performed best on C-Mod. TLS also exceeded hard targets placed at either end of its smoothing range in Deadline F1 on all three devices, so the graded target itself, rather than the position of the label boundary, improves the alarms. Truncating the survival objective at the horizon showed little benefit relative to full-distribution SA. For the risk that drives a disruption alarm, these results favor learning the risk within an operational horizon directly over modeling the residual time over its full range.

A fixed-horizon model provides little information beyond the upper bound of its target, so the label range should cover the warning time required by the intended action. Fixed-policy alarm metrics varied much more across seeds than the ranking of the underlying risk, so alarm thresholds should be evaluated together with that ranking and across seeds. Alarm performance remained limited on C-Mod and DIII-D, which suggests that the global plasma quantities used here capture only part of the precursor information on these devices.

This study has several limitations. All results are within-device and were obtained on one fixed split with three training seeds. Configurations were selected with one seed, and TLS was selected from more candidates than the other methods. The inputs are limited to the global plasma quantities of DisruptionBench, CCNN was reimplemented under the present protocol, and the evaluation is offline rather than in a real-time PCS. Future work includes stronger representations, multiple horizons for mitigation and avoidance, and cross-device transfer of EEP models.

Acknowledgements. This work was supported by the Japan Science and Technology Agency (JST) through Moonshot R&D Grant Number JPMJMS24A3, CREST Grant Number JPMJCR24Q5, and ASPIRE Grant Number JPMJAP2329, and by the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Actions Staff Exchanges grant agreement No. 101131557 (REMODEL).

Declarations

- **Funding:** This work was supported by the Japan Science and Technology Agency (JST) through Moonshot R&D Grant Number JPMJMS24A3, CREST Grant Number JPMJCR24Q5, and ASPIRE Grant Number JPMJAP2329, and by the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Actions Staff Exchanges grant agreement No. 101131557 (REMODEL).
- **Competing interests:** The authors have no competing interests to declare that are relevant to the content of this article.
- **Data availability:** The DIII-D, Alcator C-Mod, and EAST databases analyzed in this study are publicly available through DisruptionBench [3] at https://github.com/MIT-PSFC/DisruptionBench.
- **Code availability:** The code used for preprocessing, training, and evaluation is provided as Supplementary Information and will be made publicly available upon publication. DSM was implemented with Auton-Survival (https://github.com/autonlab/auton-survival).
- **Author contributions:** T.K. conceived the study, designed and implemented the methods, conducted the experiments, analyzed the results, and wrote the manuscript. T.Y. contributed to the design of the methods and to the analysis of the results. Both authors read and approved the final manuscript.

Appendix A Notation

Table 5 summarizes the notation used in the training and evaluation formulations.

Appendix B Detailed Experimental Configuration

B.1 Diagnostic Quantities

Table 6 lists the input channels. The nine quantities in the upper block are common to the three databases. The neural predictors on DIII-D and EAST additionally receive the three device-specific signals in the lower block, so that each device has twelve input channels. The C-Mod input instead contains three machine-indicator channels, which are constant within the within-device task and become zero after standardization. RF uses only the nine common quantities. Definitions and units of the common quantities follow Spangher et al. [3]. The device-specific signals are not documented there; we list them by their database column names and do not interpret them physically.

B.2 Shot Partitions and Eligibility

Data and split identification.

The inputs are prepared from the database files `d3d_for_nn.mat`, `C_Mod_for_nn.mat`, and `EAST_for_nn.mat` of the DisruptionBench data archive [3]. Shots in which an input signal is never observed are removed (808 DIII-D, 6 C-Mod, and 2,965 EAST shots), and the leading rows of a shot are discarded until every signal has been observed, so that the forward-fill never uses a later value. This leaves 7,800 DIII-D (1,161 disruptive), 4,504 C-Mod (972), and 11,382 EAST (3,361) shots. Shots are assigned to the four partitions by a random assignment stratified by the shot outcome, with seed 42, which is identical for all methods and seeds. Models were trained with revision `3b7239c` of the project code (four seed-0 fits with revision `118a11d`); the follow-up TLS, Hard EEP, RF, and survTLS fits used revisions `aeb50a1`, `bc97ea7`, `86170aa`, and `d958342`, which add options for the target bounds, the extended horizons, and the training cohort. The evaluation was computed with revisions `214804b` and `78eb775` (follow-up). All survTLS results were subsequently replaced by new fits with revision `b0234ae`, which uses the at-risk weight in eq. (2); their saved-prediction evaluation used revision `dc30a05`. Other methods were retained unchanged.

Table 5 Principal notation used in the training and evaluation formulations.

Symbol	Meaning	Unit or setting
Δ_x	Diagnostic input-grid interval	5 ms
H	Physical training horizon	ms
δ_h	survTLS output-bin width	1 or 5 ms
$K_H = \lceil H/\delta_h \rceil$	Number of survTLS hazard bins within H	bins
w	Required alarm lead time	ms
δ_i	Event indicator for shot i	binary
$\mathcal{I}_i$	Valid causal landmark set for shot i	landmarks
ρ_γ	Exponential TLS smoothing profile	dimensionless
$h_{\min}$, $h_{\max}$	Lower and upper bounds of the TLS smoothing range	ms
$q_{i,t}^{\mathrm{TLS}}$	TLS soft target	value in $[0, 1]$
$\widetilde{f}_{i,t}(k)$	survTLS smoothed event-time mass	probability mass
$u_{i,t,k}$	survTLS soft at-risk weight	value in $[0, 1]$
$r_{i,t}^{(m)}(H)$	Alarm score for method m	dimensionless
$v_i^{(w)}$	Threshold-free deadline score for shot i	dimensionless
$\psi_\alpha^{\exp}$	Exponential alarm-priority function	dimensionless
γ	TLS smoothing curvature	$0.025\,\mathrm{ms}^{-1}$
α	survTLS alarm-priority rate	$0.025\,\mathrm{ms}^{-1}$
ℓ	survTLS target length scale	dimensionless

Table 6 Input channels. Upper block: quantities common to the three databases. Lower block: device-specific signals, given by their database names.

Symbol	Physical quantity	Unit
β_p	Plasma pressure normalized by poloidal magnetic pressure	dimensionless
ℓ_i	Normalized plasma internal inductance	dimensionless
q_{95}	Safety factor at 95% normalized flux	dimensionless
$B_{n=1}$	$n = 1$ component of the perturbed magnetic field	T
n/n_G	Electron density normalized by the Greenwald limit	dimensionless
g_{lower}	Gap between the plasma and lower divertor	m
κ	Plasma elongation	dimensionless
$I_{p,\mathrm{error}}/I_{p,\mathrm{prog}}$	Plasma-current tracking error normalized by the programmed current	dimensionless
V_{loop}	Toroidal loop voltage	V
`n1rms_normalized`	DIII-D database column	–
`radiated_fraction`	DIII-D database column	–
`zcur`	DIII-D database column	–
`Mirnov_std_normalized`	EAST database column	–
`rad_input_frac`	EAST database column	–
`z_error`	EAST database column	–

Eligibility and partition sizes.

A shot enters a partition only if its prepared sequence contains between 15 and 2,048 landmarks, which excludes some long EAST discharges, and, for a disruptive shot, at least one landmark with $R_{i,t} \geq H_{\mathrm{elig}}$. For training, H_{elig} is the training horizon H; for evaluation, it is the deadline w. Table 9 gives the resulting partition sizes for $H_{\mathrm{elig}} = 40\,\mathrm{ms}$, which are also the test sets of the evaluation at $w = 40\,\mathrm{ms}$ and the common shot sets of the paired bootstrap. For $H = 120\,\mathrm{ms}$, between 15 (EAST) and 28 (DIII-D) disruptive training shots and up to 9 disruptive test shots are additionally excluded; the non-disruptive partitions do not change.

Table 7 Common optimization and reproducibility settings.

Setting	Value
Common time grid	5 ms (200 Hz)
Prepared-data split	Fixed across all runs
Training seeds	0, 1, and 2
Reported uncertainty	Sample standard deviation across training seeds (ddof = 1)
Encoder	one-layer history GRU plus recent-window MLP
GRU hidden dimension	64
GRU output normalization	LayerNorm
GRU truncated-BPTT length	256 input steps
Recent-window length	10 samples (50 ms)
Recent-window MLP	hidden dimension 64, depth 2
Encoder dropout	0
Optimizer	AdamW
Batch size	50
Learning-rate schedule	cosine decay
Warm-up	10 epochs
Gradient clipping	maximum norm 1.0
Early-stopping patience	100 epochs
Time / STFT features	disabled

Table 8 Fixed shot-level partition roles.

Partition	Fraction	Role
Training	60%	Parameter optimization
Calibration	10%	Alarm-threshold selection
Monitoring	10%	Checkpoint and configuration selection
Test	20%	Final held-out evaluation

Table 9 Partition sizes after preprocessing and eligibility at $H_{\mathrm{elig}} = 40$ ms: disruptive / non-disruptive shots, and the total observation time of non-disruptive shots D_0 in hours.

Partition	C-Mod	DIII-D	EAST
Training	543 / 2,117	692 / 3,978	1,971 / 4,680
Calibration	92 / 353	116 / 664	331 / 782
Monitoring	92 / 353	116 / 664	327 / 775
Test	184 / 707	225 / 1,324	657 / 1,561
Test D_0 (h)	0.114	1.444	2.600

Normalization.

Each input channel is standardized with the mean and standard deviation estimated on all landmarks of the eligible training shots of the corresponding run; the same statistics are applied to the calibration, monitoring, and test partitions. Channels that are constant on the training data, such as the C-Mod machine indicators, become zero.

Disruption time.

The disruption time T_i is taken from the time-to-disruption field of the DisruptionBench databases [3] and is not redefined in this work. DisruptionBench does not specify a single physical criterion for this time across the three devices; the 40 ms deadline is motivated there by the minimum activation time of the ITER disruption mitigation system. For every disruptive shot, $t + R_{i,t}$ is constant to within 1 μs over all retained landmarks, and the last retained landmark lies at most one input-grid interval (5 ms) before T_i on DIII-D. On C-Mod and EAST, the last landmark of 633 and 5 disruptive shots, respectively, lies up to 10 ms and 14 ms before T_i. The deadline $T_i - w$ is resolved on the 5 ms input grid. Because inputs are forward-filled, the most recent raw measurement at a landmark can be up to one native sampling interval old (5, 25, and 100 ms on C-Mod, DIII-D, and EAST).

B.3 Optimization and Reproducibility

Table 7 lists the optimization and encoder settings shared by all neural predictors except CCNN.

B.4 Method-Specific Settings

Table 10 lists the objective, the searched settings, and the learning rates of each method, and table 11 lists the candidate TLS smoothing intervals. Each TLS candidate is trained on the cohort of shots that are eligible for an eligibility horizon H_{elig} (section B.2). The intervals in the last row of the table were added in a follow-up search with a common cohort ($H_{\text{elig}} = 120\,\text{ms}$). The interval $(40, 120)\,\text{ms}$ appears twice, with cohorts of 40 and 120 ms. For survTLS, the smoothing support is $T_{\max} = 120\,\text{ms}$ for $H \leq 80\,\text{ms}$ and $T_{\max} = 3H$ otherwise, and the alarm score of section C.1 is integrated up to H.

Search spaces.

DSM is trained on every valid landmark, with the loss averaged over time within a shot and then over shots, because sampling one landmark per shot in each epoch would give it far fewer training landmarks than the EEP models; the sampled variant is reported in table 16. survTLS searches $(\delta_h, \eta) \in \{(1\,\text{ms}, 10^{-4}), (5\,\text{ms}, 10^{-4}), (5\,\text{ms}, 10^{-3})\}$, because its original setting was the only one with a bin width finer than the input grid and a reduced learning rate. RF and survTLS search horizons up to 360 ms, the largest upper bound of the TLS candidates. Each method therefore has four candidate configurations per device, except DSM (eight), RF (six), survTLS (eighteen), and TLS (22); Hard EEP is not searched separately.

Horizon of DCPH.

The partial likelihood of DCPH does not involve H, and for a given checkpoint the horizon risk $1 - \exp[-\Lambda_0(H)e^{g_\Theta}]$ is a strictly increasing transform of the log-risk g_Θ. On a fixed set of shots and landmarks, H therefore changes neither the deadline ranking, nor the checkpoint selected by Deadline AUROC, nor the operating points obtained from the quantile-based threshold candidates. Results for DCPH can nevertheless differ across H for two reasons. First, a disruptive shot is eligible only if it has a landmark with $R_{i,t} \geq H$, so a longer horizon removes a few short disruptive shots from the training, calibration, and monitoring splits and thus changes the training trajectory. Second, when $\Lambda_0(H)e^{g_\Theta}$ is large, the horizon risk saturates numerically at one and ties the highest-risk landmarks. On DIII-D, for example, $H = 20$ and 40 ms yield the same eligible shots and identical results on every metric. The horizon search for DCPH therefore selects among slightly different training cohorts rather than among prediction horizons; we retain it so that every method is searched over a common grid.

CCNN.

In Arnold et al. [23], each shot is truncated 40 ms before its end, and the AUC is computed from a single score per shot, so that score excursions earlier in a non-disruptive shot do not count as false alarms. In our protocol, a non-disruptive shot is scored by the maximum over its whole observed trajectory, every alarm episode contributes to the FAR, and the deadline metrics depend on when an alarm is first raised. The original implementation also used a learning rate of 2×10^{-2}, a fixed 40 ms truncation, and random clipping of non-disruptive training shots, whereas we use the common learning rate of 10^{-3} and select the truncation horizon from the same grid as the survival models. CCNN is therefore a reimplementation evaluated under the present protocol, not a reproduction of the published results.

B.5 survTLS Implementation

Let $r_{i,t} = R_{i,t}/\delta_h$ be the residual time in bins and $s_{i,t} = \max(r_{i,t}/\ell, \epsilon)$ the standard deviation of the smoothing Gaussian in bins, with $\ell = 10$ and $\epsilon = 10^{-8}$, and let $\Phi_{i,t}(x) = \Phi\big((x - r_{i,t})/s_{i,t}\big)$, where Φ is the standard normal distribution function. For a disruptive shot, the smoothed event-time mass over the $K = T_{\max}/\delta_h$ output bins is $\widetilde{f}_{i,t}(k) = \Phi_{i,t}(e_k) - \Phi_{i,t}(e_{k-1})$, for $k = 1, \ldots, K$, with bin edges $e_0 = -\infty$, $e_k = k + \frac{1}{2}$ for $1 \leq k < K$, and $e_K = +\infty$. The first bin therefore absorbs the Gaussian mass at negative and sub-bin residual times, and the last bin absorbs the mass beyond $T_{\max}$; the masses sum to one without further normalization. For non-disruptive shots, $\widetilde{f}_{i,t}(k) = 0$ for every k. Target hazards are clipped to $[0, 1]$, with the denominator of eq. (2) bounded below by ϵ. The loss in eq. (4) uses the first K_H of the

Table 10 Method-specific settings. All neural predictors except CCNN share the encoder and the common optimization settings in table 7.

Method	Objective	Settings	Learning rate
DSM	Weibull mixture, lower bound of $\mathcal{L}_{\mathrm{SA}}$	$M \in \{1, 3\}$; every valid landmark, loss averaged over time and then shots; 10,000 mixture-pretraining iterations at learning rate 10^{-2}; at most 210 epochs	10^{-3}
DCPH	Cox partial likelihood	Risk sets from all valid landmarks in a mini-batch of shots; at most 210 epochs	10^{-3}
survTLS	$\mathcal{L}^{(H)}_{\mathrm{surv}}$ with $K_H = \lceil H/\delta_h \rceil$ hazards	$H \leq 360\,\mathrm{ms}$; $\delta_h \in \{1, 5\}\,\mathrm{ms}$; smoothing length scale 10; prevalence-aware output-bias initialization; $\alpha = 0.025\,\mathrm{ms}^{-1}$ for the alarm-priority function; at most 210 epochs	10^{-4}, 10^{-3}
Hard EEP	Masked BCE with $y^{(H)}_{i,t}$	$H = h_{\min}$ (main) or $h_{\max}$ of the selected TLS bounds, with the TLS training cohort; positive-class weight 1; at most 210 epochs	10^{-3}
TLS	Masked BCE with $q^{\mathrm{TLS}}_{i,t}$	$\gamma = 0.025\,\mathrm{ms}^{-1}$; $(h_{\min}, h_{\max})$ from table 11; positive-class weight 1; at most 210 epochs	10^{-3}
RF	Random forest on $y^{(H)}_{i,t}$	$H \leq 360\,\mathrm{ms}$; 63 causal window features of the nine common signals; 500 trees; minimum leaf size 20; at most 200 uniformly sampled landmarks per training shot; no class weights	–
CCNN	Sequence-to-label BCE on $d_{i,t}$	Architecture of Arnold et al. [23]; disruptive training sequences truncated at $T_i - H$; at most 210 epochs	10^{-3}

Table 11 Candidate TLS smoothing intervals $(h_{\min}, h_{\max})$ in milliseconds and the eligibility horizon H_{elig} of their training cohort. The same 22 candidates are used on every device.

Intervals $(h_{\min}, h_{\max})$	H_{elig}
$(20, 120)$, $(40, 120)$, $(80, 120)$, $(120, 360)$	$h_{\min}$
$(0, 20)$, $(0, 40)$, $(0, 80)$, $(0, 120)$, $(40, 80)$, $(40, 120)$	$h_{\max}$
$(0, 240)$, $(0, 360)$, $(20, 40)$, $(20, 80)$, $(20, 240)$, $(20, 360)$, $(40, 240)$, $(40, 360)$, $(80, 240)$, $(80, 360)$, $(120, 240)$, $(240, 360)$	120

K bins, and the target masses are not renormalized after this truncation. The output bias of each hazard logit is initialized to the logit of the mean target hazard of that bin over the training landmarks. The values of δ_h, H, and $T_{\max}$ are listed in tables 10 and 11.

B.6 Checkpoint Selection and Threshold Calibration

For DSM, TLS, and survTLS, the within-run checkpoint is selected by monitor-split deadline AUROC at $w = 40\,\mathrm{ms}$. For seed 0, the configuration (training horizon, TLS smoothing interval, DSM mixture count, and survTLS bin width and learning rate) is then selected by monitor-split deadline F1 after threshold calibration on the calibration split. These configuration choices are frozen for seeds 1 and 2. Test data are not used for checkpoint, configuration, or alarm-policy selection.

Validation calibration evaluates at most 512 automatically generated threshold candidates. The alarm configuration is fixed to one-sample smoothing, one required consecutive threshold crossing, zero hysteresis, and zero cooldown; under this configuration it reduces to the rule in section C.2. The low, middle, and high budgets target approximately 0.5%, 1%, and 5% normal-shot FPR, respectively; the middle budget is primary. Normal-shot FPR is the fraction of non-disruptive shots with at least one alarm. For N_0 non-disruptive shots with total normal exposure D_0, target p_{FPR} is converted to the nominal event-based ceiling $C(p_{\mathrm{FPR}}) = p_{\mathrm{FPR}} N_0 (24\,\mathrm{h})/D_0$. Because FAR counts repeated alarm events, this conversion is an approximate FPR interpretation. The resulting low/middle/high ceilings are $(110, 220, 1100)$ for DIII-D, $(650, 1300, 6500)$ for

C-Mod, and (63, 126, 630) for EAST, in alarms per 24 normal observation hours. Test FAR is reported without clipping or post-hoc threshold adjustment. These monitor- and calibration-based selections define the validation-selected fixed-policy evaluation.

Appendix C Detailed Evaluation Definitions

C.1 Alarm Scores

For DSM, the score at landmark t is the cumulative event probability within the training horizon: $r_{i,t}^{\mathrm{DSM}}(H) = \int_0^H f_\Theta(\tau \mid \mathbf{z}_{i,t})\, \mathrm{d}\tau$. DCPH uses the corresponding horizon risk $1-S_\Theta(H \mid \mathbf{z}_{i,t})$ implied by its proportional-hazards model. For TLS and Hard EEP, the score is the model output, which is a smoothed horizon-aware risk score for TLS and an estimate of the fixed-horizon event probability for Hard EEP: $r_{i,t}^{\mathrm{TLS}}(H) = p_\theta^{(H)}(\mathbf{z}_{i,t})$. For survTLS, the horizon-resolved hazards defined in section 3.4 first give the cumulative event probability

$$F_\theta(k \mid \mathbf{z}_{i,t}) = 1 - \prod_{j=1}^{k} \left[1 - \lambda_\theta(j \mid \mathbf{z}_{i,t})\right], \quad \text{(C1)}$$

and the alarm score aggregates these horizon-specific risks after weighting them by imminence:

$$r_{i,t}^{\mathrm{survTLS}}(H) = \max_{1 \le k \le K_H} \psi_\alpha^{\exp}(k\delta_h) F_\theta(k \mid \mathbf{z}_{i,t}). \quad \text{(C2)}$$

Here, $\psi_\alpha^{\exp}$ is the exponential alarm-priority function

$$\psi_\alpha^{\exp}(\tau) = \begin{cases} \dfrac{e^{-\alpha\tau} - e^{-\alpha H}}{1 - e^{-\alpha H}}, & 0 < \tau \le H, \\ 0, & \tau > H, \end{cases} \quad \text{(C3)}$$

which decreases from approximately one at short offsets to zero at the horizon. The parameter $\alpha > 0$, in ms^{-1}, controls how strongly shorter offsets are prioritized; we use $\alpha = 0.025\,\mathrm{ms}^{-1}$ ($25\,\mathrm{s}^{-1}$). RF and CCNN use their predicted class probabilities.

C.2 Alarm Events and the Operational Deadline

For shot i, let $\mathcal{I}_i$ be the set of valid causal landmarks, let $\delta_i \in \{0, 1\}$ indicate whether a disruption is observed, and let T_i be the disruption time when $\delta_i = 1$. Method m produces an alarm score $r_{i,t}^{(m)}(H)$ at landmark t for training horizon H. Let ϕ denote the causal score averaging, persistence, and cooldown configuration selected on validation. The pointwise threshold state is $b_{i,t}(\eta) = \mathbb{I}\left\{r_{i,t}^{(m)}(H) \ge \eta\right\}$, and the fixed causal alarm operator Φ_ϕ emits $\mathcal{A}_i(\eta, \phi) = \Phi_\phi\left((r_{i,t}^{(m)}(H))_{t \in \mathcal{I}_i}, \eta\right)$. Each emitted time is counted as one alarm event; a sustained active state is not counted once per landmark.

For a required lead time $w > 0$, measured in milliseconds and defined independently of the training horizon H, the last eligible landmark of a disruptive shot is $t_i^{\mathrm{dl}}(w) = \max\{t \in \mathcal{I}_i : t \le T_i - w\}$. The shot is detected when at least one alarm is emitted no later than this deadline,

$$z_i(\eta, \phi; w) = \mathbb{I}\left\{\min \mathcal{A}_i(\eta, \phi) \le t_i^{\mathrm{dl}}(w)\right\}, \quad \text{(C4)}$$

with $\min \emptyset = \infty$. There is no lower time bound: an alarm earlier than $T_i - w$ remains a successful detection. An alarm after the deadline is late and does not detect the shot.

C.3 Deadline Recall, Alarm Precision, and F1

Only one true alarm is credited for each detected disruptive shot. Every remaining alarm event—including alarms on normal shots and late or duplicate alarms on disruptive shots—is false. Thus,

$$N_{\mathrm{TP}}(\eta, \phi; w) = \sum_{i:\delta_i=1} z_i(\eta, \phi; w), \quad \text{(C5)}$$

$$N_{\mathrm{FP}}(\eta, \phi; w) = \sum_i |\mathcal{A}_i(\eta, \phi)| - N_{\mathrm{TP}}(\eta, \phi; w), \quad \text{(C6)}$$

$$N_{\mathrm{FN}}(\eta, \phi; w) = \sum_i \delta_i - N_{\mathrm{TP}}(\eta, \phi; w). \quad \text{(C7)}$$

The fixed-policy metrics are

$$\mathrm{Rec}_w(\eta,\phi) = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}}+N_{\mathrm{FN}}}, \tag{C8}$$

$$\mathrm{Prec}_{\mathrm{alarm},w}(\eta,\phi) = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}}+N_{\mathrm{FP}}}, \tag{C9}$$

$$F_{1,w}(\eta,\phi) = \frac{2N_{\mathrm{TP}}}{2N_{\mathrm{TP}}+N_{\mathrm{FP}}+N_{\mathrm{FN}}}. \tag{C10}$$

When a denominator is zero, the corresponding precision or F1 is reported as undefined in a results table and treated as zero for model selection. The main experiment sets $w = 40\,\mathrm{ms}$.

C.4 FAR

FAR is computed only from non-disruptive shots. Let D_i be the valid observed duration of normal shot i, including its final sampling interval. The total normal observation time is $D_0 = \sum_{i:\delta_i=0} D_i$, and the number of false alarms on normal shots is $N_{\mathrm{FA},0}(\eta,\phi) = \sum_{i:\delta_i=0} |\mathcal{A}_i(\eta,\phi)|$. The normalized FAR is

$$\mathrm{FAR}_{24}(\eta,\phi) = \frac{N_{\mathrm{FA},0}(\eta,\phi)}{D_0}\,(24\,\mathrm{h})\,. \tag{C11}$$

Repeated alarm events on the same normal shot all contribute to FAR.

C.5 Threshold-Free Deadline AUROC and AUPRC

For a disruptive shot, the threshold-free deadline score is the largest risk available by the deadline; for a non-disruptive shot, it is the largest risk over the observed record:

$$v_i^{(w)} = \begin{cases} \max\limits_{t\in\mathcal{I}_i:\, t\le t_i^{\mathrm{dl}}(w)} r_{i,t}^{(m)}(H), & \delta_i = 1, \\ \max\limits_{t\in\mathcal{I}_i} r_{i,t}^{(m)}(H), & \delta_i = 0. \end{cases} \tag{C12}$$

A disruptive shot with no eligible landmark receives a score below the valid score range. For a threshold η, define the corresponding shot-level prediction by $\widehat{d}_i(\eta;w) = \mathbb{I}\{v_i^{(w)} \ge \eta\}$. Sweeping η over the score range gives the shot-level true-positive rate (TPR), FPR, and precision

$$\mathrm{TPR}_w(\eta) = \mathrm{Rec}_w^{\mathrm{shot}}(\eta) = \frac{\sum_i \delta_i \widehat{d}_i(\eta;w)}{\sum_i \delta_i}, \tag{C13}$$

$$\mathrm{FPR}_w(\eta) = \frac{\sum_i (1-\delta_i)\widehat{d}_i(\eta;w)}{\sum_i (1-\delta_i)}, \tag{C14}$$

$$\mathrm{Prec}_w^{\mathrm{shot}}(\eta) = \frac{\sum_i \delta_i \widehat{d}_i(\eta;w)}{\sum_i \widehat{d}_i(\eta;w)}. \tag{C15}$$

This shot-level precision is distinct from the alarm-event precision in eq. (C10). Let $\mathrm{ROC}_w(\xi)$ denote the ROC curve, expressed as TPR at FPR ξ along the threshold sweep, and let $\mathrm{PR}_w(\nu)$ denote the PR curve, expressed as shot-level precision at recall ν. Deadline AUROC and AUPRC are

$$\mathrm{AUROC}_{\mathrm{deadline}}(w) = \int_0^1 \mathrm{ROC}_w(\xi)\,\mathrm{d}\xi, \tag{C16}$$

$$\mathrm{AUPRC}_{\mathrm{deadline}}(w) = \int_0^1 \mathrm{PR}_w(\nu)\,\mathrm{d}\nu. \tag{C17}$$

For a finite evaluation set, the integrals are evaluated from the empirical curves obtained by sweeping the distinct values of $v_i^{(w)}$. These quantities depend only on the ranking of the deadline scores and are independent of the deployed alarm threshold.

C.6 Validation Calibration and Horizon Selection

Let $\mathcal{G}_{m,H}^{\mathrm{cal}}$ be the finite set of candidate threshold and alarm-post-processing configurations for method m and training horizon H, and let C be the device-specific FAR ceiling corresponding to a normal-shot FPR budget. The calibration policy is

$$(\eta_{m,H,C}^{\star}, \phi_{m,H,C}^{\star}) = \arg \max_{(\eta,\phi)\in\mathcal{G}_{m,H}^{\mathrm{cal}}} \mathrm{Rec}_w^{\mathrm{cal}}(\eta,\phi) \quad \text{subject to} \quad \mathrm{FAR}_{24}^{\mathrm{cal}}(\eta,\phi) \le C. \tag{C18}$$

Ties are resolved by higher alarm precision, lower FAR, and then a higher threshold. If no candidate satisfies the budget, the minimum-FAR candidate is retained. The calibrated policy is applied unchanged to the disjoint monitor split. Let $\mathcal{H}_m$ be the set of candidate configurations of method m (section 4.4), indexed here by H; for TLS, H stands for the smoothing interval and its cohort. For the primary middle budget, the

selected configuration is

$$H_m^\star = \arg\max_{H \in \mathcal{H}_m} F_{1,w}^{\mathrm{mon}}\left(\eta_{m,H,C_{\mathrm{mid}}}^\star, \phi_{m,H,C_{\mathrm{mid}}}^\star\right). \tag{C19}$$

with remaining ties resolved by monitor recall, alarm precision, lower FAR, shorter horizon, and fewer mixture components. The selection uses seed 0 and is frozen for seeds 1 and 2. Test data do not enter either selection.

C.7 Recall–FAR Curves

For the validation-selected checkpoint, configuration, and post-processing $\phi_m^\star$ of method m, the threshold is swept over every critical level of the test scores. Each threshold η gives one point $\left(\mathrm{FAR}_{24,m}^{\mathrm{test}}(\eta, \phi_m^\star),\ \mathrm{Rec}_{w,m}^{\mathrm{test}}(\eta, \phi_m^\star)\right)$ of the Recall–FAR curve. For averaging over seeds, each curve is resampled on a common logarithmic FAR grid, taking the largest recall whose FAR does not exceed each grid value. Because the threshold is swept with test labels, the curves describe the test score distribution and are not a deployable threshold-selection procedure. The curves for the selected configurations are shown in fig. 4.

Appendix D Supplementary Results

Table 12 gives the numerical values underlying fig. 2, together with Deadline AUROC and the components of Deadline F1. Table 13 gives the paired differences between TLS and all other main methods, including Hard EEP at both selected TLS bounds. Figure 4 shows the Recall–FAR curves, and table 14 summarizes the sensitivity of the fixed policy to the calibration split. Table 15 reports the ranking of every method at the deadlines of the horizon study. Table 16 compares the DSM and survTLS protocol variants, and table 17 reports the DisruptionBench metrics for comparison with previous benchmark results.

Table 12 Results at the $w = 40$ ms deadline (mean ± sample standard deviation over three training seeds; the values of the individual seeds are shown in fig. 2). Deadline AUPRC and AUROC are threshold-free; the remaining metrics use the validation-selected fixed policy. The configuration column gives the seed-0 choice of the training horizon H (ms), with the number of mixture components for DSM and the smoothing interval $(h_{\min}, h_{\max})$ (ms) for TLS. Bold entries mark the largest mean Deadline AUPRC and Deadline F1 within each device.

Device	Method	Config.	Deadline AUPRC	Deadline AUROC	Deadline Recall	Alarm precision	Deadline F1	Test FAR/24h
C-Mod	CCNN	20	0.430 ± 0.016	0.724 ± 0.001	0.094 ± 0.013	0.617 ± 0.113	0.162 ± 0.014	2,112 ± 1,285
	DSM	80, 3	0.581 ± 0.042	0.808 ± 0.009	0.098 ± 0.061	0.268 ± 0.042	0.139 ± 0.069	1,056 ± 634
	DCPH	120	0.562 ± 0.043	0.803 ± 0.007	0.127 ± 0.121	0.254 ± 0.118	0.162 ± 0.136	2,887 ± 2,520
	survTLS	80	0.600 ± 0.006	0.817 ± 0.007	0.103 ± 0.061	0.142 ± 0.062	0.118 ± 0.063	1,267 ± 762
	RF	240	**0.722** ± 0.003	0.884 ± 0.000	0.293 ± 0.009	0.324 ± 0.003	**0.308** ± 0.006	4,647 ± 1,118
	Hard EEP	240	0.691 ± 0.042	0.866 ± 0.022	0.138 ± 0.071	0.386 ± 0.017	0.197 ± 0.076	775 ± 976
	TLS	(240, 360)	0.696 ± 0.019	0.871 ± 0.017	0.228 ± 0.039	0.356 ± 0.052	0.278 ± 0.045	1,549 ± 800
DIII-D	CCNN	120	0.302 ± 0.072	0.703 ± 0.071	0.081 ± 0.071	0.376 ± 0.112	0.129 ± 0.106	338 ± 242
	DSM	120, 1	0.731 ± 0.021	0.904 ± 0.003	0.188 ± 0.091	0.294 ± 0.042	0.220 ± 0.066	94 ± 75
	DCPH	80	0.725 ± 0.014	0.908 ± 0.004	0.136 ± 0.064	0.184 ± 0.022	0.151 ± 0.052	216 ± 175
	survTLS	40	0.642 ± 0.007	0.849 ± 0.003	0.138 ± 0.012	0.309 ± 0.046	0.189 ± 0.003	6 ± 10
	RF	120	0.807 ± 0.005	0.933 ± 0.002	0.274 ± 0.076	0.391 ± 0.075	0.311 ± 0.038	39 ± 10
	Hard EEP	40	0.686 ± 0.060	0.903 ± 0.015	0.039 ± 0.010	0.053 ± 0.013	0.044 ± 0.011	44 ± 63
	TLS	(40, 120)	**0.819** ± 0.046	0.935 ± 0.018	0.409 ± 0.295	0.368 ± 0.229	**0.384** ± 0.262	238 ± 277
EAST	CCNN	120	0.613 ± 0.092	0.764 ± 0.053	0.290 ± 0.141	0.715 ± 0.170	0.385 ± 0.136	686 ± 557
	DSM	80, 3	0.850 ± 0.012	0.915 ± 0.007	0.222 ± 0.165	0.415 ± 0.102	0.272 ± 0.185	34 ± 23
	DCPH	120	0.922 ± 0.003	0.963 ± 0.003	0.331 ± 0.094	0.546 ± 0.009	0.407 ± 0.072	55 ± 37
	survTLS	360	0.931 ± 0.057	0.964 ± 0.027	0.340 ± 0.530	0.413 ± 0.447	0.350 ± 0.509	28 ± 48
	RF	80	0.907 ± 0.001	0.955 ± 0.000	0.217 ± 0.002	0.418 ± 0.012	0.286 ± 0.004	3 ± 5
	Hard EEP	120	0.939 ± 0.033	0.963 ± 0.017	0.437 ± 0.343	0.661 ± 0.118	0.479 ± 0.344	25 ± 23
	TLS	(120, 240)	**0.975** ± 0.025	0.984 ± 0.013	0.756 ± 0.201	0.815 ± 0.079	**0.781** ± 0.141	52 ± 56

Table 13 Differences between TLS and each comparator (TLS minus comparator) at $w = 40$ ms, with 95% paired-bootstrap intervals from 2,000 stratified test-shot resamples, averaging the three fixed training seeds. These intervals condition on the selected configurations and exclude training-seed and selection uncertainty; no multiplicity adjustment is applied. An asterisk marks intervals that exclude zero.

Device	Comparator	Δ Deadline AUPRC [95% CI]	Δ Deadline F1 [95% CI]
C-Mod	CCNN	+0.267 [+0.209, +0.319]*	+0.116 [+0.047, +0.179]*
	DSM	+0.116 [+0.070, +0.161]*	+0.139 [+0.096, +0.179]*
	DCPH	+0.135 [+0.089, +0.177]*	+0.116 [+0.075, +0.157]*
	survTLS	+0.096 [+0.043, +0.154]*	+0.160 [+0.105, +0.212]*
	RF	−0.025 [−0.074, +0.028]	−0.030 [−0.087, +0.023]
	Hard EEP ($H = h_{\min}$)	+0.005 [−0.013, +0.024]	+0.081 [+0.044, +0.118]*
	Hard EEP ($H = h_{\max}$)	−0.020 [−0.042, +0.002]	+0.069 [+0.036, +0.104]*
DIII-D	CCNN	+0.517 [+0.463, +0.563]*	+0.256 [+0.205, +0.305]*
	DSM	+0.087 [+0.042, +0.131]*	+0.165 [+0.123, +0.209]*
	DCPH	+0.094 [+0.048, +0.137]*	+0.233 [+0.187, +0.277]*
	survTLS	+0.176 [+0.122, +0.232]*	+0.195 [+0.147, +0.247]*
	RF	+0.012 [−0.028, +0.053]	+0.073 [+0.025, +0.123]*
	Hard EEP ($H = h_{\min}$)	+0.133 [+0.098, +0.162]*	+0.340 [+0.301, +0.377]*
	Hard EEP ($H = h_{\max}$)	+0.085 [+0.049, +0.118]*	+0.252 [+0.213, +0.292]*
EAST	CCNN	+0.361 [+0.335, +0.385]*	+0.396 [+0.360, +0.429]*
	DSM	+0.125 [+0.108, +0.142]*	+0.509 [+0.484, +0.536]*
	DCPH	+0.053 [+0.039, +0.068]*	+0.374 [+0.345, +0.404]*
	survTLS	+0.044 [+0.034, +0.054]*	+0.431 [+0.410, +0.453]*
	RF	+0.068 [+0.053, +0.083]*	+0.495 [+0.461, +0.532]*
	Hard EEP ($H = h_{\min}$)	+0.036 [+0.028, +0.044]*	+0.302 [+0.283, +0.320]*
	Hard EEP ($H = h_{\max}$)	+0.028 [+0.021, +0.035]*	+0.157 [+0.136, +0.177]*

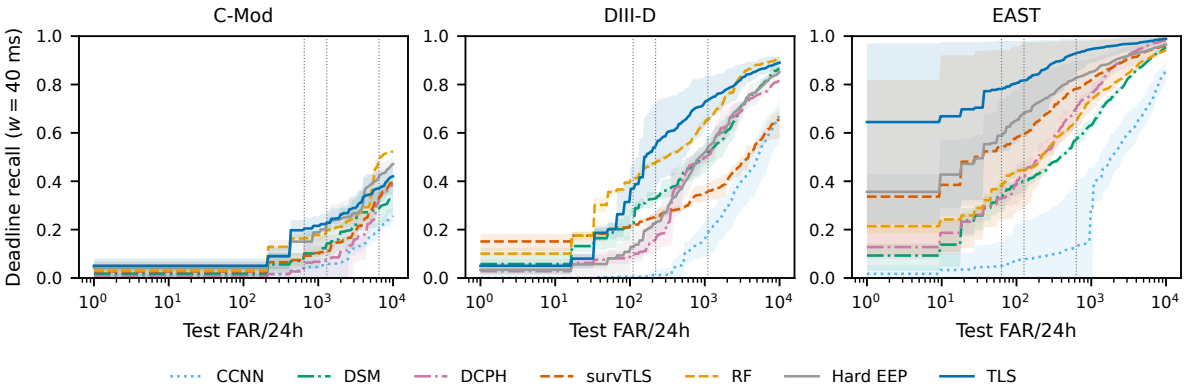


Fig. 4 Deadline Recall versus realized test FAR/24h at $w = 40$ ms, obtained by sweeping the threshold on the test data with the validation-selected post-processing fixed. Lines show the mean over three training seeds, and shaded bands the range. Dotted vertical lines mark the low, middle (primary), and high FAR budgets. The curves use test labels and are descriptive.

Table 14 Test Deadline F1 when the calibration shots are resampled 1,000 times with the checkpoint and test data fixed. Median and 2.5th–97.5th percentiles, averaged over three training seeds.

Device	Method	Median test F1	95% interval of test F1
C-Mod	CCNN	0.162	[0.051, 0.191]
	DSM	0.135	[0.022, 0.249]
	DCPH	0.157	[0.093, 0.268]
	survTLS	0.113	[0.084, 0.210]
	RF	0.293	[0.164, 0.359]
	Hard EEP	0.207	[0.059, 0.309]
	TLS	0.265	[0.055, 0.366]
DIII-D	CCNN	0.129	[0.124, 0.159]
	DSM	0.236	[0.169, 0.316]
	DCPH	0.184	[0.115, 0.313]
	survTLS	0.216	[0.192, 0.252]
	RF	0.316	[0.267, 0.336]
	Hard EEP	0.113	[0.079, 0.258]
	TLS	0.420	[0.288, 0.526]
EAST	CCNN	0.387	[0.373, 0.419]
	DSM	0.282	[0.092, 0.375]
	DCPH	0.415	[0.255, 0.542]
	survTLS	0.377	[0.343, 0.551]
	RF	0.286	[0.115, 0.286]
	Hard EEP	0.493	[0.377, 0.660]
	TLS	0.780	[0.614, 0.826]

Table 15 Deadline AUPRC of the selected configurations at different evaluation deadlines w (ms), averaged over three training seeds.

Device	Method	$w = 10$	$w = 40$	$w = 100$	$w = 300$
C-Mod	CCNN	0.436	0.430	0.394	0.281
	DSM	0.686	0.581	0.446	0.187
	DCPH	0.635	0.562	0.472	0.230
	survTLS	0.768	0.600	0.470	0.228
	RF	0.804	0.722	0.642	0.270
	Hard EEP	0.744	0.691	0.620	0.228
	TLS	0.745	0.696	0.626	0.279
DIII-D	CCNN	0.314	0.302	0.276	0.220
	DSM	0.812	0.731	0.611	0.436
	DCPH	0.795	0.725	0.631	0.504
	survTLS	0.718	0.642	0.556	0.404
	RF	0.886	0.807	0.648	0.377
	Hard EEP	0.871	0.686	0.448	0.300
	TLS	0.927	0.819	0.495	0.315
EAST	CCNN	0.616	0.613	0.606	0.584
	DSM	0.894	0.850	0.743	0.559
	DCPH	0.943	0.922	0.871	0.770
	survTLS	0.961	0.931	0.852	0.479
	RF	0.953	0.907	0.784	0.547
	Hard EEP	0.963	0.939	0.835	0.432
	TLS	0.984	0.975	0.934	0.403

Table 16 Protocol variants. DSM with one sampled landmark per shot per epoch versus every landmark (main), and survTLS with its original setting ($\delta_h = 1\,\text{ms}$, $\eta = 10^{-4}$) versus the reselected configuration (main): (H, δ_h, η) is $(40\,\text{ms}, 1\,\text{ms}, 10^{-4})$ on DIII-D, $(80\,\text{ms}, 5\,\text{ms}, 10^{-4})$ on C-Mod, and $(360\,\text{ms}, 5\,\text{ms}, 10^{-3})$ on EAST. Both survTLS variants use the corrected at-risk loss weight. Deadline AUPRC and Deadline F1 at $w = 40\,\text{ms}$, averaged over three training seeds.

Device	Variant	H, M	AUPRC	F1
C-Mod	DSM, sampled	40, 1	0.522	0.145
	DSM, all (main)	80, 3	0.581	0.139
	survTLS, original	80, –	0.611	0.141
	survTLS, resel. (main)	80, –	0.600	0.118
DIII-D	DSM, sampled	20, 1	0.648	0.203
	DSM, all (main)	120, 1	0.731	0.220
	survTLS, original	120, –	0.653	0.180
	survTLS, resel. (main)	40, –	0.642	0.189
EAST	DSM, sampled	80, 3	0.817	0.144
	DSM, all (main)	80, 3	0.850	0.272
	survTLS, original	20, –	0.761	0.080
	survTLS, resel. (main)	360, –	0.931	0.350

Table 17 DisruptionBench metrics at the validation-selected threshold (equal high and low thresholds, one-sample hysteresis, $t_{\text{useful}} = 40\,\text{ms}$), averaged over three training seeds.

Device	Method	AUC (Zhu)	TPR	FPR	F1
C-Mod	CCNN	0.722	0.094	0.014	0.164
	DSM	0.808	0.098	0.004	0.172
	DCPH	0.803	0.127	0.010	0.205
	survTLS	0.817	0.103	0.006	0.179
	RF	0.884	0.293	0.014	0.435
	Hard EEP	0.866	0.138	0.003	0.234
	TLS	0.871	0.228	0.008	0.362
DIII-D	CCNN	0.702	0.081	0.015	0.131
	DSM	0.904	0.188	0.002	0.307
	DCPH	0.899	0.136	0.003	0.232
	survTLS	0.849	0.138	0.000	0.242
	RF	0.933	0.274	0.002	0.423
	Hard EEP	0.902	0.039	0.001	0.074
	TLS	0.935	0.409	0.007	0.516
EAST	CCNN	0.698	0.290	0.041	0.402
	DSM	0.915	0.222	0.002	0.339
	DCPH	0.963	0.331	0.003	0.489
	survTLS	0.964	0.340	0.001	0.367
	RF	0.955	0.217	0.000	0.357
	Hard EEP	0.963	0.437	0.001	0.543
	TLS	0.984	0.756	0.003	0.848

References

[1] Rea, C., Montes, K.J., Erickson, K.G., Granetz, R.S., Tinguely, R.A.: A real-time machine learning-based disruption predictor in DIII-D. Nuclear Fusion **59**(9), 096016 (2019) https://doi.org/10.1088/1741-4326/ab28bf

[2] Keith, Z., Nagpal, C., Rea, C., Tinguely, R.A.: Risk-aware framework development for disruption prediction: Alcator C-Mod and DIII-D survival analysis. Journal of Fusion Energy **43**, 21 (2024) https://doi.org/10.1007/s10894-024-00413-y

[3] Spangher, L., Bonotto, M., Arnold, W., Chayapathy, D., Gallingani, T., Spangher, A., Cannarile, F., Bigoni, D., Marchi, E., Rea, C.: DisruptionBench and complimentary new models: Two advancements in machine learning driven disruption prediction. Journal of Fusion Energy **44**, 26 (2025) https://doi.org/10.1007/s10894-025-00495-2

[4] Kates-Harbeck, J., Svyatkovskiy, A., Tang, W.: Predicting disruptive instabilities in controlled fusion plasmas through deep learning. Nature **568**(7753), 526–531 (2019) https://doi.org/10.1038/s41586-019-1116-4

[5] Zhu, J., Rea, C., Montes, K., Granetz, R., Sweeney, R., Tinguely, R.A.: Hybrid deep-learning architecture for general disruption prediction across multiple tokamaks. Nuclear Fusion **61**(2), 026007 (2021) https://doi.org/10.1088/1741-4326/abc664

[6] Kalbfleisch, J.D., Prentice, R.L.: The Statistical Analysis of Failure Time Data, 2nd edn. John Wiley & Sons, Hoboken, NJ (2002). https://doi.org/10.1002/9781118032985

[7] Tinguely, R.A., Montes, K.J., Rea, C., Sweeney, R., Granetz, R.S.: An application of survival analysis to disruption prediction via random forests. Plasma Physics and Controlled Fusion **61**(9), 095009 (2019) https://doi.org/10.1088/1361-6587/ab32fc

[8] Faraggi, D., Simon, R.: A neural network model for survival data. Statistics in Medicine

14(1), 73–82 (1995) https://doi.org/10.1002/sim.4780140108

[9] Nagpal, C., Li, X.R., Dubrawski, A.: Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. IEEE Journal of Biomedical and Health Informatics **25**(8), 3163–3175 (2021) https://doi.org/10.1109/JBHI.2021.3052441

[10] Yèche, H., Burger, M., Veshchezerova, D., Rätsch, G.: Dynamic survival analysis for early event prediction. In: Pollard, T., Choi, E., Singhal, P., Hughes, M., Sizikova, E., Mortazavi, B., Chen, I., Wang, F., Sarker, T., McDermott, M., Ghassemi, M. (eds.) Proceedings of the Fifth Conference on Health, Inference, and Learning. Proceedings of Machine Learning Research, vol. 248, pp. 540–557. PMLR, New York, NY, USA (2024). https://proceedings.mlr.press/v248/yeche24a.html

[11] Tomašev, N., Glorot, X., Rae, J.W., *et al.*: A clinically applicable approach to continuous prediction of future acute kidney injury. Nature **572**(7767), 116–119 (2019) https://doi.org/10.1038/s41586-019-1390-1

[12] Hyland, S.L., Faltys, M., Hüser, M., Lyu, X., Gumbsch, T., Esteban, C., Bock, C., Horn, M., Moor, M., Rieck, B., *et al.*: Early prediction of circulatory failure in the intensive care unit using machine learning. Nature Medicine **26**(3), 364–373 (2020) https://doi.org/10.1038/s41591-020-0789-4

[13] Yèche, H., Pace, A., Rätsch, G., Kuznetsova, R.: Temporal label smoothing for early event prediction. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 39913–39938. PMLR, Honolulu, HI, USA (2023). https://proceedings.mlr.press/v202/yeche23a.html

[14] Saito, T., Rehmsmeier, M.: The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE **10**(3), 0118432 (2015) https://doi.org/10.1371/journal.pone.0118432

[15] Windsor, C.G., Pautasso, G., Tichmann, C., Buttery, R.J., Hender, T.C., JET EFDA Contributors, ASDEX Upgrade Team: A cross-tokamak neural network disruption predictor for the JET and ASDEX Upgrade tokamaks. Nuclear Fusion **45**(5), 337–350 (2005) https://doi.org/10.1088/0029-5515/45/5/004

[16] Cannas, B., Delogu, R.S., Fanni, A., Sonato, P., Zedda, M.K., JET-EFDA Contributors: Support vector machines for disruption prediction and novelty detection at JET. Fusion Engineering and Design **82**(5–14), 1124–1130 (2007) https://doi.org/10.1016/j.fusengdes.2007.07.004

[17] Murari, A., Vagliasindi, G., Arena, P., Fortuna, L., Barana, O., Johnson, M., JET-EFDA Contributors: Prototype of an adaptive disruption predictor for JET based on fuzzy logic and regression trees. Nuclear Fusion **48**(3), 035010 (2008) https://doi.org/10.1088/0029-5515/48/3/035010

[18] Murari, A., Vega, J., Rattá, G.A., Vagliasindi, G., Johnson, M.F., Hong, S.H., JET-EFDA Contributors: Unbiased and non-supervised learning methods for disruption prediction at JET. Nuclear Fusion **49**(5), 055028 (2009) https://doi.org/10.1088/0029-5515/49/5/055028

[19] Rattá, G.A., Vega, J., Murari, A., Vagliasindi, G., Johnson, M.F., Vries, P.C., JET-EFDA Contributors: An advanced disruption predictor for JET tested in a simulated real-time environment. Nuclear Fusion **50**(2), 025005 (2010) https://doi.org/10.1088/0029-5515/50/2/025005

[20] Rea, C., Granetz, R.S., Montes, K., Tinguely, R.A., Eidietis, N., Hanson, J.M., Sammuli, B.: Disruption prediction investigations using machine learning tools on DIII-D and Alcator C-Mod. Plasma Physics and Controlled Fusion **60**(8), 084004 (2018) https://doi.org/10.1088/1361-6587/aac7fe

[21] Churchill, R.M., Tobias, B., Zhu, Y., DIII-D Team: Deep convolutional neural networks for multi-scale time-series classification and application to tokamak disruption prediction using raw, high temporal resolution diagnostic data. Physics of Plasmas **27**(6), 062510 (2020) https://doi.org/10.1063/1.5144458

[22] Aymerich, E., Sias, G., Pisano, F., Cannas, B., Carcangiu, S., Sozzi, C., Stuart, P.J., Carvalho, A.F., *et al.*: Disruption prediction at JET through deep convolutional neural networks using spatiotemporal information from plasma profiles. Nuclear Fusion **62**(6), 066005 (2022) https://doi.org/10.1088/1741-4326/ac525e

[23] Arnold, W.F., Spangher, L., Rea, C.: Continuous convolutional neural networks for disruption prediction in nuclear fusion plasmas. arXiv preprint arXiv:2312.01286 (2023)

[24] Olofsson, K.E.J., Humphreys, D.A., La Haye, R.J.: Event hazard function learning and survival analysis for tearing mode onset characterization. Plasma Physics and Controlled Fusion **60**(8), 084002 (2018) https://doi.org/10.1088/1361-6587/aac662

[25] Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American Statistical Association **53**(282), 457–481 (1958) https://doi.org/10.1080/01621459.1958.10501452

[26] Cox, D.R.: Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) **34**(2), 187–202 (1972) https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

[27] Gensheimer, M.F., Narasimhan, B.: A scalable discrete-time survival model for neural networks. PeerJ **7**, 6257 (2019) https://doi.org/10.7717/peerj.6257

[28] Lee, C., Zame, W.R., Yoon, J., Schaar, M.: DeepHit: A deep learning approach to survival analysis with competing risks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018). https://doi.org/10.1609/aaai.v32i1.11842

[29] Kvamme, H., Borgan, Ø., Scheel, I.: Time-to-event prediction with neural networks and Cox regression. Journal of Machine Learning Research **20**(129), 1–30 (2019)

[30] Chen, G.H.: An introduction to deep survival analysis models for predicting time-to-event outcomes. Foundations and Trends in Machine Learning **17**(6), 921–1100 (2024) https://doi.org/10.1561/2200000114

[31] Houwelingen, H.C.: Dynamic prediction by landmarking in event history analysis. Scandinavian Journal of Statistics **34**(1), 70–85 (2007) https://doi.org/10.1111/j.1467-9469.2006.00529.x

[32] Lee, C., Yoon, J., Schaar, M.: Dynamic-DeepHit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. IEEE Transactions on Biomedical Engineering **67**(1), 122–133 (2020) https://doi.org/10.1109/TBME.2019.2909027

[33] Harutyunyan, H., Khachatrian, H., Kale, D.C., Ver Steeg, G., Galstyan, A.: Multitask learning and benchmarking with clinical time series data. Scientific Data **6**, 96 (2019) https://doi.org/10.1038/s41597-019-0103-9

[34] Yèche, H., Kuznetsova, R., Zimmermann, M., Hüser, M., Lyu, X., Faltys, M., Rätsch, G.: HiRID-ICU-Benchmark: A comprehensive machine learning benchmark on high-resolution ICU data. arXiv preprint arXiv:2111.08536 (2021) https://doi.org/10.48550/arXiv.2111.08536

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- horizonawareeep.zip