Hybrid deep learning architecture for general disruption prediction across tokamaks

J.X. Zhu, C. Rea, K. Montes, R.S. Granetz, R. Sweeney, R.A. Tinguely

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

Abstract: In this letter, we present a new disruption prediction algorithm based on Deep Learning that effectively allows knowledge transfer from existing devices to new ones, while predicting disruptions using very limited disruptive data from the new devices. Future fusion reactors will need to run disruption-free or with very few unmitigated disruptions. The algorithm presented in this letter achieves high predictive accuracy on C-Mod, DIII-D and EAST tokamaks with limited hyperparameter tuning. Through numerical experiments, we show that good accuracy (AUC=0.959) is achieved on EAST predictions by including a small number of disruptive discharges, thousands of non-disruptive discharges from EAST, and combining this with more than a thousand discharges from DIII-D and C-Mod. This holds true for all permutations of the three devices. This cross-machine data-driven study finds that non-disruptive data is machine-specific while disruptions are machine-independent.

Introduction – Utilizing nuclear fusion energy via magnetic-confinement tokamaks is one of a few encouraging paths toward future sustainable energy. Along the way, scientists need to learn to avoid plasma disruptions: these sudden and unexpected plasma terminations still represent one of the key challenges for tokamak devices, as their deleterious consequences can damage the whole devices and prevent the realization of a burning plasma reactor. Forecasting plasma instabilities and disruptions using first-principle models has demonstrated to be extremely difficult due to the complexity of the problem [1]. Whereas in fact, recent statistical and machine learning (ML) approaches based on experimental data have shown attractive results for disruption prediction in real-time systems [2-10]. Different tokamak devices have different operational spaces, spatiotemporal scales for physics events and plasma diagnostics. Therefore, most of the previous approaches were developed and optimized specifically for one device and did not show promising cross-device prediction ability [2-6, 8, 10]. Specifically, cross-machine studies such as [7] focused on neural networks that were trained on datasets purely or mostly from one device: these predictors achieved great performances on the training device but lacked the generalization capabilities derived from an understanding of the underlying physics, and therefore tended to fail on new, unseen device data. This is a well-known problem which is called the shortcut learning in the ML community. For example, a neural network might be able to perfectly classify cows in the training set—but it performs badly when tested on pictures where cows appear outside the grass background, revealing “grass” as a shortcut predictor for “cow” [11].

In this letter, we discuss a new Deep Learning (DL) application to disruption prediction that allows effective knowledge transfer from existing devices to new ones using very limited disruptive data from the new devices, while retaining high accuracy on the individual datasets. To this aim, we selected a set of disruption-relevant physical signals, available on all of the analyzed tokamak devices, and developed a powerful general algorithm using large databases from three tokamaks: Alcator C-Mod, DIII-D and EAST [8] (see the supplementary material for detailed datasets description). In addition, we combined data from the three different devices to add some randomization to the training domain which can alleviate over-learning of one specific device's behavior and designed a set of cross-machine experiments to find general guidelines for disruption prediction on new devices using very limited disruptive data from themselves. Moreover, unlike previous studies [7, 9-10], the three machines have very different features: EAST is a medium size ($R=1.85m, a=0.45m^2$) superconducting tokamak with a hybrid first wall, its lower divertor is in Carbon, the middle wall Molybdenum (Mo), while the upper divertor is made of Tungsten [12]. DIII-D is a medium size ($R=1.67m, a=0.67m$) tokamak with a carbon wall and relatively big error field:

1\(R\) and \(a\) are respectively the major and minor radius of the toroidal device.
most of disruptive shots in our DIII-D database contain a locked mode as the last precursor in their event chain toward disruption [13-14]. C-Mod is a small size tokamak ($R=0.68$m, $\sigma=0.22$m) with high energy density (plasma pressure up to 2.05atm), high magnetic field ($B_t$ up to 8T) and high-Z metal (Mo) wall. The combination of these different characters covers a substantial fraction of ITER’s features [15-16]. A cross-machine study using data from these devices is well suited for investigating disruption prediction solutions for ITER.

**The Hybrid Deep Learning (HDL) architecture** — Figure 1(a) shows the architecture of the hybrid neural network used for cross-machine disruption prediction. The network consists of two Gated Recurrent Unit (GRU) layers [17], one fully connected layer and three novel Multi-Scale Temporal Convolution (MSTConv) layers plus the input and the classification layer. The MSTConv layer is inspired from work in machine translation [18] and the detailed structure of one MSTConv layer is shown in Figure 1(b). It consists of six 1-D causal convolution layers [19] with different window lengths $L$ from one to six. The first 1-D convolution layer can only access the current time step $t_0$. The $L$-th 1-D convolution layer can look at $L$ time steps from $t_{0+L-1}$ to $t_0$. This structure enables different 1-D convolution layers to capture local temporal information at different levels ($1^{st}$ order time derivative, $2^{nd}$ order time derivative...). The resulting outputs from these six layers are concatenated and then processed through a batch normalization layer [20] and a ReLU activation to develop new features. It is important to highlight that different parts of the HDL architecture serve different purposes: The first two MSTConv layers are used to extract local temporal patterns from the input plasma sequences to form a richer representation of the input space. The following two GRU layers — with their long-term memory capability - can capture the long-range dependencies across different signals in the sequences. Then the following MSTConv and fully connected layers can compress and summarize the output representation from the GRU layers to a 12-dimension latent encoding\(^2\) which can be mapped to the output by the classification layer.

![Diagram of HDL architecture and MSTConv layer](image)

**Figure 1:** The HDL architecture (a) and the detailed structure of MSTConv layer (b). Notice that it consists 6 1-D causal convolution layers with window lengths $L$ from 1 to 6.

A shot-by-shot testing scheme was designed following [8] to simulate alarms triggered in the Plasma Control System (PCS), using the test shots from different devices. Given an input plasma sequence $S$ (10x12 matrix), the predictor maps $S$ to a ‘disruptivity’ between 0 to 1 at the last time step of the sequence where 1 is the disruptive class and 0 is the non-disruptive class. During testing, the whole flattop phase of each test shot is being subdivided in batches of 10 step sequences, given the HDL architecture design. Each neighboring testing sequence will have 9 steps overlap, and there are N-9 sequences for a test shot with N steps. If the disruptivity exceeds a preset threshold at any test time step, the test shot is classified

\(^2\) The dimension of final latent space is a tunable hyperparameter.
as disruptive and the warning time is recorded for truly disruptive shots, defined as the difference between the alarm time and the final current quench. A successfully detected disruption on C-Mod is shown in Figure 2: under a binary classification scheme, this is regarded as true positive (TP), while false positives (FP) correspond to a false warning, or a healthy plasma being terminated. This latter situation can still lead to some machine damage, but on the other side, being unable to predict a disruption early enough (false negative, FN) is even more costly because it prevents any damage control of disruption consequences. A trade-off can be achieved by adjusting the alarm threshold of the disruptivity, as visually demonstrated by a receiver–operator characteristic (ROC) curve [21]. The area under the ROC curve (AUC) is used as performance metric for the HDL predictor. Throughout the paper, we evaluate the predictive performances on all tokamaks at 50 ms before the disruption event: this is chosen as the minimum warning time for successful disruption mitigation on future devices [22].

The HDL predictor successfully achieves state-of-the-art performance on all three test sets (figure 3) comparable to other fully-optimized deep neural network disruption predictors [9]. To carry on a comparison with previous approaches to disruption prediction developed by the authors, we report the performance of the Random Forest (RF) predictors [6, 8] that are specifically optimized for each machine. The HDL predictor exceeds RF performances on all three datasets and shows the strong applicability and generalization power of the model. Besides its impressive performance, the inference time of our model is very short, allowing it to make a prediction in roughly 1ms using an 8-core CPU. This fast and novel model is not only an important step towards the prediction requirement of future devices, but also suggests a powerful conclusion. Although different devices may have disjoint operational regimes, there seems to exist a common type of discriminant function – same model hyperparameters - capable of separating the disruptive from non-disruptive phase on all these machines.

![Figure 2: A successfully detected disruption on C-Mod.](image)

![Figure 3: The ROC curves from test sets for HDL model and the Random Forest (RF) model, for C-Mod, DIII-D and EAST. (a) C-Mod; (b) DIII-D; (c) EAST.](image)

**Cross-machine study** – The availability of a huge amount of experimental data across several tokamaks allows us to design numerical experiments to test transfer learning capabilities of the HDL architecture. Future reactors like ITER cannot tolerate more than a few unmitigated disruptions [1], so we must be able to predict their disruptions given very limited disruptive data from themselves. Recently published work by Kates-Harbeck J. *et al.* [9] demonstrates the potential of DL-based predictors in learning general representation of experimental data that can be used in cross-machine applications. Expanding from this valuable work, we have designed complete numerical experiments to test transfer learning capabilities of the HDL architecture. Given the availability of a large database of aggregated data from different
tokamaks, it is tempting to verify if and how useful the data from existing devices is to be able to predict unstable plasmas on a new device. In the following section, we consider C-Mod and DIII-D as ‘existing machines’ and investigate the effect of their data for the HDL disruption predictor when used on EAST, chosen as a ‘new device’. However, all following qualitative conclusions are machine-independent: they always hold no matter which device is elected as the ‘new device’. The remaining two cases can be found in the supplementary material.

The first set of cross-machine experiments was conducted using limited disruptive training shots from the new device. This practice reconnects and builds upon approaches already presented in literature [9, 10], where it was found that limited disruptive data from the target device can boost performances of data-driven cross-machine predictors. Beyond performances, we are rather interested in investigating how data from different existing devices influences predictions of disruptions on a new one, and in particular if any effect can be linked to general, device-independent knowledge.

The results of these first numerical experiments are shown in figure 4(a)-(b). In the first experiment, the disruption predictor was trained on 0.9% of all EAST disruptive shots (20 randomly selected discharges) and all EAST non-disruptive data (5995 shots) plus all disruptive shots from C-Mod (929) and DIII-D (1049). This combination achieved the best performances on the EAST test dataset: AUC=0.959. In the second and third experiment, we first remove all EAST disruptive shots and then 50% of EAST non-disruptive shots from the first training dataset, separately. In the fourth experiment, the predictor was trained only using selected EAST data (20 disruptive shots, 5995 non-disruptive shots), this being our baseline model. In the fifth experiment, we add all non-disruptive shots from C-Mod and DIII-D to the first training dataset. In the sixth experiment, the predictor was trained only on DIII-D and C-Mod and its low performance highlights the importance of non-disruptive data from the target machine. From these numerical experiments, it is possible to draw the following conclusions:

1. HDL achieves relatively good performances on a new device if using a few disruptive shots and many non-disruptive shots from the new device plus many disruptive data from existing devices. All components mentioned above are necessary because removing any of them will decrease the performance (cases 1 to 4 in figure 4(a)).

2. Non-disruptive data from existing devices is harmful to HDL performance but disruptive data from existing devices improves the predictive power (cases 1, 4, 5 in figure 4(b)).

3. Non-disruptive data from the target device can substantially improve the predictive power (case 6 in figure 4(b)).

To further investigate the effect of the class imbalance in the training set, we conducted another set of experiments using all disruptive training shots of the new device. The results are reported in figure 5(a)-(b): Again, in the first experiment, the disruption predictor was trained on all EAST disruptive and non-disruptive shots, including all disruptive shots from C-Mod and DIII-D and it achieves the best performance on the EAST test dataset: AUC=0.983. In the second experiment, we add all non-disruptive shots from C-Mod and DIII-D to the first training dataset. In the third experiment, the predictor is trained only on EAST data which is a reference case for comparison. In experiments 4-6 (figure 5(b)), we randomly remove 2/3 of EAST non-disruptive training shots, thus reducing the non-disruptive training data to be less than disruptive training data, i.e. an inversely imbalanced situation. The test results from figure 5(a)-(b) point to the following further conclusions:

4. Adding disruptive data from existing machines can still improve test performances on the new device even though you have abundant new machine data (cases 1, 3 in figure 5(a)). However, adding non-disruptive data from existing machines is still harmful in this situation (cases 1, 2 in figure 5(a)).
5. The effects of disruptive data (positive) and non-disruptive data (negative) do not result from the class imbalance of the new machine dataset, because disruptive data from existing device continually has positive effects while the non-disruptive data still has negative effects in the inversely imbalanced situation (figure 5(b)). **This difference between disruptive and non-disruptive data is machine-independent**, i.e. a universal conclusion.

6. Also, removing non-disruptive data from the target device will always decrease the test performance no matter how imbalanced the target dataset is (cases 1, 3 in figure 4(a), case 6 in figure 5(b)).

Therefore, considering all the results above, disruptive data from existing devices can improve the performance on the new device while the non-disruptive data seems to have negative effects and these do not result from the label imbalance of training datasets. This suggests that the non-disruptive data is specific to one device, but disruptive data contains some general knowledge about disruptions dynamics that could be transferred to a new device.

![Figure 4: ROC curves from the EAST test set using 0.9% of EAST disruptive training shots. In figure (a), we show a curve (magenta) without any disruptive EAST training data, while in (b) we show a curve without any EAST training data (green).](image1)

![Figure 5: ROC curves from the EAST test set using 100% of EAST disruptive training shots. Figure (a) reports ROCs obtained including 100% non-disruptive EAST data, while (b) shows results when reducing to 33% the non-disruptive EAST training data.](image2)

Probability distributions of plasma signals for the different machines can be used to investigate the different features of non-disruptive and disruptive data. Figure 6(a) shows the distributions of non-disruptive poloidal beta ($\beta_{pol}$) signal on the three tokamaks, which exhibit clear differences across devices. The distributions of all other non-disruptive input signals show substantial distinction across different devices, too. This striking difference of the distributions of each individual non-disruptive signal across tokamaks reflects the fact that different devices have disjoint operational regimes. From the data-driven perspective, this further implies that finding a numerical transformation that maps a set of signals
from DIII-D to EAST and vice versa is very challenging without incorporating machine-specific information, and this might indeed pose a great challenge when comparing ITER operational space to all existing devices. Due to all these considerations, we are inclined to conclude that non-disruptive data of existing devices is machine-specific and will only decrease the accuracy of the predictive models on the new device when it is directly mixed with data from the target device. Nevertheless, different devices show similar behavior when operating close to a disruption, as shown in Figures 6(b-d). The distributions of disruptive $li$ signal on three machines exhibit a similar trend: a noticeable shift towards positive values, as the plasma gets closer to the disruption event. The plasma normalized internal inductance $li$ will increase when a disruption is imminent, and it happens consistently on different tokamaks. We also observe similar behavior across the three different datasets for six (of twelve) input signals: $ip$-error-fraction, locked-mode-proxy, $v$-loop, betap, kappa, $z$-error-proxy. These similar behaviors explain the general knowledge about disruptions hidden beneath the disruptive data, which can be transferred to a new device.

![Figure 6: Distributions of plasma signals. (a) Non-disruptive $\beta$-tap signal on three machines; (b) disruptive $li$ signal on C-Mod; (c) disruptive $li$ signal on DIII-D; (d) disruptive $li$ signal on EAST. Each signal is shown in its normalized form (see the supplementary material for the details of normalization and signal description).](image)

**Summary and future plans** -- In this letter, we have presented a new, powerful disruption prediction algorithm based on Deep Learning and also a general, effective way to transfer knowledge from existing devices to new devices which offers a guideline for disruption prediction on new devices using very limited disruptive data from target new devices. The cross-machine study on Alcator C-Mod, DIII-D, and EAST shows that the relatively good prediction performances on target devices can be achieved using a small set of disruptive shots while thousands of non-disruptive shots from themselves, and that such performances improve when including hundreds of disruptive discharges from other devices, no matter which device is chosen as the target new device. Furthermore, disruptive and non-disruptive data are found to have a different impact on the disruption prediction framework which implies that non-disruptive data is machine-specific and the disruptive data contains general knowledge about disruptions, and the distributions of plasma signals have been investigated to further support this conclusion. These results are an important milestone for disruption prediction research for next-generation burning plasma reactors, such as ITER. Future efforts will focus on two main topics. Firstly, the precision of our hyperparameter scan is limited by our computing power: Given enough computational resources, we can conduct a fine hyperparameter tuning which might increase the performance of the predictor and find new insights in the cross-machine study. Secondly, in the future version of HDL model we will explore how to directly incorporate device features such as minor radius, major radius, toroidal magnetic field, wall material, etc. The machine-specific character of non-disruptive data suggests that it could be beneficial to
mix the device-specific representation with the plasma signal representation to increase the model’s expressive power. This may enable us to extract information from machine-specific non-disruptive data and benefit the prediction on the new devices. We believe that the continuation of this project will let us contribute to the development of ITER disruption mitigation system, by designing a robust disruption prediction framework for ITER.

**Acknowledgments** -- This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Awards DE-FC02-04ER54698 and DE-SC0014264. The HDL architecture reported in the paper was developed using TensorFlow library [23]. Part of the data analysis reported in this paper was performed using the OMFIT integrated modelling framework [24]. The authors are grateful to Nik Logan of PPPL; Earl Marmar of MIT; Richard Buttery and Chuck Greenfield of General Atomics for their support and valuable discussions.

**Disclaimer** -- This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

**Reference:**

[1] De Vries P. C. et al. 2016 *Fus. Sci. Technol.* 69, 471–484

[2] Wroblewski et al. 1997 *Nucl. Fusion* 37, 725.

[3] Cannas B. et al. 2004 *Nucl. Fusion* 44, 68.

[4] Murari, A. et al. 2008 *Nucl. Fusion* 48, 035010.

[5] Murari A. et al. 2018 *Nucl. Fusion* 58

[6] Rea C. et al. 2018 *Plasma Phys. Control. Fusion* 60, 084004

[7] Windsor C. G. et al. 2005 *Nucl. Fusion* 45

[8] Montes K. J. et al. 2019 *Nucl. Fusion* 59 096015.

[9] Kates-Harbeck J. et al. 2019 *Nature* 568, 526–531

[10] Rattá G. A. et al. 2018 *Fus. Sci. Technol.* 74

[11] Beery S. et al. 2018 *Proceedings of the European Conference on Computer Vision*, 456–473

[12] Xu Y. et al. 2014 *Phys. Scr.* 2014 014008

[13] Turco F. et al. 2018 *Nucl. Fusion* 58 106043

[14] Sweeney R. et al. 2017 *Nucl. Fusion* 57 016019

[15] ITER Physics Basis Editors et al. 1999 *Nucl. Fusion* 39

[16] Ikeda K. et al. 2007 *Nucl. Fusion* 39
[17] Cho K. et al. 2014 *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*.

[18] Lee J. et al. 2017 *Transactions of the Association for Computational Linguistics*, Volume 5, 365-378

[19] Oord A. V. D. et al. 2016 WaveNet: A Generative Model for Raw Audio

[20] Ioffe S. et al. 2015 *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Volume 37, 448-456

[21] Bradley A. P. 1997 *Pattern Recognition*, 30, 1145–1159.

[22] Hollmann E. et al. 2015 *Physics of Plasmas* 22

[23] Abadi M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015 tensorflow.org

[24] Meneghini O. et al. 2015 *Nucl. Fusion* 55 083008
Supplementary Material:

1. Dataset description

The choice of which parameters to include in the databases is guided by our knowledge of the plasma physics inherent in disruption phenomena, as well as the accessibility and consistency of these parameters on all three machines. Many of the disruption-relevant parameters included in this study are influenced by several papers published in literature [S1-S3]. The considered signals for the predictive models reported in this letter and their definition can be found in supplementary table 1, while the composition of the three training datasets is shown in supplementary table 2. Given these databases, we formalize the disruption prediction problem in a sequence-to-label supervised machine learning framework, where we assign a label to each example (10 step consecutive sequence in time of 12 plasma signals) and train an algorithm to learn the functional representation mapping the input sequences to the assigned labels. To this aim, we explicitly defined different time thresholds for each machine to identify the unstable phase of the disruptive training discharges and assigned the disruptive label to plasma sequences that intersect the unstable phase of disruptive experimental runs, while the non-disruptive label is assigned to sequences extracted from the non-disruptive discharges. This classification scheme implicitly assumes that it is possible to detect a transition in time from a safe operational regime to a disruptive one and is another instance of incorporating physics knowledge into the AI workflow [S4-S5]. The chosen time thresholds vary on different considered devices which depend on the transition points where some of plasma parameters exhibit identifiable changes in behavior when disruptions occur, considering a notable fraction of disruptions [S6] and the suggestions from tokamak operators.

The training samples are ordered into sequences of ten time slices extracted from each shot of the training dataset. For each shot, we randomly select a subset of examples: this is a model’s hyperparameter, tuned for each machine. The disruptive training sequences are randomly extracted from all sequences that intersect the unstable phase of each disruptive shot, while those sequences outside of the unstable region are not included in the training set. If disruptive patterns are learned properly, the algorithm will be able to identify similar trends also at times prior to the formally set time threshold, enabling the detection of early disruptive precursors. The non-disruptive sequences are randomly extracted from the flattop of non-disruptive training discharges. It is interesting to notice that that the database population consists in most non-disruptive data, thus resulting in a dataset imbalanced with respect to disruptive data.

| Signal description | Symbol |
|--------------------|--------|
| Plasma current – programed plasma current | ip-error-fraction |
| Programed plasma current |  |
| Perburbed field of nonrotating mode a (n = 1 Fourier component), Bn=1/Btor | locked-mode-proxy |
| Electron density | Greenwald-density |
| Greenwald density |  |
| Distance between the plasma and the lower divertor | lower-gap |
| Current centroid vertical position error b | z-error-proxy |
| Plasma elongation | kappa |
| Normalized plasma pressure (ratio of thermal to poloidal magnetic pressure) | betap |
Radiated power

Standard deviation of the magnetic field measured from an array of Mirnov coils, normalized by $B_{tor}$

Loop voltage $V_{loop}$

Safety factor at the 95% flux surface

Normalized internal inductance

|                      | radiated-fraction | rotating-mode-proxy | v-loop | q95 |
|----------------------|-------------------|---------------------|--------|-----|
| Radiated power Input power |                   |                     |        |     |
| Standard deviation of the magnetic field measured from an array of Mirnov coils, normalized by $B_{tor}$ | |                      |        |     |
| Loop voltage $V_{loop}$ | |                     |        |     |
| Safety factor at the 95% flux surface | | |        |     |
| Normalized internal inductance | | | |     |

Table 2: The dataset composition of the three disruption warning databases

|                      | Number of training shots | Number of test shots | Number of validation shots | Sampling Rate (ms) | Time Threshold (ms) | Number of samples per training shots |
|----------------------|--------------------------|----------------------|---------------------------|--------------------|---------------------|--------------------------------------|
| C-Mod                | 3343 (692 disruptive)    | 651                  | 463                       | 5                  | 75                  | 15                                   |
| DIII-D               | 5286 (732 disruptive)    | 1085                 | 734                       | 10                 | 400                 | 25                                   |
| EAST                 | 8296 (2301 disruptive)   | 1674                 | 1137                      | 25                 | 500                 | 20                                   |

2. Training technicalities for HDL model

Training complex deep neural networks (NN) effectively is a challenging task that involves several technicalities, as detailed also in [S8] by Kates-Harbeck et al. Among other things, it is important to address the proper input feature normalization, or understand which are the tunable parameters that would increase the transfer ability of the cross machine predictor, while stabilizing its performances. In this section, we will describe the methods implemented to tackle these challenges for optimally training our deep NN predictor.

Normalization -- NNs usually need all input features to have similar numerical ranges for all training examples [S7-S8]. This makes the use of raw plasma signals as inputs to any NN numerically difficult as different signals have values that can range over many different orders of magnitude. Hence all 12 signals should be normalized before being used in the network. The normalization should ideally be a common transformation such that maps a set of signals with the same physical value from different devices to similar numerical values. Different tokamak devices have different operational spaces, spatiotemporal scales and diagnostics. Moreover, different machines have different event chain toward disruptions and the most important disruption-relevant physics parameters are different on each machine. Therefore, such physics-based common transformation is hard to find and its extrapolation to ITER is uncertain. However, we find that the best-performing method is to standardize each signal on one machine by its mean and standard deviation across the entire dataset. For each signal on one machine, its normalized form is obtained as follows: $x_{norm} = (x - \text{mean}(x))/\text{std}(x)$. The normalization parameter sets of all considered signals on each machine can be found in supplementary table 3.
Table 3: Normalization parameters of each signal on three machines

| Plasma signals       | Mean EAST | Std EAST | Mean DIII-D | Std DIII-D | Mean C-Mod | Std C-Mod |
|----------------------|-----------|----------|-------------|------------|------------|-----------|
| ip-error-fraction    | -0.0018   | 0.0206   | -0.0157     | 0.0545     | -0.0021    | 0.0425    |
| locked-mode-proxy    | 0.0018    | 0.0037   | 1.5462      | 3.5028     | 7.4951     | 4.3644    |
| Greenwald-fraction   | 0.4368    | 0.3270   | 0.4069      | 0.1840     | 0.2608     | 0.1315    |
| lower-gap            | 0.1598 m  | 0.0321 m | 0.1704 m    | 0.0823 m   | 0.0564 m   | 0.0163 m  |
| z-error-proxy        | 0.0072 m  | 0.0222 m | 9.1142 x 10^{-4} m | 0.0059 m | -8.4587 x 10^{-4} m | 0.0020 m |
| kappa                | 1.6298    | 0.1136   | 1.7675      | 0.1054     | 1.6176     | 0.0920    |
| betap                | 0.6890    | 0.3795   | 0.8261      | 0.5011     | 0.2387     | 0.1839    |
| radiated-fraction    | 0.1378    | 0.3470   | 0.5157      | 1.2370     | 0.3690     | 0.9520    |
| rotating-mode-proxy  | 0.0049    | 0.0112   | 6.8231 x 10^{-5} | 1.5536 x 10^{-4} | 0.6784 s^{-1} | 1.0623 s^{-1} |
| v-loop               | 0.4296 V  | 0.8612 V | 0.2931 V    | 0.9360 V   | -0.3294 V  | 1.7742 V  |
| q95                  | 6.0093    | 1.2748   | 4.8596      | 1.4172     | 4.4222     | 0.9427    |
| li                   | 1.1865    | 0.2280   | 1.0148      | 0.2148     | 1.4043     | 0.1720    |

The normalization process is independently done on all three machines which implies it is not machine-independent: a simple normalization scheme is instead chosen to solve the numerical challenge and leave the generalized signal transformation to the neural network. A machine-independent normalization method has also been tested for the three datasets: this normalization standardizes all datasets with a common set of parameters and gives slightly worse results. A performance comparison of HDL predictors using the two normalizations (machine-specific and machine-independent normalization) is shown in supplementary figure 1.

Figure 1: The ROC curves from test sets for machine specific normalization (blue) and machine independent normalization (red), for C-Mod (a), DIII-D (b), and EAST (c).

Cross-machine label smoothing – In our cross-machine numerical experiments, we combine training data from different machines to form a new training set. However, direct mixing of data from various device can result in a problem: the initial assigned target labels for other devices might not suitable for the new test device. For example, a certainly disruptive sequence from EAST might not be that disruptive to C-
Mod. Also, a non-disruptive sequence from C-Mod might be slightly unstable to EAST. In other words, you do not know what will happen if you run a C-Mod discharge on EAST or DIII-D and vice versa. To deal with this problem, we choose two smoothing parameters $\varepsilon_1$, $\varepsilon_2$ for each device ($\varepsilon_1$ for non-disruptive examples and $\varepsilon_2$ for disruptive examples) and use these two parameters to modify the target value of the training examples from other machines. When we train the HDL predictor with part of the data from other machines, instead of using their initial (0, 1) target values for non-disruptive examples and disruptive examples, we modify their target values as $(\varepsilon_1, 1-\varepsilon_2)$ (The new target values for non-disruptive examples are $\varepsilon_1$ and the target values for disruptive examples are $1-\varepsilon_2$. Notice that this modification is only applied to those training examples from other devices, those examples from the test device itself are not modified). We refer to this change in ground-truth target value as cross-machine label smoothing technique which can greatly improve the cross-machine ability of the HDL predictor.

Hyperparameter tuning and neural network ensemble -- The HDL disruption predictor has fourteen architectural and two labeling hyperparameters for each device. Guided by our previous numerical experiments on the C-Mod dataset, we scanned hyperparameters until finding a plateau where any hyperparameter set in this region gives high model performance. Within this region, changes in hyperparameters will only result in minor changes to the model’s performance for all three devices. Outside this region, performance on at least one device starts to drop drastically. The hyperparameters of the HDL predictor are therefore selected from the middle of this region and all following qualitative cross-machine conclusions consistently hold for all hyperparameter sets in this plateau. Additionally, our approach includes the adoption of an ensemble of twelve neural networks, each one identical in their HDL architecture and tunable hyperparameters but with different initialization seeds: The final prediction comes therefore from an ensemble average. This method is popularly known in the ML community and shown to significantly improve the accuracy and stability of the predictor [S9-S11]. A comprehensive list of tunable hyperparameters for our HDL model can be found in supplementary table 4.

| Hyperparameter   | Explanation                                                                 | Best Value  |
|------------------|-----------------------------------------------------------------------------|-------------|
| $\eta$           | Learning rate                                                               | $5 \times 10^{-4}$ |
| beta2            | The exponential decay rate for the second-moment estimates                   | 0.970       |
| $N_{\text{GRU}}$ | Number of Gated Recurrent Unit (GRU) Layers                                  | 2           |
| $n_{\text{cells-1}}$ | Number of GRU cells in layer 1                                               | 130         |
| $n_{\text{cells-2}}$ | Number of GRU cells in layer 2                                               | 90          |
| $n_{\text{batch}}$ | Batch size                                                                  | 300         |
| Target           | Type of target function                                                      | Negative Log-Likelihood (NLL) |
| $N_f$            | Number of convolutional filters in the 1-D causal convolution sublayers of each Multi-Scale Temporal Convolution (MSTConv) layer | 10          |
| Optimizer        | Stochastic optimization scheme                                              | Adam        |
| Dropout          | Dropout probability                                                         | 0.1         |
| L2 regularization | Weight regularization of all weight                                          | $1 \times 10^{-3}$ |
| $n_{\text{epoch}}$ | Number of training epochs                                                   | 32          |
| $d_{\text{latent}}$ | Dimension of final latent representation                                     | 12          |
| \(N_{\text{MSTConv}}\) | Number of MSTConv Layers | 3 |
|----------------------|--------------------------|---|
| \(\epsilon_1\text{-C-Mod}\) | Smoothing parameter \(\epsilon_1\) when test on C-Mod | 0.00 |
| \(\epsilon_2\text{-C-Mod}\) | Smoothing parameter \(\epsilon_2\) when test on C-Mod | 0.08 |
| \(\epsilon_1\text{-DIII-D}\) | Smoothing parameter \(\epsilon_1\) when test on DIII-D | 0.00 |
| \(\epsilon_2\text{-DIII-D}\) | Smoothing parameter \(\epsilon_2\) when test on DIII-D | 0.05 |
| \(\epsilon_1\text{-EAST}\) | Smoothing parameter \(\epsilon_1\) when test on EAST | 0.09 |
| \(\epsilon_2\text{-EAST}\) | Smoothing parameter \(\epsilon_2\) when test on EAST | 0.00 |

3. Cross-machine numerical experiments using C-Mod and DIII-D as target devices

**Cross-machine experiments testing on C-Mod** -- In these experiments, we consider EAST and DIII-D as ‘existing machines’ and investigate the effect of their data for the HDL disruption predictor when used on C-Mod, chosen as a ‘new device’. The numerical results are shown in supplementary figure 2-3 which consistently support our previous cross-machine conclusions, to be found in Section 3, cross-machine study.

Figure 2: ROC curves from the C-Mod test set using 20 C-Mod disruptive training shots (2.9%). In figure (a), we show a curve (magenta) without any disruptive C-Mod training data, while in (b) we show a curve without any C-Mod training data (green).

Figure 3: ROC curves from the C-Mod test set using 100% of C-Mod disruptive training shots. Figure (a) reports ROCs obtained including 100% non-disruptive C-Mod data, while (b) shows results when reducing to 25% the non-disruptive C-Mod training data.

**Cross-machine experiments testing on DIII-D** -- In these experiments, we consider EAST and C-Mod as ‘existing machines’ and investigate the effect of their data for the HDL disruption predictor when used on
DIII-D, chosen as a ‘new device’. The numerical results are shown in supplementary figure 4-5 which consistently support our previous cross-machine conclusions, to be found in Section 3, cross-machine study.

Figure 4: ROC curves from the DIII-D test set using 20 DIII-D disruptive training shots (2.7%). In figure (a), we show a curve (magenta) without any disruptive DIII-D training data, while in (b) we show a curve without any DIII-D training data (green).

Figure 5: ROC curves from the DIII-D test set using 100% of DIII-D disruptive training shots. Figure (a) reports ROCs obtained including 100% non-disruptive DIII-D data, while (b) shows results when reducing to 15% the non-disruptive DIII-D training data.

Supplementary Reference:
[S1] Gerhardt S. et al. 2013 Nucl. Fusion 53
[S2] Cannas B. et al. 2007 Nucl. Fusion 47 1559
[S3] Vega J. et al. 2013 Fusion Eng. Des. 88 1228
[S4] Rea C. et al. 2019 Nucl. Fusion 59 096016
[S5] Pau A. et al. 2019 Nucl. Fusion 59 106017
[S6] Montes K. J. et al. 2019 Nucl. Fusion 59 096015.
[S7] Goodfellow, I. 2016 Deep Learning, MIT Press, Cambridge, MA
[S8] Kates-Harbeck J. et al. 2019 Nature 568, 526–531
[S9] Haykin, S. 1999 Neural Networks: A Comprehensive Foundation. 2nd Edition, Prentice Hall, Englewood Cliffs, NJ
[S10] Hansen L. K. et al. 1990 IEEE Trans. Pattern Anal. Machine Intell 12 993-1000

[S11] Perrone M. et al. 1993 Neural Networks for Speech and Image Processing, R. J. Mammone, Ed. London, U.K