PAPER

# Hybrid deep-learning architecture for general disruption prediction across multiple tokamaks

IOP Publishing | Bureau international des poids et mesures Nuclear Fusion

Nucl. Fusion **61** (2021) 049501 (2pp) https://doi.org/10.1088/1741-4326/abe2e3

## Corrigendum

# Corrigendum: Hybrid deep learning architecture for general disruption prediction across tokamaks (2021 *Nucl. Fusion* 61 026007)

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

Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, MA, United States of America

E-mail: jxzhu@mit.edu

Received 29 January 2021
Accepted for publication 3 February 2021
Published 15 March 2021

Keywords: disruption, prediction, machine learning, stability, tokamak

(Some figures may appear in colour only in the online journal)

The case numbers in figure 7 should be 7, 8, 9, 10, 11, 12 instead of 1, 2, 3, 4, 5, 6 appearing in the original paper and the case numbers in figure 13 should be 7, 8, 9, 10, 11, 12 instead of 1, 8, 9, 10, 11, 12. The revised figures can be found below

In subsection 4.1.2 (cross-machine label smoothing), the referred table number should be 3 instead of 4 (i.e. see table 3 instead of table 4). In paragraph 1 of subsection 5.1, the referred table number should be 3 instead of 4 (i.e. can be found in table 3 instead of 4).

* Author to whom any correspondence should be addressed.

 

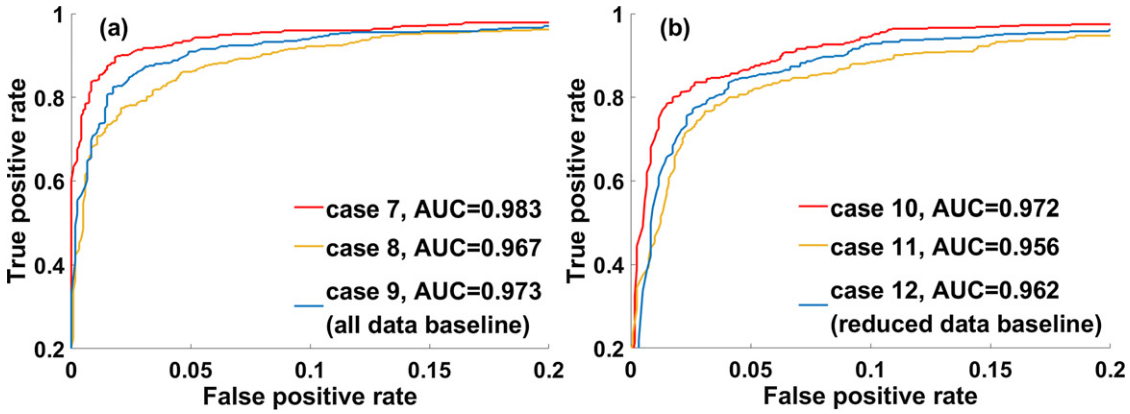


**Figure 7.** ROC curves from the EAST test set using all EAST disruptive training data.

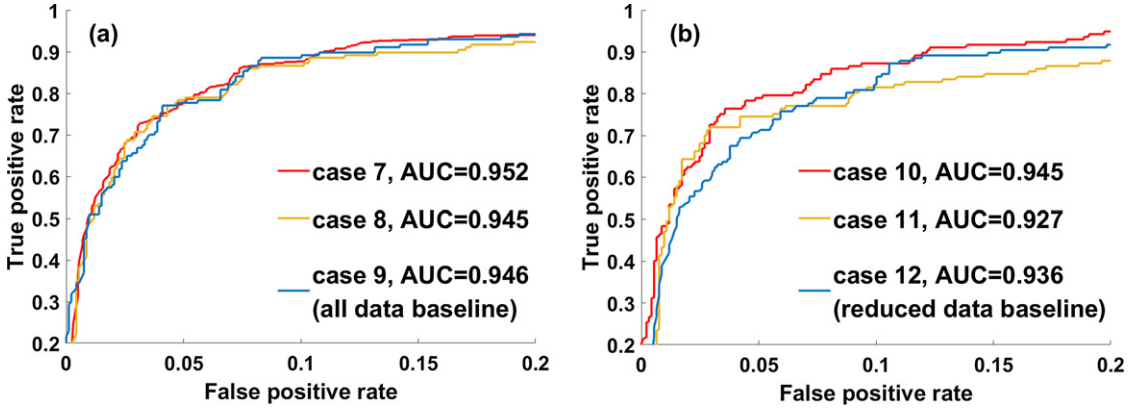


**Figure 13.** ROC curves from the DIII-D test set using all DIII-D disruptive-training data.

IOP Publishing | International Atomic Energy Agency Nuclear Fusion

Nucl. Fusion **61** (2021) 026007 (16pp) https://doi.org/10.1088/1741-4326/abc664

# Hybrid deep-learning architecture for general disruption prediction across multiple tokamaks

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

Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, MA United States of America

E-mail: jxzhu@mit.edu

Received 28 August 2020, revised 19 October 2020
Accepted for publication 30 October 2020
Published 22 December 2020

**Abstract**

In this paper, we present a new deep-learning disruption-prediction algorithm based on important findings from explorative data analysis which effectively allows knowledge transfer from existing devices to new ones, thereby predicting disruptions using very limited disruption data from the new devices. The explorative data analysis, conducted via unsupervised clustering techniques confirms that time-sequence data are much better separators of disruptive and non-disruptive behavior than the instantaneous plasma-state data, with further advantageous implications for a sequence-based predictor. Based on such important findings, we have designed a new algorithm for multi-machine disruption prediction that achieves high predictive accuracy for the C-Mod (AUC = 0.801), DIII-D (AUC = 0.947) and EAST (AUC = 0.973) tokamaks with limited hyperparameter tuning. Through numerical experiments, we show that a boosted accuracy (AUC = 0.959) is achieved for the EAST predictions by including only 20 disruptive discharges with thousands of non-disruptive discharges from EAST in the training, ***combined*** with more than a thousand discharges from DIII-D and C-Mod. The improvement in the predictive ability obtained by combining disruption data from other devices is found to be true for all permutations of the three devices. Furthermore, by comparing the predictive performance of each individual numerical experiment, we find that non-disruption data are machine-specific, while disruption data from multiple devices contain device-independent knowledge that can be used to inform predictions for disruptions occurring in a new device.

Keywords: disruption, prediction, machine learning, stability, tokamak

(Some figures may appear in colour only in the online journal)

## 1. Introduction

The use of 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 fusion device and prevent the realization of a functioning plasma reactor. Forecasting plasma instabilities and disruptions using first-principle models has been demonstrated to be extremely difficult, due to the complexity of the problem and the high non-linearity of the system [1]. To date, disruption prediction has been studied through two main approaches: data-driven versus physics-driven (or model-based). On the one hand, recent statistical and machine

* Author to whom any correspondence should be addressed.

 

learning (ML) approaches based on experimental data have shown attractive results for disruption prediction, even in real-time environments [2–10]. Different tokamak devices have different operational spaces, spatiotemporal scales for physics events, and plasma diagnostics. Therefore, most of these data-driven approaches were developed and optimized specifically for one device and did not show promising cross-device predictive ability [2–6, 8, 10]. Specifically, cross-machine studies such as [7] focused on predictors that were trained on datasets purely or mostly from one device: these predictors achieved excellent performance 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. In addition, the complexity of these data-driven models limits their physics interpretability. Recently, a published work by Kates-Harbeck *et al* [9] demonstrated for the first time the potential of predictors based on deep learning (DL) for acquiring a general representation of experimental data that can be used in cross-machine applications. On the other hand, model-based disruption prediction studies [11–15] seek to identify event chains that can lead to disruptions through early event detection, which can help operators to avoid disruptions. As an example, the disruption event characterization and forecasting (DECAF) [12] suite can detect a variety of events by taking advantage of first-principle, physics-based modules for tearing stability, resistive-wall modes, etc. These models are tailored for stability-limit detection on different devices and are sometimes accelerated through the adoption of surrogate ML models [16]; this latter approach tries to take advantage of both paradigms by incorporating physics and ML aspects in the same model to achieve better prediction as well as improved interpretability.

In this paper, we focus on the data-driven approach by exploiting existing experimental databases for disruption prediction available for the Alcator C-Mod, DIII-D, and EAST tokamaks. We first introduce an application of a dimensionality-reduction algorithm to the high-dimensional plasma data; we then discuss a new DL framework for disruption prediction based on important findings from the explorative data analysis, which allows effective knowledge transfer from existing devices to new ones using very limited disruption data from the new devices, while retaining high accuracy for the individual datasets. To support 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 these three very different tokamaks. In addition, we combined data from the three different devices to add some randomization to the training domain, which can alleviate the over-learning of one specific device's behavior. We also designed a set of cross-machine experiments to find general guidelines for making disruption predictions about new devices using very limited disruption data from those devices. Moreover, unlike the machines used for previous studies [7, 9, 10], here, the three machines have very different features: EAST is a medium-sized ($R = 1.85$ m, $a = 0.45$ m) superconducting tokamak with a hybrid first wall: its lower divertor is carbon, the middle wall is molybdenum (Mo), while the upper divertor is made of tungsten [17]. DIII-D is a medium-sized ($R = 1.67$ m, $a = 0.67$ m) tokamak with a carbon wall and a relatively large error field: most of disruptive shots in our DIII-D database contain a locked mode as the last precursor in their event chain toward disruption [18, 19]. C-Mod is a smaller tokamak ($R = 0.68$ m, $a = 0.22$ m) with high energy density (a plasma pressure of up to 2.05 atm), a high magnetic field ($B_T$ up to 8 T) and a high-Z metal (Mo) wall. The combination of these different characteristics covers a substantial fraction of ITER's features [20, 21], although no existing device can, by itself, fully represent ITER at scale. A cross-machine study using data from these existing devices is nevertheless well-suited for an investigation of disruption prediction solutions for ITER.

**Table 1.** Descriptions and symbols of all the signals considered.

| Signal description | Symbol |
|---|---|
| $\frac{\text{Plasma current}-\text{programed plasma current}}{\text{Programed plasma current}}$ | `ip-error-fraction` |
| **Perturbed field of nonrotating mode**[a] **(n = 1 Fourier component), $B^{n=1}/B_{tor}$** | `locked-mode-proxy` |
| $\frac{\text{Electron density}}{\text{Greenwald density}}$ | `Greenwald-fraction` |
| **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` |
| $\frac{\text{Radiated power}}{\text{Input power}}$ | `radiated-fraction` |
| **Standard deviation of the magnetic field**[c] **measured from an array of Mirnov coils, normalized by $B_{tor}$** | `rotating-mode-proxy` |
| **Loop voltage $V_{loop}$** | `v-loop` |
| **Safety factor at the 95% flux surface** | `q95` |
| **Normalized internal inductance** | `li` |

[a]For the C-Mod database, the `locked-mode-proxy` signal is obtained from a Mirnov coil array instead of the saddle coil.
[b]For the DIII-D database, we use the current centroid vertical position instead of positional error for the `z-error-proxy` signal.
[c]For the DIII-D database, we use $n = 1$ component of magnetic field measured by a Mirnov coil array normalized by $B_{tor}$ for the `rotating-mode-proxy` signal.

## 2. Dataset description

Our disruption prediction studies are conducted on disruption warning datasets for three machines [8]: Alcator C-Mod (2012–2016 campaigns), DIII-D (2014–2018 campaigns) and EAST (2014–2018 campaigns). For all three databases, we include all types of disruption, except for intentional ones. The choice of which parameters to include in the databases is guided by our knowledge of the plasma physics mechanisms inherent to the disruption characteristics of the different devices, as well as the accessibility and consistency of these

**Table 2.** The dataset composition of the three disruption warning databases[a].

| | No. training shots | No. test shots | No. validation shots | Sampling rate (ms) | Time threshold (ms) | No. samples/ training shots |
|---|---|---|---|---|---|---|
| C-Mod | 3343 (692) | 651 (142) | 463 (98) | 5 | 75 | 16 |
| DIII-D | 5286 (732) | 1085 (157) | 734 (107) | 10 | 400 | 25 |
| EAST | 8296 (2301) | 1674 (475) | 1137 (322) | 25 | 500 | 20 |

[a]Values in parentheses give the exact number of disruptive shots.

parameters on all three machines. Many of the disruption-relevant parameters included in this study are also influenced by several papers [22–24]. The signals considered for the predictive models reported in this paper and their definitions can be found in table 1, while the composition of the three training datasets is shown in table 2. Given these databases, we formalize the disruption prediction problem in a *sequence-to-label* supervised ML framework, where we assign a label to each input plasma sequence, $S$ (a 10-step consecutive temporal sequence of 12 plasma signals) and train an algorithm to learn the functional representation, mapping the input sequences to one of two possible labels, 'disruptive' or 'non-disruptive'. To further 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 intersected 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 artificial intelligence (AI) workflow [25, 26]. The chosen time thresholds vary for the different devices considered and depend on the transition points where some of plasma parameters exhibit identifiable changes in behavior when disruptions occur, considering a significant fraction of the disruptions [8] and suggestions from the 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 the 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 the disruptive patterns are learned properly, the algorithm will be also able to identify similar trends 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 note that the database population consists of mostly non-disruption data, thus resulting in an imbalanced dataset with respect to disruption data.

## 3. Explorative data analysis through a data visualization technique

As mentioned in the introduction, disruptions are highly complicated phenomena, characterized by diverse physics events with different spatiotemporal scales and non-linear dynamics [1]. In particular, we usually have to deal with high-dimensional data from multiple plasma signals, which complicates both analysis and physics interpretation when studying disruptive events. In this section, we discuss the application of a nonlinear dimensionality-reduction technique called t-distributed stochastic neighbor embedding (t-SNE) [27] (see appendix A for further details of the t-SNE algorithm) to visualize high-dimensional plasma data in a 2D plane to study the inherent data structure of the plasma signals considered. In principle, the t-SNE algorithm can be applied to any high-dimensional database. However, in this section, we only show its application to the C-Mod database as this database is considered more difficult to predict via a data-driven approach than those of EAST and DIII-D [6]. Analysis of the DIII-D and EAST disruption databases can be found in appendix A.

Figure 1 shows the t-SNE algorithm applied to time-slice data (left) and aggregated-sequence data (right) for the C-Mod disruption warning database. In the left subplot, each blue point represents a randomly sampled time slice (a 12-element array composed of the 12 plasma signals from table 1) from the flattop of a non-disruptive shot, while each red point represents a time slice randomly sampled from the last 75 ms of a disruptive shot. On the right, each red point represents a 10-step (a $10 \times 12$ element matrix) sequence randomly sampled from the last 75 ms of a disruptive shot, while each blue point represents a 10-step sequence randomly sampled from the flattop of a non-disruptive shot. We include all disruptions without discriminating by cause. The coloring of each datapoint in the plots is done *a posteriori*, i.e. not provided during the training process, thereby characterizing the t-SNE as an unsupervised clustering technique.

Several important conclusions can be drawn: first, the clustering of individual time slices cannot isolate clear data clusters in the low-dimensional map. However, by performing t-SNE on 10-step plasma sequences, it is possible to isolate three major clusters—identified by dashed circles in figure 1—which account for approximately 60% of the dis-

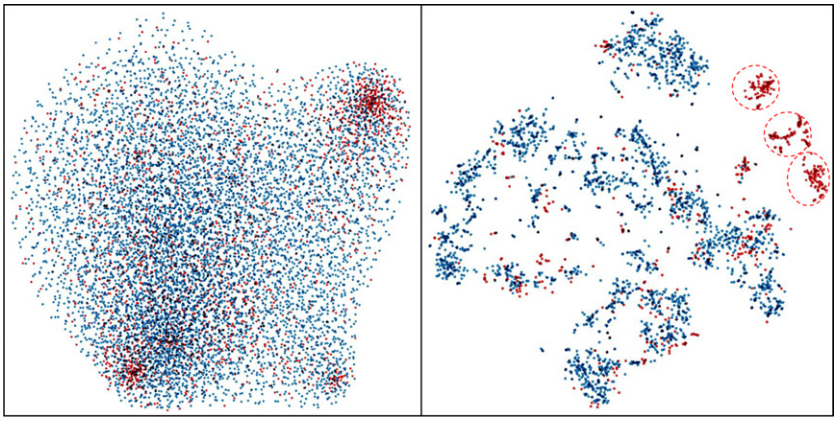

**Figure 1.** t-SNE clustering for the visualization of C-Mod data. On the left, t-SNE is performed on individual disruptive (red) and non-disruptive (blue) time slices, while on the right t-SNE is performed on 10-step disruptive (red) and non-disruptive (blue) sequences. Three major clusters of disruption data can be isolated. The coloring is done *a posteriori*.

ruption data. The improved separation of disruptive and non-disruption data obtained from clustered sequences highlights the importance of temporal correlation and mutual information between consecutive time slices. This further suggests that sequence-based classifiers could have a clear advantage over the time-slice-based ones. Secondly, the application of t-SNE to the C-Mod sequences reveals that a substantial fraction ($\sim$40%) of the disruptive sequences remains mixed with non-disruptive sequences. Further analysis of such data finds that these disruptive sequences are primarily linked to fast radiative collapses caused by molybdenum impurities. These disruptions have fast timescales, up to a few tens of milliseconds, and we argue that any data-driven disruption prediction algorithm for C-Mod would struggle to predict such cases (at least with the current set of input features) and thus be affected by a high proportion of false negatives (missed predictions). The three isolated clusters, identified by red dashed circles in figure 1, are representative of specific disruption dynamics such as VDEs, impurity accumulations and MHD-driven disruptions. These precursors can be identified through inspection of the specific time series.

## 4. The hybrid deep-learning (HDL) disruption-prediction framework

Based on our findings about the importance of temporal information, we introduced a hybrid deep learning (HDL) network for time-series processing. Figure 2(*a*) shows the architecture of the network that was used for the cross-machine disruption-prediction application reported in this paper. The HDL network consists of two gated recurrent unit (GRU) layers [28], 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 by work in machine translation [29], and the detailed structure of one MSTConv layer is shown in figure 1(*b*). It consists of six 1D causal-convolution layers [30] with different window lengths, $L$, from one to six. The first 1D convolution layer can only access the current time step, $t_0$. The $L$th 1D convolution layer can look at $L$ time steps from $t_{0-L+1}$ to $t_0$. This structure enables different 1D convolution layers to capture local temporal information at different levels (e.g., the first-order time derivative, second-order time derivative, ...). The resulting outputs from these six layers are concatenated and then processed through a batch normalization layer [31] and a rectified linear unit (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 long-range dependencies across different signals in the sequences. The subsequent MSTConv and fully connected layers can then compress and summarize the output representation from the GRU layers to a 12-dimension latent encoding (the dimensionality of the latent encoding is a tunable parameter) which can be mapped to the output by the classification layer.

A shot-by-shot testing scheme was designed following [8] to simulate alarms triggered in the plasma-control system using the test shots from different devices. Given an input plasma sequence $S$ which is a 2D matrix consisting of ten time steps and twelve input features, i.e. a $10 \times 12$ matrix, the predictor maps $S$ to a disruption level between 0 and 1 at the last time step of the sequence; here, 1 is the disruptive class and 0 is the non-disruptive class. During testing, the whole flattop phase of each test shot is subdivided into batches of 10 step sequences, given the HDL architecture design. Each neighboring testing sequence will have 9 steps of overlap, and there are $N$-9 sequences for a test shot with $N$ steps. If the disruption level exceeds a pre-set threshold—e.g., 0.5—at any test time step, the test shot is classified 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 ($t_{\mathrm{dis}}$). A disruption successfully detected at C-Mod is shown in figure 3: under a binary classification scheme, this is regarded as a true positive, while false positives correspond to a false warning, or a healthy plasma being declared to be disruptive. This latter situation can still lead to some machine damage, but on the other hand, being unable to predict a disruption early

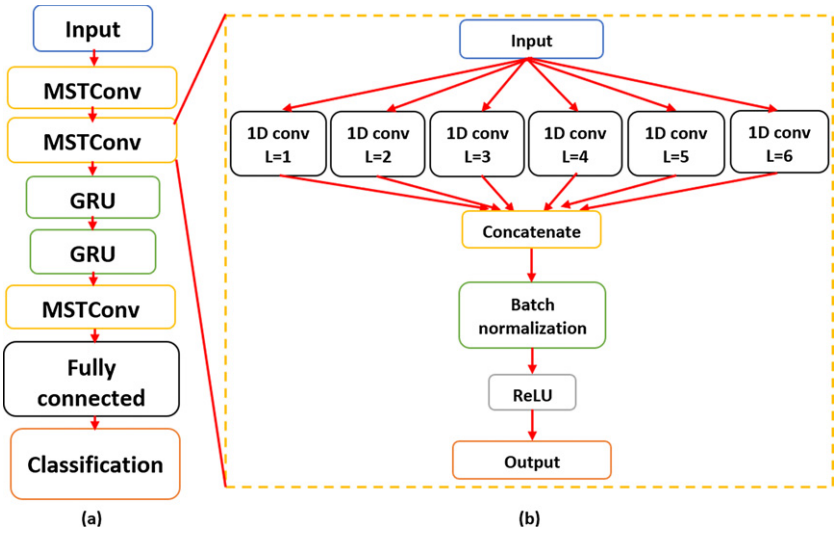


**Figure 2.** The HDL architecture (*a*) and the detailed structure of the MSTConv layer (*b*). Notice that the MSTConv layer consists of six 1D causal convolution layers with window lengths, $L$, from 1 to 6.

enough (a 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 for the disruption, as visually demonstrated by a receiver-operator characteristic (ROC) curve [32]. The area under the ROC curve (AUC) is used as a performance metric for the HDL predictor. Throughout this paper, we evaluate the predictive performances for all tokamaks 50 ms before the disruption event: this is chosen as the minimum warning time required to successfully trigger disruption-mitigation systems for future devices [33].

### *4.1. Training technicalities for the HDL model*

Training complex deep neural networks (NNs) effectively is a challenging task that involves several technicalities, as also detailed in [9]. Among other things, it is important to address the appropriate input feature normalization, or to understand which tunable parameters can increase the transferability of the cross-machine predictor, while stabilizing its performance. In the following subsections, we will describe the methods implemented to tackle these challenges to optimally train our deep NN predictor.

*4.1.1. Normalization.* NNs usually need all input features to have similar numerical ranges for all training examples [9, 34]. This requirement makes the use of raw plasma signals as inputs to any NN numerically difficult, as different signals have values that can range over many orders of magnitude. Hence, all twelve signals should be normalized before being used in the network. The normalization should ideally be a common transformation, such that it maps a set of signals with the same physical values from different devices to similar numerical values. Different tokamak devices have different operational spaces, spatiotemporal scales, and diagnostics. Moreover, different machines have different event chains preceding disruptions, and the most important disruption-relevant physics parameters are different for each machine. Therefore, such a physics-based common transformation is difficult 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: $x_{\mathrm{norm}} = (x - \mathrm{mean}(x))/\mathrm{std}(x)$. The normalization parameters for all the signals considered on each machine, as well as the common normalization parameters, can be found in appendix A.

The normalization process is performed independently for all three machines, which implies it is *not* machine-independent: this simple normalization scheme is instead chosen to solve the numerical challenge and leave the generalized signal transformation to the NN. A machine-independent normalization method has also been tested for the three datasets: this normalization standardizes all datasets with a common set of parameters. A performance comparison of the HDL predictors using the two normalizations (machine-specific and machine-independent) is shown in figure 4. For the machine-specific cases (blue curves), the HDL predictor is trained and evaluated using the training and test sets of each machine normalized by the corresponding normalization parameters. For the machine-independent cases (red curves), the HDL predictor is trained and evaluated using the training and test sets for each machine but normalized using the '*common*' normalization parameters (i.e. fixed for all three machines), and these give only slightly worse results, hinting that the HDL performance is only weakly dependent on the normalization parameters, as long as all signals have appropriate numerical ranges (approximately $-1$ to 1) (table 3).

*4.1.2. Cross-machine label smoothing (CMLS).* To train a multi-machine predictor, we combine the training data from different machines to form a new training set. However, the direct mixture of data from various devices can result in a problem: the initially assigned target labels for other devices

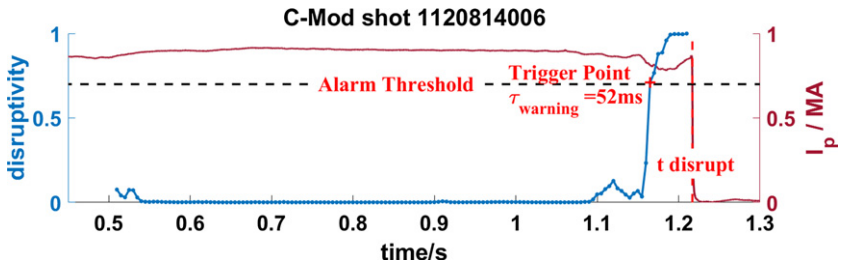


**Figure 3.** A successfully detected disruption on C-Mod.

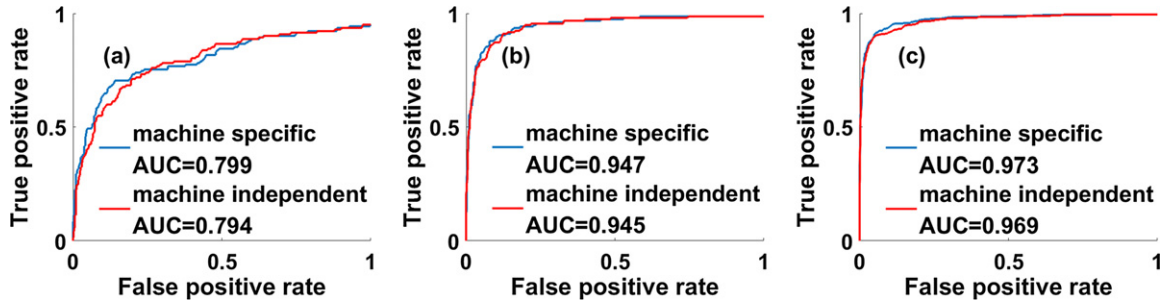


**Figure 4.** The ROC curves from the test sets for machine-specific normalization (blue) and machine-independent normalization (red), for C-Mod (*a*), DIII-D (*b*), and EAST (*c*).

**Table 3.** Cross-machine prediction results of HDL.

| Training set<br>Test set | C-Mod + DIII-D<br>EAST | C-Mod + DIII-D + few EAST data<br>EAST | EAST + C-Mod<br>DIII-D | EAST + C-Mod + few DIII-D data<br>DIII-D | EAST + DIII-D<br>C-Mod | EAST + DIII-D + few C-Mod data<br>C-Mod |
|---|---|---|---|---|---|---|
| HDL ensemble | 0.788 | 0.819 | 0.622 | 0.741 | 0.564 | 0.605 |
| HDL ensemble + CMLS | 0.806 | 0.837 | 0.659 | 0.765 | 0.588 | 0.631 |

might not be suitable for the new test device. For example, a certain disruptive sequence from EAST might not be that disruptive to C-Mod, when compared to C-Mod's disruption data. Also, a non-disruptive sequence from C-Mod might be slightly unstable for EAST, when compared to EAST's non-disruption data. In other words, we need to take into account the uncertainty associated with running a C-Mod discharge at EAST or at DIII-D and vice versa. To deal with this problem, we choose two smoothing parameters $\varepsilon_1$ and $\varepsilon_2$ for each device ($\varepsilon_1$ for non-disruptive examples and $\varepsilon_2$ for disruptive examples) and use these two parameters to modify the target values 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 to ($\varepsilon_1$, $1-\varepsilon_2$). The new target values for the non-disruptive examples are $\varepsilon_1$, and the target values for the disruptive examples are $1-\varepsilon_2$. Notice that this modification is only applied to training examples from other devices; examples from the test device itself are not modified. We refer to this change in ground-truth target value as the cross-machine label smoothing (CMLS) technique, which we find further improves the cross-machine ability of the HDL predictor (see table 4).

*4.1.3. Hyperparameter tuning and NN 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 roughly scanned the hyperparameter space using a random search for all three machines' data until a plateau was found whereby any hyperparameter set in this region produces high performance for all three devices. Within this region, changes in hyperparameter choice will only result in minor changes to the model's performance for all three devices. Outside this region, performance drops drastically for at least one device. The hyperparameters of the HDL predictor are therefore selected from the middle of this plateau, and all following qualitative cross-machine conclusions consistently hold for all hyperparameter sets that exist in this region. Additionally, our approach includes the adoption of an ensemble of twelve NNs, all identical in their HDL architecture and tunable hyperparameters but with different initialization seeds.

**Table 4.** Training-set composition of all cross-machine experiments using EAST as the '*new machine*'[a].

| | Existing machines (C-Mod + DIII-D) | | New machine (EAST) | |
|---|---|---|---|---|
| Case no. | Non-disruptive | Disruptive | Non-disruptive | Disruptive |
| 1 | None | All (692 + 732) | All (5995) | 20 |
| 2 | None | All | All | None |
| 3 | None | All | 50% (2998) | 20 |
| 4 | None | None | All | 20 |
| 5 | All (2651 + 4554) | All | All | 20 |
| 6 | All | All | None | None |
| 7 | None | All | All | All (2301) |
| 8 | All | All | All | All |
| 9 | None | None | All | All |
| 10 | None | All | ∼33% (1998) | All |
| 11 | ∼20% (692 + 732) | None | ∼33% | All |
| 12 | None | None | ∼33% | All |

[a]Values in parentheses give the exact number of shots.

The final prediction therefore comes from an ensemble average. This method is popularly known in the ML community and has been shown to significantly improve the accuracy and stability of the predictor [35–37]. A comprehensive list of tunable hyperparameters for our HDL model can be found in appendix A.

### *4.2. HDL performances for the three devices and a benchmark using a random forest*

The HDL predictor successfully achieves state-of-the-art performance on all three test sets when compared to other fully-optimized deep NN disruption predictors [9]. To verify this, we trained three HDL predictors (with the fixed hyperparameters given in section 4.1.3) and three random forest (RF) predictors [6, 8, 19] using the training set of each machine and evaluated their performances on the test set corresponding to that machine. The results are shown in figure 5. To carry out a fair comparison with the previous approaches, the RF predictors for each machine are specifically optimized using the corresponding validation set: we carried out a $K$-fold cross-validation procedure together with a parallelized grid search to find the optimal set of time threshold and forest hyperparameters for each machine using a binary-classification-performance metric called the $F_\gamma$-score [6, 8]. The HDL predictor exceeds the RF performance for all three datasets: it triggers fewer false alarms for good discharges while at the same time missing fewer real disruptions, which shows the strong applicability and generalization power of this model. This general improvement on multiple machines seems to mainly arise from the advantage of the sequence-based model designed for time-series processing, as suggested in section 3. Besides its impressive performance, the inference time of our model is short, allowing it to make a prediction in roughly 1 ms using an eight-core CPU. This fast and novel model is not only an important step toward the prediction requirement of future devices, but also suggests a powerful conclusion. Given that a common set of model hyperparameters used for three predictors can achieve high performance on all three machines, it suggests that although different devices may have disjoint operational regimes, a common type of discriminant function appears to exist—with the same model hyperparameters—capable of separating the disruptive from the non-disruptive phases for all these machines.

## 5. HDL cross-machine study on Alcator C-Mod, DIII-D, and EAST data

The availability of a huge amount of experimental data across several tokamaks allows us to design numerical experiments to test the transfer learning capabilities of the HDL architecture. Future reactors such as ITER cannot tolerate more than a few unmitigated disruptions [1], so we must be able to predict their disruptions given very limited disruption data from them. Expanding from previous cross-machine disruption prediction study described in [7, 9, 10], we have designed complete numerical experiments to test the transfer-learning capabilities of the HDL architecture. Given the availability of a large database of aggregated data from very different tokamaks, it is important to verify *how useful* the data from existing devices is for predicting unstable plasmas in a new device, *if* it can be done. In this section, we nominate two machines to play the parts of '*existing machines*' and investigate the effect of their data on the HDL disruption predictor when used for the third machine in the role of '*new device*'. We primarily focus on the EAST case (EAST is chosen to be the 'new device') in the following section. However, **all the following qualitative conclusions are machine-independent**: they always hold, no matter which device is selected as the 'new device'. The results for the other two cases can be found in appendix A.

### *5.1. Cross-machine prediction performance using the HDL architecture*

As a first step, we would like to test the cross-machine performance of the HDL model. To do this, we train the HDL network using data from two '*existing devices*' and test its performance on the third unseen 'new device'. Following the predictor approaches known as 'with a glimpse' or 'from scratch' [9, 10], we then add ten disruptive and ten

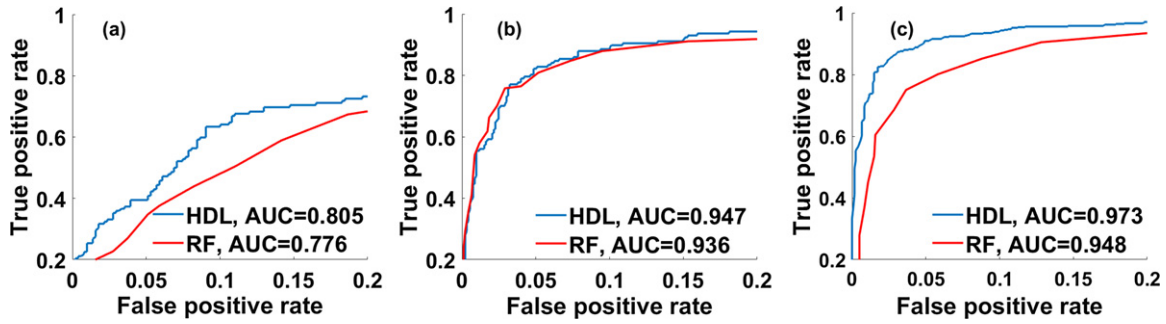


**Figure 5.** The ROC curves from the test sets for the HDL model and the RF model, for C-Mod (*a*), DIII-D (*b*) and EAST (*c*). We only show the upper-left region of the curves where the predictors have highest performance.

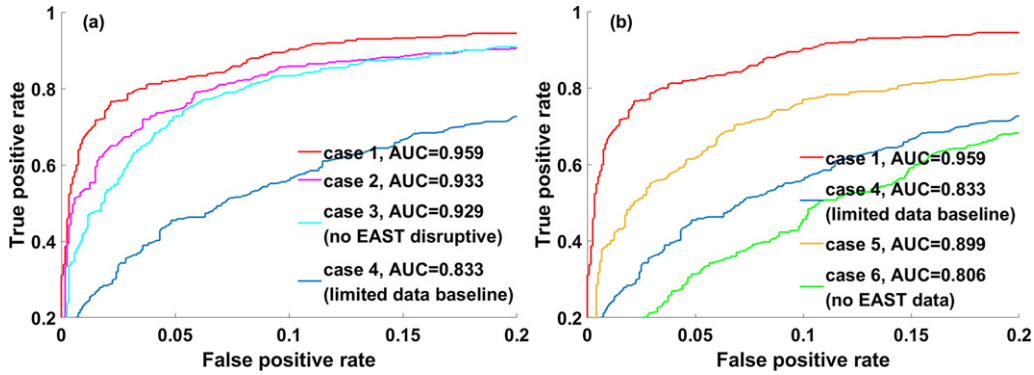


**Figure 6.** ROC curves from the EAST test using limited EAST disruption training data.

non-disruptive discharges from the 'new device' to the training sets and do indeed observe a boost in the test-set performances when using limited data from the target device. In the context of previous cross-machine studies [9, 10], our HDL framework shows promising transferability for these three different devices, and these test results can be found in table 4 (all values reported here are AUCs averaged over the network ensemble).

Apart from the performances, we are interested in investigating how data from different existing devices influence the prediction of disruptions to a new device, and in particular if any effect can be linked to general, device-independent knowledge. To support this aim, we design two further sets of cross-machine numerical experiments. The training-set composition for each experiment can be found in table 4.

### *5.2. Cross-machine experiments using limited disruption data from the 'new device'*

The first set of cross-machine experiments was conducted using limited disruptive training shots from the new device. The results of these numerical experiments are shown in figures 6(*a*) and (*b*). In the first experiment, the disruption predictor is trained on 20 randomly selected disruptive training shots and all non-disruptive training shots from the target new device, plus disruptive shots from two other devices (existing machines). This combination achieves the best performances on the new device test dataset (AUC = 0.959, for the EAST case). In the second and third experiment, we first remove all the new device disruptive shots and then, separately, 50% of the new device non-disruptive shots from the first training dataset. In the fourth experiment, the predictor was trained only using selected new-device training data (20 disruptive training shots, all non-disruptive training shots), this being our limited-data baseline model. In the fifth experiment, we add non-disruptive shots from two other machines to the first training dataset. In the sixth experiment, the predictor is only trained on data from other machines (with no new device data) and the low performance of this predictor highlights the importance of including non-disruption data from the target machine. From these numerical experiments, it is possible to draw the following conclusions:

- HDL achieves relatively good performance on a new device if using a few disruptive shots and many non-disruptive shots from the new device, plus a large amount of disruption data from existing devices. All the components mentioned above are necessary because removing any of them will reduce the performance (cases 1 to 4 in figure 6(*a*)).
- Non-disruption data from existing devices is harmful to HDL performance but disruption data from existing devices improves the predictive power (cases 1, 4, 5 in figure 6(*b*)).
- Non-disruption data from the target device can substantially improve the predictive power (case 6 in figure 6(*b*)).

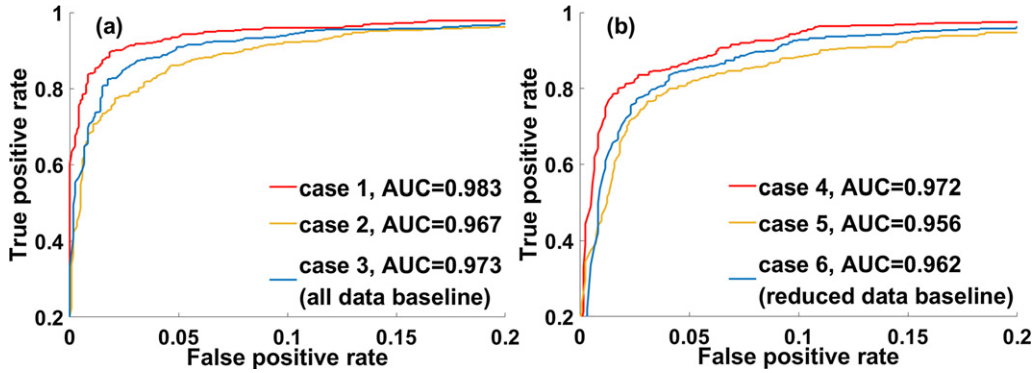


**Figure 7.** ROC curves from the EAST test set using all EAST disruption training data.

### *5.3. Cross-machine experiments using all the disruption data from the 'new device'*

To further investigate the effect of the class imbalance in the training set, we conducted another set of experiments using all the disruptive training shots of the new device. The results are reported in figure 7; again, in the seventh experiment, the disruption predictor is trained on all the disruptive and non-disruptive training shots from the new device, including the disruptive shots from the other two machines, and this predictor achieves the best performance on the new device test dataset (AUC = 0.983, for the EAST case). In the eighth experiment, we add the non-disruptive shots from two other machines to the first training dataset. In the ninth experiment, the predictor is only trained on all the new device training data, which is the 'all data' baseline case for comparison. In experiments 10–12 (figure 7(*b*)), we randomly remove most of the new-device non-disruptive training shots, thus reducing the new-device non-disruptive training data to be fewer than the new device disruptive training data, i.e. an inversely imbalanced situation. The test results from figures 7(*a*) and (*b*) point to the following further conclusions:

- Adding disruption data from existing machines can still slightly improve the test performances for the new device, even if abundant new machine data exists (cases 7, 9 in figure 7(*a*)). However, adding non-disruption data from existing machines is still harmful in this situation (cases 7, 8 in figure 7(*a*)).
- The effects of disruption data (positive) and non-disruption data (negative) do not result from the class imbalance of the new machine dataset, because the disruption data from existing devices consistently have positive effects, while the non-disruption data still have negative effects in the inversely imbalanced situation (figure 7(*b*)). This difference between disruptive and non-disruption data is machine-independent, i.e. it is a universal conclusion.
- Also, removing non-disruption data from the target device always decreases the test performance, no matter how imbalanced the target dataset is (cases 1, 3 in figure 6(*a*), case 9, 12 in figure 7(*b*)).

### *5.4. Summary of cross-machine numerical conclusions*

Considering all the conclusions in sections 5.2 and 5.3, it is possible to state that the knowledge obtained from the disruption data of existing devices improves the performance with a new device, while the non-disruption data seem to have negative effects, which do not result from the label imbalance of the training datasets. This suggests that the non-disruption data are specific to one device, but that the disruption data contain some general knowledge about disruption dynamics which is transferable to a new device when using predictive, data-driven models. Indeed, different machines usually have different operational spaces, spatiotemporal scales for physics events, and plasma diagnostics [6, 8, 9]. In other words, the distributions of plasma signals can vary significantly from one machine to another. From a data-driven perspective, this further implies that finding a numerical transformation that maps a set of signals from one device to any other device can be very challenging without incorporating machine-specific information, and this might indeed pose a great challenge when comparing ITER's operational space to all the existing devices. Due to these considerations, we are inclined to conclude that the non-disruption data of existing devices are machine-specific and will only decrease the accuracy of predictive models regarding a new device when they are directly mixed with data from the target device. Nevertheless, different devices show similar behaviors when operating close to a disruption. For example, `li` [6], `v-loop` and `lock mode` [6, 8] signals have consistently been observed to increase in multiple machines when disruptions are imminent. These universal trends can be effectively captured by our time-sequence-based model as general knowledge about disruptions hidden beneath the disruption data, which then helps with disruption prediction for new devices.

## 6. Summary and future plans

In this paper, we have discussed findings from an explorative data-analysis study of the C-Mod disruption database using a dimensionality-reduction technique called t-SNE and demonstrated that time-sequence data can better separate disruptive from non-disruptive behaviors, compared to the instantaneous plasma-state data (i.e. individual time slices). Based on this

conclusion, we have designed a new, powerful disruption-prediction algorithm based on DL and also demonstrated a general, effective way to transfer knowledge from existing devices to new devices, which offers guidelines for disruption predictions for new devices using limited disruption data from the target new devices. The cross-machine study on Alcator C-Mod, DIII-D, and EAST shows that, given the highly elaborated DL architecture designed, it is not enough to only use data from existing devices to predict disruptions in a new tokamak device. The numerical experiments discussed in section 5 demonstrate that in addition to data from existing devices, the model's performance greatly improves if both non-disruption and some disruption data are included from the target device. In particular, the HDL predictors can reach an AUC $> 0.95$ for EAST if the trained includes only a small set (20) of disruptive discharges from the target device (EAST), *but* using all the available non-disruption information from the target machine.

Furthermore, disruption and non-disruption data are found to have different impacts on the cross-machine disruption -prediction framework presented in this paper, with the implication that the non-disruption data are machine-specific and the disruption data contain general knowledge about disruptions. These results are an important milestone for disruption-prediction research for the next generation of 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 fine hyperparameter tuning, which might further increase the performance of the predictor and find new insights in cross-machine studies. Secondly, in the future version of the 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 characteristics of the non-disruption data suggest that it could be beneficial to mix device-specific representations with plasma signal representations to increase the model's expressive power. This may enable us to extract information from machine-specific non-disruption data and improve the predictions about new devices. In conclusion, the continuation of this project may indeed contribute to the development of a reliable trigger for ITER's disruption-mitigation system, by further refining the guidelines for a robust disruption-prediction scheme for yet-to-be-built tokamaks.

## Disclaimer



## Acknowledgments

This material is based upon work supported by the US 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. Additionally, this work is supported by the National MCF Energy R & D Program of China, Grant No. 2018YFE0302100. The HDL architecture reported in the paper was developed using the TensorFlow library [38]. Part of the data analysis reported in this paper was performed using the 'one modeling framework for integrated tasks' (OMFIT) integrated modeling framework [39]. The authors are grateful to N. Logan, E. Marmar and R. Buttery for their support and valuable discussions.

## Appendix A

### *A.1. Details about the sampling rates and time thresholds of the three disruption-warning databases*

Our DIII-D and EAST databases were initially created to have non-uniform sampling rates [40]. For DIII-D, all shots (with an average flattop duration of about 3 s) were sampled every 25 ms, and additional sampling was done every 2 ms for the 100 ms period before each disruption. For EAST, all shots (average flattop duration $\sim$6 s) were sampled every 100 ms (some discharges were 100 s long), and additional sampling was done every 10 ms for the 250 ms period before each disruption. The HDL required that all input signals were aligned to a uniform sampling rate: this avoided distortions in the correlations learned from the data due to the increased sampling frequency around the disruption time. Guided by our previous experience about the timescales of disruption dynamics at DIII-D and EAST, we decided to interpolate the DIII-D and EAST databases to have a uniform time base with intermediate sampling rates (10 ms for DIII-D, 25 ms for EAST), chosen as a trade-off between the additional high sampling rate before disruption and the low sampling rate for the remaining flattop. Our clustering study combined with further preliminary analysis showed that C-Mod had considerably faster disruption dynamics than the two other machines. Therefore, we repopulated the database to have the highest possible sampling rate: the EFIT equilibrium reconstruction code eventually determined the maximum achievable sampling rate of 5 ms.

Regarding the threshold needed to define the different class labels i.e. disruptive versus stable/non-disruptive sequences, we took into account several different factors, as mentioned in our dataset description section. First, we consulted tokamak operators to obtain the typical range of the unstable phase for the disruptive shots in each machine. We then studied the distribution of several plasma signals (li, kappa, lockmode, ip-error-fraction etc.), to further narrow down such

**Algorithm 1.** Algorithm for t-distributed stochastic neighbor embedding.

**Data**: data set $X = \{x_1, x_2, \ldots, x_n\}$
Cost function parameter: perplexity *Perp*,
Optimization parameters: number of iterations $T$, learning rate $\eta$, momentum $\alpha(t)$
**Result**: low-dimensional data representation $Y = \{y_1, y_2, \ldots, y_n\}$
**begin**
  Compute pairwise similarities $p_{ij}$ with *Perp* (using equations (1) and (2))
  Sample initial solution $Y^0 = \{y_1, y_2, \ldots, y_n\}$ from $N(0, 10^{-4}I)$
  **for** $t = 1$ **to** $T$ **do**
    Compute low dimensional similarities $q_{ij}$ (using equation (5))
    Compute gradient $g$ (using equation (6))
    Set $Y^t = Y^{t-1} + \eta g + \alpha(t)(Y^{t-1} - Y^{t-2})$
  **end**
**end**

**Table 5.** Normalization parameters for each signal in three machines and the common normalization set.

| Plasma signal | Mean EAST | Std EAST | Mean DIII-D | Std DIII-D | Mean C-Mod | Std C-Mod | Mean common | Std common |
|---|---|---|---|---|---|---|---|---|
| `ip-error-fraction` | −0.002 | 0.021 | −0.016 | 0.055 | −0.002 | 0.043 | −0.007 | 0.040 |
| `locked-mode-proxy` | 0.002 | 0.004 | $1.546 \times 10^{-4}$ | $3.503 \times 10^{-4}$ | $7.495 \times 10^{-4}$ | $4.364 \times 10^{-4}$ | $9.680 \times 10^{-4}$ | $1.596 \times 10^{-4}$ |
| `Greenwald-fraction` | 0.437 | 0.327 | 0.407 | 0.184 | 0.261 | 0.132 | 0.368 | 0.214 |
| `lower-gap` | 0.160 m | 0.032 m | 0.170 m | 0.082 m | 0.056 m | 0.016 m | 0.129 m | 0.043 m |
| `z-error-proxy` | 0.007 m | 0.022 m | 9.114 m | $0.006 \times 10^{-4}$ m | $-8.459 \times 10^{-7}$ m | $0.002 \times 10^{-3}$ m | 2.637 m | 0.010 m |
| `kappa` | 1.630 | 0.114 | 1.768 | 0.105 | 1.618 | 0.092 | 1.672 | 0.104 |
| `betap` | 0.689 | 0.380 | 0.826 | 0.501 | 0.239 | 0.184 | 0.585 | 0.355 |
| `radiated-fraction` | 0.138 | 0.347 | 0.516 | 1.237 | 0.369 | 0.952 | 0.341 | 0.845 |
| `rotating-mode-proxy` | 0.005 | 0.011 | $6.823 \times 10^{-5}$ | $1.554 \times 10^{-4}$ | 0.678 $S^{-1}$ | 1.062 $S^{-1}$ | 0.228 | 0.358 |
| `v-loop` | 0.430 V | 0.861 V | 0.293 V | 0.936 V | −0.329 V | 1.774 V | 0.131 V | 1.190 V |
| `q95` | 6.009 | 1.275 | 4.860 | 1.417 | 4.422 | 0.943 | 5.097 | 1.212 |
| `li` | 1.187 | 0.228 | 1.015 | 0.215 | 1.404 | 0.172 | 1.202 | 0.205 |

empirical ranges to those thresholds in time at which most distributions started to deviate from their stable counterparts (distributions of the non-disruptive phase), as disruptions were approached in these three tokamaks. Finally, for each machine, we scanned all the remaining time threshold 'candidates' and chose the one that maximized the performance on the validation set.

### *A.2. t-distributed stochastic neighbor embedding (t-SNE) algorithm*

As detailed in [27], t-SNE is a nonlinear dimensionality-reduction method that aims to convert the high-dimensional data set $X = \{x_1, x_2, \ldots, x_n\}$ into a low-dimensional manifold (usually 2D or 3D) set $Y = \{y_1, y_2, \ldots, y_n\}$ and preserve as much of the significant structure of the high-dimensional data as possible in its low-dimensional representation. The pairwise 'distance' in the low-dimensional map represents a matrix of pairwise similarities between objects: this can be visualized as capturing the local structure of the high-dimensional data but it also reveals global structures, such as the presence of clusters. In the resulting low-dimensional map, nearby points are grouped through similarity criteria. Therefore, two similar objects will appear as two close points, while two dissimilar objects will appear as separated points.

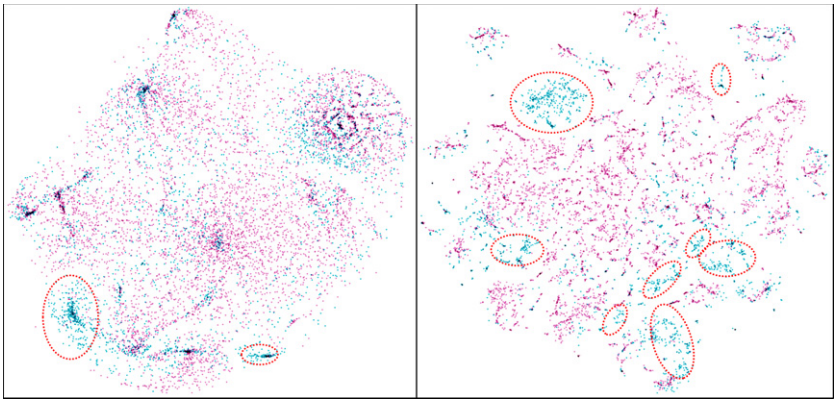

**Figure 8.** Clustering using t-SNE for visualization of EAST data. On the left, t-SNE is performed on individual disruption (cyan) and non-disruption (magenta) time slices, while on the right, t-SNE is performed on ten-step disruption (cyan) and non-disruption (magenta) sequences. The coloring is done *a posteriori*.

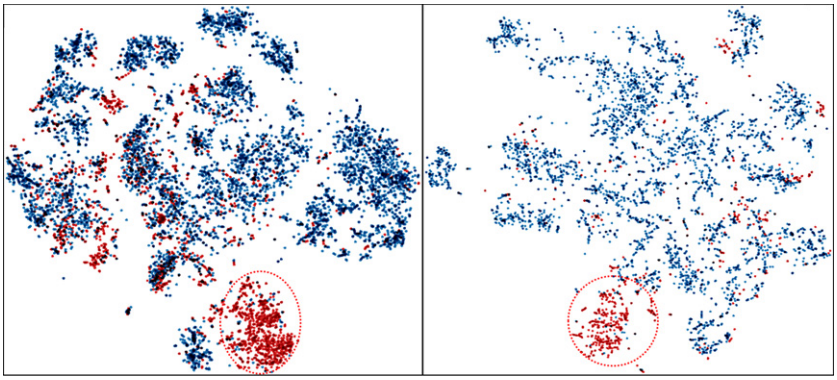

**Figure 9.** Clustering using t-SNE for the visualization of DIII-D data. On the left, t-SNE is performed on individual disruption (red) and non-disruption (blue) time slices, while on the right, t-SNE is performed on ten-step disruption (red) and non-disruption (blue) sequences. The coloring is done *a posteriori*.

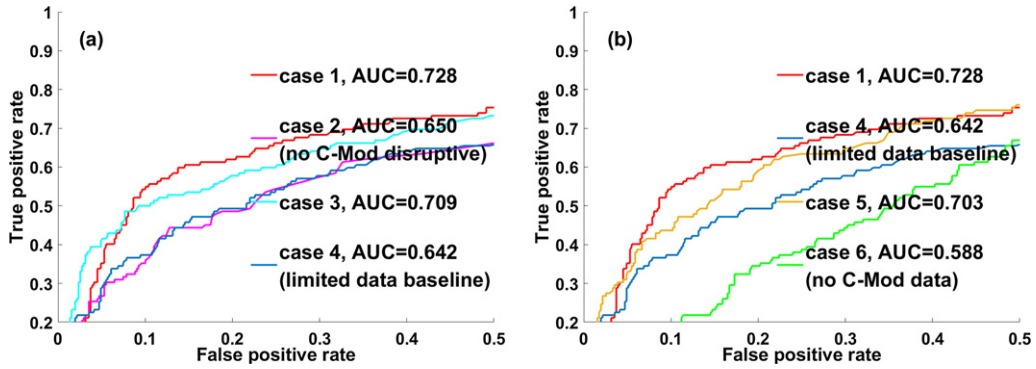


**Figure 10.** ROC curves from the C-Mod test set using limited C-Mod disruptive-training data.

The t-SNE algorithm 1 comprises three main stages. First, t-SNE constructs a conditional probability distribution over pairs of high-dimensional objects by converting the high-dimensional Euclidean distances between data points into conditional probabilities $p_{j|i}$ that represent similarities and defining the joint probabilities $p_{ij}$ as:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)} \tag{1}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2}$$

The standard deviation $\sigma_i$ is the standard deviation of the Gaussian that is centered over each high-dimensional datapoint, $x_i$. To find $\sigma_i$ for each Gaussian, t-SNE introduces a hyperparameter 'perplexity' and performs a binary search for the value of $\sigma_i$ that produces a probability distribution $P_i$ with

**Table 6.** Explanation of all hyperparameters and their recommended values.

| 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 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) |
| $N_{\text{F}}$ | Number of convolutional filters in the 1D | 10 |
| Optimizer | causal convolution sublayers of each MSTConv layer second-moment estimates | Adam |
| Dropout | Dropout probability | 0.1 |
| L2 regularization | Weight regularization of all weights | $1 \times 10^{-3}$ |
| $n_{\text{epoch}}$ | Number of training epochs | 32 |
| $d_{\text{latent}}$ | Dimension of final latent representation | 12 |
| $N_{\text{MSTConv}}$ | Number of MSTConv layers | 3 |
| $\varepsilon_{\text{1-C-Mod}}$ | Smoothing parameter $\varepsilon_1$ when tested on C-Mod | 0.00 |
| $\varepsilon_{\text{2-C-Mod}}$ | Smoothing parameter $\varepsilon_2$ when tested on C-Mod | 0.08 |
| $\varepsilon_{\text{1-DIII-D}}$ | Smoothing parameter $\varepsilon_1$ when tested on DIII-D | 0.00 |
| $\varepsilon_{\text{2-DIII-D}}$ | Smoothing parameter $\varepsilon_2$ when tested on DIII-D | 0.05 |
| $\varepsilon_{\text{1-EAST}}$ | Smoothing parameter $\varepsilon_1$ when tested on EAST | 0.09 |
| $\varepsilon_{\text{2-EAST}}$ | Smoothing parameter $\varepsilon_2$ when tested on EAST | 0.00 |

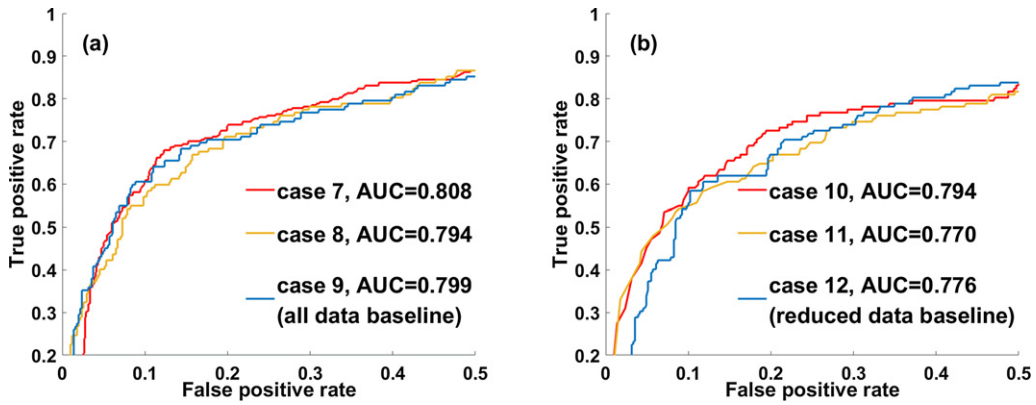


**Figure 11.** ROC curves from the C-Mod test set using all C-Mod disruptive-training data.

a fixed perplexity. The perplexity is defined as

$$\text{Perp}\,(P_i) = 2^{H(P_i)} \tag{3}$$

where $H(P_i)$ is the Shannon entropy of $P_i$ measured in bits:

$$H\,(P_i) = -\sum_j p_{j|i}\,\log_2\,p_{j|i}. \tag{4}$$

Second, t-SNE calculates a probability distribution over the points in the low-dimensional map using a student t-distribution [41] with a single degree of freedom. The joint probabilities $q_{ij}$ are defined as

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l}\left(1 + \|y_k - y_l\|^2\right)^{-1}}. \tag{5}$$

The t-SNE process will then minimize the Kullback–Leibler divergence (KL divergence) between the two distributions with respect to the low-dimensional representation using gradient descent. The gradient of the KL divergence between a high-dimensional distribution, $P$, and a low-dimensional distribution, $Q$, is given by

$$\frac{\partial C}{\partial y_i} = 4\sum_j \left(p_{ij} - q_{ij}\right)\left(y_i - y_j\right)\left(1 + \|y_i - y_j\|^2\right)^{-1}. \tag{6}$$

### *A.3. t-SNE clustering results for the EAST and DIII-D databases*

*A.3.1. EAST clustering result.* The figure above shows the t-SNE algorithm applied to time-slice data (left) and aggregated sequence data (right) for the EAST disruption-warning database. In the left subplot, each magenta point represents a randomly sampled time slice (a 12-element array composed of 12 plasma signals from table 5 in the main text) from the

**Table 7.** Training set composition of all cross-machine experiments using C-Mod as the 'new device'[a].

| | Existing machines (DIII-D + EAST) | | New machine (C-Mod) | |
|---|---|---|---|---|
| Case no. | Non-disruptive | Disruptive | Non-disruptive | Disruptive |
| 1 | None | All (732 + 2301) | All (2651) | 20 |
| 2 | None | All | All | None |
| 3 | None | All | 50% (1326) | 20 |
| 4 | None | None | All | 20 |
| 5 | All (4554 + 5995) | All | All | 20 |
| 6 | All | All | None | None |
| 7 | None | All | All | All (692) |
| 8 | All | All | All | All |
| 9 | None | None | All | All |
| 10 | None | All | ∼25% (662) | All |
| 11 | ∼29% (732 + 2301) | None | ∼25% | All |
| 12 | None | None | ∼25% | All |

[a]Values in parenthesis give the exact numbers of shots.

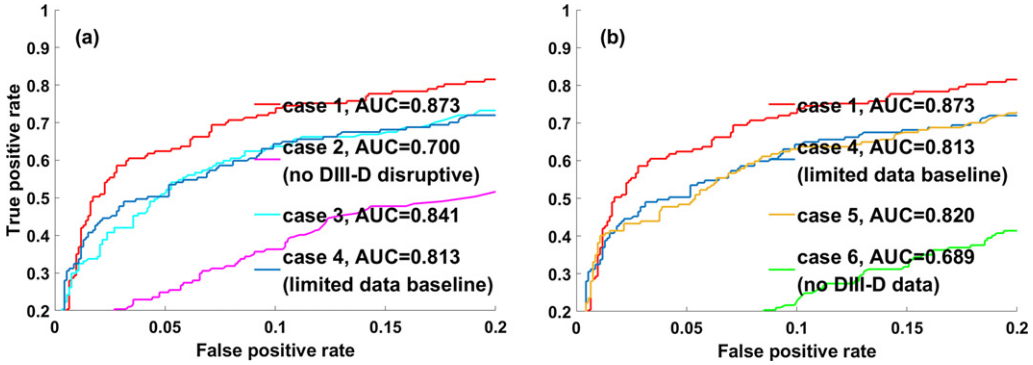


**Figure 12.** ROC curves from the DIII-D test set using limited DIII-D disruptive-training data.

**Table 8.** Training-set composition of all cross-machine experiments using DIII-D as the 'new device'[a].

| | Existing machines (C-Mod + EAST) | | New machine (DIII-D) | |
|---|---|---|---|---|
| Case no. | Non-disruptive | Disruptive | Non-disruptive | Disruptive |
| 1 | None | All (692 + 2301) | All (4554) | 20 |
| 2 | None | All | All | None |
| 3 | None | All | 50% (2277) | 20 |
| 4 | None | None | All | 20 |
| 5 | All (2651 + 5995) | All | All | 20 |
| 6 | All | All | None | None |
| 7 | None | All | All | All (732) |
| 8 | All | All | All | All |
| 9 | None | None | All | All |
| 10 | None | All | ∼15% (700) | All |
| 11 | ∼35% (692 + 2301) | None | ∼15% | All |
| 12 | None | None | ∼15% | All |

[a]Values in the parentheses give the exact numbers of shots.

flattop of a non-disruptive shot, while each cyan point represents a time slice randomly sampled from the last 300 ms of a disruptive shot. On the right, each cyan point represents a ten-step (a 10 × 12 element matrix) sequence randomly sampled from the last 300 ms of a disruptive shot, while each magenta point represents a ten-step sequence randomly sampled from the flattop of a non-disruptive shot. The coloring of each datapoint in the plots is done *a posteriori*, and therefore not provided during the training process, characterizing the t-SNE as an unsupervised clustering technique (figures 8–10).

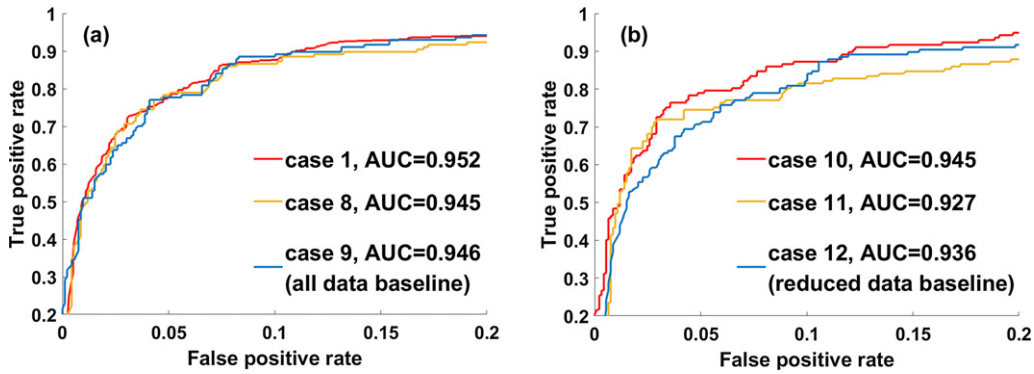


**Figure 13.** ROC curves from the DIII-D test set using all DIII-D disruptive-training data.

As can be seen from the two subplots, the clustering of individual time slices separates one major disruption cluster and one minor disruption cluster from the bulk of the non-disruption samples. However, more than half of the cyan points are still mixed with the magenta points. On the contrary, the clustering of the time-sequence data successfully separates one major disruption cluster as well as several smaller disruption clusters from the magenta points. There are only a few disruption samples ($<$15%) remaining mixed with the non-disruption samples. Again, migrating from time-slice clustering to time-sequence clustering gives considerably better separation of disruption and non-disruption regions.

*A.3.2. DIII-D clustering result.* The figure above shows the t-SNE algorithm applied to time-slice data (left) and aggregated sequence data (right) for the DIII-D disruption-warning database. In the left subplot, each blue point represents a randomly sampled time slice (a 12-element array composed of the 12 plasma signals from table 5 in the main text) from the flattop of a non-disruptive shot, while each red point represents a time slice randomly sampled from the last 200 ms of a disruptive shot. On the right, each red point represents a ten-step (a $10 \times 12$ element matrix) sequence randomly sampled from the last 200 ms of a disruptive shot while each blue point represents a 10-step sequence randomly sampled from the flattop of a non-disruptive shot. The coloring of each datapoint in the plots is done *a posteriori*, therefore not provided during the training process, characterizing the t-SNE as an unsupervised clustering technique.

The DIII-D case is slightly different from C-Mod and EAST cases. The clustering of individual time slices successfully separates one major disruption cluster from non-disruption samples, which accounts for $\sim$70% of all disruption samples. The clustering of time sequences also separates one major cluster from non-disruption samples which counts for $>$80% of all disruption samples. In the DIII-D case, we find that migrating from time-slice clustering to sequence clustering still provides better separation, but only marginally. And the clustering of time-slice data already gives good results. Further analysis shows that both major clusters in the two clustering schemes are related to locked-mode-induced disruptions. These results suggest that the features used to describe DIII-D data are well suited to identify locked-mode disruptions, independently of whether the data is described in terms of time slices or sequences. This reconnects with previously documented literature [40, 42], where various time-slice-based models (e.g. RF) worked well to describe the DIII-D instantaneous plasma state and sequence-based HDL was only able to slightly improve the performance over the results of the optimized RF predictor (see figure 12(*b*)). However, in order to detect various other types of disruption, we argue that a more suitable approach is the study of time sequences. Compared to C-Mod and EAST, DIII-D data seems to have some special characteristics, with a high prevalence of one particular class of disruptions (locked-mode-related disruptions), that makes both predictive approaches (time sequence vs time slices) comparably effective.

### *A.4. Normalization parameters for each signal*

See table 5.

### *A.5. Recommended hyperparameters for the HDL framework*

See table 6.

### *A.6. Cross-machine numerical experiments using C-Mod and DIII-D as target devices*

*A.6.1. Cross-machine experimental testing on C-Mod.* In these experiments, we consider DIII-D and EAST to be 'existing machines' and investigate the effect of their data on the HDL disruption predictor when used on C-Mod, chosen as a 'new device'. The numerical results shown in supplementary figures 3 and 4 (https://stacks.iop.org/NF/61/026007/mmedia) consistently support our previous cross-machine conclusions, to be found in section 5, cross-machine study. Notice that, given enough C-Mod data (cases in figure 11), continually adding data from existing machines (DIII-D, EAST) only slightly changes the performance for C-Mod. This fact implies that the C-Mod disruptions are quite different from the disruptions in other existing devices, which agrees with our findings from the clustering study. Therefore, with enough data from C-Mod, data from other existing devices barely help the disruption prediction for C-Mod. The detailed training-set composition for each case can be found in table 7.

*A.6.2. Cross-machine experimental testing for DIII-D.* In these experiments, we consider C-Mod and EAST to be 'existing machines' and investigate the effect of their data on the HDL disruption predictor when used for DIII-D, chosen as a 'new device'. The numerical results shown in supplementary figures 5 and 6 consistently support our previous cross-machine conclusions found in section 5, cross-machine study. The detailed training-set composition for each case can be found in table 8 (figure 13).

## ORCID iDs

C. Rea https://orcid.org/0000-0002-9948-2649
K. Montes https://orcid.org/0000-0002-0762-3708
R. Sweeney https://orcid.org/0000-0003-3408-1497
R.A. Tinguely https://orcid.org/0000-0002-3711-1834

## References

[1] De Vries P.C., Pautasso G., Humphreys D., Lehnen M., Maruyama S., Snipes J.A., Vergara A. and Zabeo L. 2016 *Fusion Sci. Technol.* **69** 471–84
[2] Wroblewski D., Jahns G.L. and Leuer J.A. 1997 *Nucl. Fusion* **37** 725
[3] Cannas B., Fanni A., Marongiu E. and Sonato P. 2004 *Nucl. Fusion* **44** 68
[4] Murari A., Vagliasindi G., Arena P., Fortuna L., Barana O. and Johnson M. 2008 *Nucl. Fusion* **48** 035010
[5] Murari A. *et al* 2018 *Nucl. Fusion* **58** 056002
[6] Rea C., Granetz R.S., Montes K., Tinguely R.A., Eidietis N., Hanson J.M. and Sammuli B. 2018 *Plasma Phys. Control. Fusion* **60** 084004
[7] Windsor C.G. *et al* 2005 *Nucl. Fusion* **45** e01
[8] Montes K.J. *et al* 2019 *Nucl. Fusion* **59** 096015
[9] Kates-Harbeck J., Svyatkovskiy A. and Tang W. 2019 *Nature* **568** 526–31
[10] Rattá G.A. *et al* 2018 *Fusion Sci. Technol.* **74** 3–22
[11] De Vries P.C., Johnson M.F., Alper B., Buratti P., Hender T.C., Koslowski H.R. and Riccardo V. 2011 *Nucl. Fusion* **51** 053018
[12] Sabbagh S.A. *et al* 2018 Disruption event characterization and forecasting in tokamaks *Preprint: 2018 IAEA Fusion Energy Conf.* (Gandhinagar, India, 22–27 October 2018) [EX/P6-26] (https://conferences.iaea.org/event/151/contributions/5924/)
[13] Berkery J.W., Sabbagh S.A., Bell R.E., Gerhardt S.P. and LeBlanc B.P. 2017 *Phys. Plasmas* **24** 056103
[14] Berkery J.W. *et al* 2017 Characterization and forecasting of global and tearingmodestability for tokamak disruption avoidance *44th EPS Conf. Plasma Physics* (Belfast, Northern Ireland, 26–30 June 2017) [P1.138] (http://ocs.ciemat.es/EPS2017PAP/pdf/P1.138.pdf)
[15] Maraschek M. *et al* 2018 *Plasma Phys. Control. Fusion* **60** 014047
[16] Piccione A., Berkery J.W., Sabbagh S.A. and Andreopoulos Y. 2020 *Nucl. Fusion* **60** 046033
[17] Xu Y. *et al* 2014 *Phys. Scr.* **2014** 014008
[18] Turco F., Luce T.C., Solomon W., Jackson G., Navratil G.A. and Hanson J.M. 2018 *Nucl. Fusion* **58** 106043
[19] Sweeney R., Choi W., La Haye R.J., Mao S., Olofsson K.E.J. and Volpe F.A. 2017 *Nucl. Fusion* **57** 016019
[20] ITER Physics Basis Editors *et al* 1999 *Nucl. Fusion* **39** 2541–75
[21] Ikeda K. *et al* 2007 *Nucl. Fusion* **47** E01
[22] Gerhardt S. *et al* 2013 *Nucl. Fusion* **53** 023005
[23] Cannas B., Fanni A., Sonato P. and Zedda M.K. 2007 *Nucl. Fusion* **47** 1559
[24] Vega J., Dormido-Canto S., López J.M., Murari A., Ramírez J.M., Moreno R., Ruiz M., Alves D. and Felton R. 2013 *Fusion Eng. Des.* **88** 1228
[25] Rea C., Montes K.J., Erickson K.G., Granetz R.S. and Tinguely R.A. 2019 *Nucl. Fusion* **59** 096016
[26] Pau A., Fanni A., Carcangiu S., Cannas B., Sias G., Murari A. and Rimini F. 2019 *Nucl. Fusion* **59** 106017
[27] Maaten L.V.D. *et al* 2008 *J. Mach. Learn. Res.* **9** 2579–605
[28] Cho K. *et al* 2014 *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)* (Doha, Qatar) 103–11
[29] Lee J., Cho K. and Hofmann T. 2017 *Trans. Assoc. Comput. Linguist.* **5** 365–78
[30] Oord A.V.D. *et al* 2016 WaveNet: a generative model for raw audio, https://arxiv.org/abs/1609.03499
[31] Ioffe S. *et al* 2015 *Proc. of the 32nd Int. Conf. on Machine Learning* (Lille, France) vol 37 pp 448–56
[32] Bradley A.P. 1997 *Pattern Recognit.* **30** 1145–59
[33] Hollmann E. *et al* 2015 *Phys. Plasmas* **22** 021802
[34] Goodfellow I. 2016 *Deep Learning* (Cambridge: MIT Press)
[35] Haykin S. 1999 *Neural Networks: A Comprehensive Foundation* 2nd edn (Englewood Cliffs: Prentice-Hall)
[36] Hansen L.K. and Salamon P. 1990 *IEEE Trans. Pattern Anal. Mach. Intell.* **12** 993–1001
[37] Perrone M.P. *et al* 1995 *How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems* (World Scientific) pp 342–58
[38] Abadi M. *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems tensorflow.org
[39] Meneghini O. *et al* 2015 *Nucl. Fusion* **55** 083008
[40] Montes K.J. *et al* 2019 *Nucl. Fusion* **59** 096015
[41] Helmert F.R. 1876 *Astron. Nachr.* **88** 113–31
[42] Rea C. *et al* 2018 *Plasma Phys. Control. Fusion* **60** 084004