A Semi-Supervised Machine Learning Detector for Physics Events in Tokamak Discharges

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

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

E-mail: kmontes@mit.edu

Abstract. Databases of physics events have been used in various fusion research applications, including the development of scaling laws and disruption avoidance algorithms, yet they can be time-consuming and tedious to construct. This paper presents a novel application of the label spreading semi-supervised learning algorithm to accelerate this process by detecting distinct events in a large dataset of discharges, given few manually labeled examples. A high detection accuracy ($> 85\%$) for H-L back transitions and initially rotating locked modes is demonstrated on a dataset of hundreds of discharges from DIII-D with manually identified events for which only 3 discharges are initially labeled by the user. Lower yet reasonable performance ($\sim 75\%$) is also demonstrated for the core radiative collapse, an event with a much lower prevalence in the dataset. Additionally, analysis of the performance sensitivity indicates that the same set of algorithmic parameters is optimal for each event. This suggests that the method can be applied to detect a variety of other events not included in this paper, given that the event is well described by a set of 0D signals robustly available on many discharges. Procedures for analysis of new events are demonstrated, showing automatic event detection with increasing fidelity as the user strategically adds manually labeled examples. Detections on Alcator C-Mod and EAST are also shown, demonstrating the potential for this to be used on a multi-tokamak dataset.

Keywords: disruption, event, detection, semi-supervised, machine-learning

1. Introduction

Large databases of labeled physics events have frequently been relied upon to facilitate progress in fusion research. These kinds of databases have helped increase understanding of multiple different physics phenomena, validate physical models, and produce widely relied upon scaling laws when model-based predictions are lacking. For example, manually identified time records occurring immediately before L-H transitions were collected from multiple tokamaks to construct a database from which the power required to access H-mode in ITER was deduced [1]. In a more recent study, plasma discharges

from the ASDEX Upgrade tokamak exhibiting initially rotating locked mode events were assembled to validate a tearing mode locking model [2]. Each of these studies was made possible only after an expert, or group of experts, collected a set of discharges and times at which the event of interest for the study had occurred.

Perhaps most striking in terms of database size and complexity is the study that motivated this paper - a survey of disruption causes on JET using over 2300 manually analyzed discharges [3]. In it, the authors labeled a variety of events corresponding to physics and technical problems that causally preceded a disruption. Most often, each disruption could then be traced back to a root cause by following the chain of events that preceded it. This required a tedious amount of manual analysis, yet produced a foundational framework for more recent development of a disruption event characterization and forecasting (DECAF) code [4–7].

Many alternative methods for disruption avoidance that utilize machine-learning have been developed. Only a few of them [8, 9], however, have used the framework described above to incorporate disruptive event chains. This is likely because machine-learning algorithms typically need to be trained and tested using many plasma discharges, so including these event chains would require investing considerable time and effort to manually label events of interest in each of the included disruptions (indeed, this limited studies like [8,9] to using approximately 100 disruptions). Therefore, most larger scale machine-learning studies for disruption avoidance have focused only on predicting the final disruptive event, rather than its precursors.

These approaches to disruption avoidance found in the literature to date can be broadly categorized as either *supervised* or *unsupervised* learning algorithms. For supervised learners, the target of prediction (usually the time of disruption, in this case) is provided, and the algorithm is trained to discriminate disruptive data from non-disruptive data using these labeled targets. Examples include decision tree ensembles [10–12], support vector machines [13,14], and neural networks [15–18]. For unsupervised learners, no labeled targets are provided. Rather, the objective is to uncover patterns in the data by probing its structure using some similarity metric. Examples include generative topographical maps [9], k-means clustering [19], and self-organizing maps [20].

In this work, the label spreading [21] algorithm is used to detect the types of events described above. This algorithm is known as a *semi-supervised* learner since it is designed to be used on large datasets for which only a few of the samples' targets are labeled. The algorithm is supervised in a sense, since it relies on a small number of labeled targets, but it also shares unsupervised characteristics because it utilizes the structure of the data and a similarity metric to deduce the targets of unlabeled examples (hence the term semi-supervised). Common applications of this algorithm include web categorization and multimedia classification [22], since the volume of data in these fields is often too large to manually label. We propose using it to construct a database of events of interest in an accelerated fashion after manually analyzing just a few representative examples, saving the fusion scientist both valuable time and effort. The proof of concept in this work thus presents the opportunity to use disruptive event chain information to

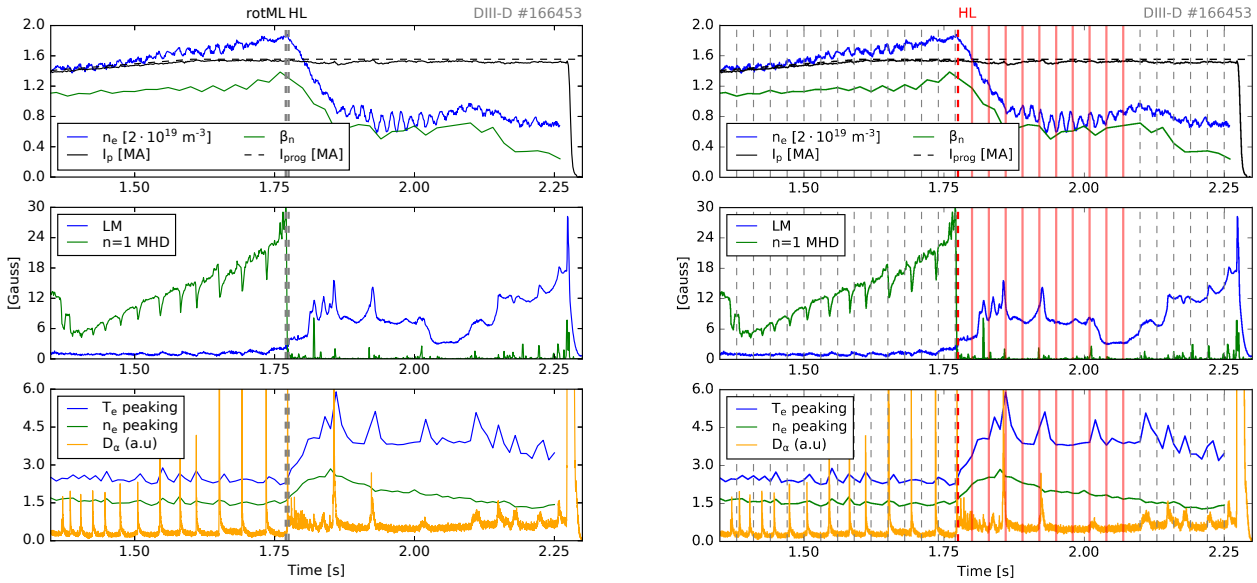
interpret and improve the performance of machine-learning predictors.

In the following Section 2, the label spreading algorithm is explained in detail along with a description of the data and tools used. The subsequent Section 3 demonstrates the application of the same semi-supervised learner to detect three different events that often precede disruptions. The robustness of this specific choice of semi-supervised algorithm to changes in parameterization is discussed in Section 4. Further applications of the algorithm and caveats for its use are discussed in Section 5, and the paper is finished with a summary and main conclusions for the reader in Section 6.

2. Methodology

In order to develop the proposed event detection algorithm, a set of over 300 flattop disruptions from the 2015 and 2016 DIII-D campaigns was manually analyzed in a manner similar to [3]. Only lower single null configuration shots were considered for the following analyses, so that peaking factor signals [23] could be incorporated to track the poloidal movement of radiative structures in the plasma. For each discharge, the chain of events leading to disruption was studied using a variety of routine diagnostics, and each corresponding event was recorded along with the time at which it initiated. An example is shown in Figure 1(a), which shows a discharge that disrupted due to a growing locked mode. The chain of events begins with a rotating n=1 tearing mode, made evident by the rising RMS amplitude of the $n = 1$ MHD signal. This signal drops as the mode locks near 1.77 s, coinciding with an increase in the locked mode proxy signal LM, which measures the $n = 1$ Fourier component of the radial magnetic field, $B_r^{n=1}$. Around the same time, the plasma evolves from an H-mode phase to an L-mode phase. The onset of this event, the H-L back transition, is indicated by 'HL' in the figure. Signatures of the event include a drop in the density and a loss of ELMs as shown by the D_α signal. Additionally, one can see the loss of the pedestal characteristic of an H-L transition reflected in the T_e and n_e profile peaking factors, which give the ratio of the value at the core to its average across the profile [23].

Now, suppose we would like to gather other discharges with similar events and record the times at which these events occur. Instead of repeating this manual analysis for each successive shot, we can attempt to detect the event occurrences by following a procedure utilizing the semi-supervised approach. To start, time sequences representing the evolution of the plasma over distinct periods are extracted from the data for all shots. These sequences must have a duration large enough to cover the event timescale, enough time steps to resolve the event dynamics, and overlap frequently enough to avoid missing any event occurrences. For each event analyzed in this paper, sequences of $S = 6$ equally spaced time steps each are chosen and a new sequence is sampled every 30 ms, meaning that the endpoints of any two neighboring sequences are 30 ms apart (as seen in Figure 1(b)). Though this sampling period is somewhat arbitrarily chosen, it should be small enough so that most of each shot is sampled and so that each event occurrence is well represented by at least one sequence that completely encompasses the



(a) (b)

Figure 1: This shot has a chain of events, marked in (**a**) by dashed vertical lines, that starts before 1.4 s with a growing n=1 rotating tearing mode which locks [rotML] before an H-L back transition [HL]. Sequences drawn in uniform intervals from the shot are represented in (**b**) by vertical lines at their endpoints. Those overlapping with the time of the event to be detected (HL) are classified as positive and have endpoints shown as red solid lines, whereas others are classified negative and shown with dashed gray lines.

event. Sequence durations are chosen to match the event timescale (see Table 1). At each time in each sequence, the values of N features ($0D$ signals) relevant for identifying the event are recorded. Therefore, each sequence $\vec{x}$ is now represented as a single point in an $N \times S$-dimensional space, i.e. $\vec{x} \in \mathbb{R}^{N \cdot S}$. In order to more sensibly compare Euclidean distances in this space, each of the N signal distributions is standardized, or scaled and offset so that its mean $\mu = 0$ and its standard deviation $\sigma = 1$.

The sequences extracted from the manually analyzed shot(s) are now placed into two distinct classes based on when they occur relative to the time of the event of interest. Each sequence in this set of ℓ labeled time sequences $X_L = \{\vec{x}_1, \ldots, \vec{x}_\ell\}$ is classified as positive ($y_i = 1$) if it overlaps with the event occurrence, and negative ($y_i = -1$) otherwise (see Figure 1(b)). For all shots not manually analyzed, the class of each time sequence is unknown because event occurrences have not been identified. Therefore, each sequence in this set of u unlabeled time sequences $X_U = \{\vec{x}_{\ell+1}, \ldots, \vec{x}_{\ell+u}\}$ is given a placeholder class $y_i = 0$. Formally, the goal of semi-supervised learning is to infer the set of labels $\{y_{\ell+1}, \ldots, y_{\ell+u}\}$ from the set of initial labels $\{y_1, \ldots, y_\ell\}$ using the structure of the entire dataset $X = X_L \cup X_U$. Typically $\ell \ll u$ since most samples are not labeled, or manually analyzed, by the user, as is the case for all applications in this paper.

2.1. Label Spreading

The specific semi-supervised learning algorithm used in this study is called label spreading [21]. It can be thought of as a variant of the simpler label propagation algorithm [24]. Both algorithms frame the dataset X as a fully connected graph of $n = \ell + u$ nodes in $\mathbb{R}^{N \cdot S}$, where each node corresponds to a unique sequence $\vec{x}_i \in X$. The edges connecting each pair of nodes $\vec{x}_i, \vec{x}_j$ have weights $w_{ij} = f(\vec{x}_i, \vec{x}_j)$, where f is a kernel function that gauges the proximity of the two sequences. It defines the degree to which any two sequences are similar to each other, reflecting the models' foundational assumption - that data points which lie close together should have similar labels. Lastly, Y is an $n \times 1$ vector representing the values at each node. Its i^{th} component is a value $0 \leq Y_i \leq 1$ representing the probability that the corresponding node $\vec{x}_i$ is in the positive class (it overlaps with the event of interest). Since each algorithm iteratively updates this vector of probabilities, we will write $Y(t)$ to refer to the vector at iteration t.

The label propagation algorithm is the simpler, more intuitive alternative. It updates the probabilities Y using a transition matrix T via the rule $Y(t+1) = TY(t)$. Each transition matrix element $T_{ij} \propto w_{ij}$ represents the probability that node j will be assigned the value Y_i of node i. Since this probability increases with the weight, and therefore the similarity of the two nodes being compared, the initial labeled information is propagated iteratively to similar nodes. On each iteration, Y is row normalized after applying the update rule and the values of the initially labeled nodes are *clamped*, or reset, back to their original values. This ensures that the manually verified, initially labeled information is retained on each iteration.

The label spreading procedure is similar to that of label propagation, with modifications to make the algorithm more robust. In this case, the dataset's network graph has no self-loops, so that $w_{ii} = 0$ for all i. On the first iteration ($t = 0$), we set $Y_i(0) = 1$ if $y_i = 1$ and $Y_i(0) = 0$ otherwise. Then, the vector Y is updated iteratively until convergence via the rule

$$Y(t+1) = \alpha TY(t) + (1-\alpha)Y(0) \tag{1}$$

where $0 < \alpha < 1$ and T is an $n \times n$ *transition matrix* with elements T_{ij} given by

$$T_{ij} = \frac{w_{ij}}{(\sum_k w_{ik})^{\frac{1}{2}} (\sum_k w_{jk})^{\frac{1}{2}}} \tag{2}$$

Here, α is known as the *clamping factor* since it introduces a 'soft clamping' effect, rather than the hard clamping done after each iteration in label propagation. Since the first term in Equation 1 passes information from neighboring nodes and the second term passes initially labeled information, the choice of α determines to what extent these two pieces of information influence the value at any given node. Unless otherwise noted, the detection algorithms in this work use $\alpha = 0.05$ so that the information from the manual labeling process is heavily weighted. All applications in this work utilize the label spreading procedure as implemented in the scikit-learn [25] Python library‡.

‡ https://scikit-learn.org/stable/modules/label_propagation.html

3. Detection of Physics Events

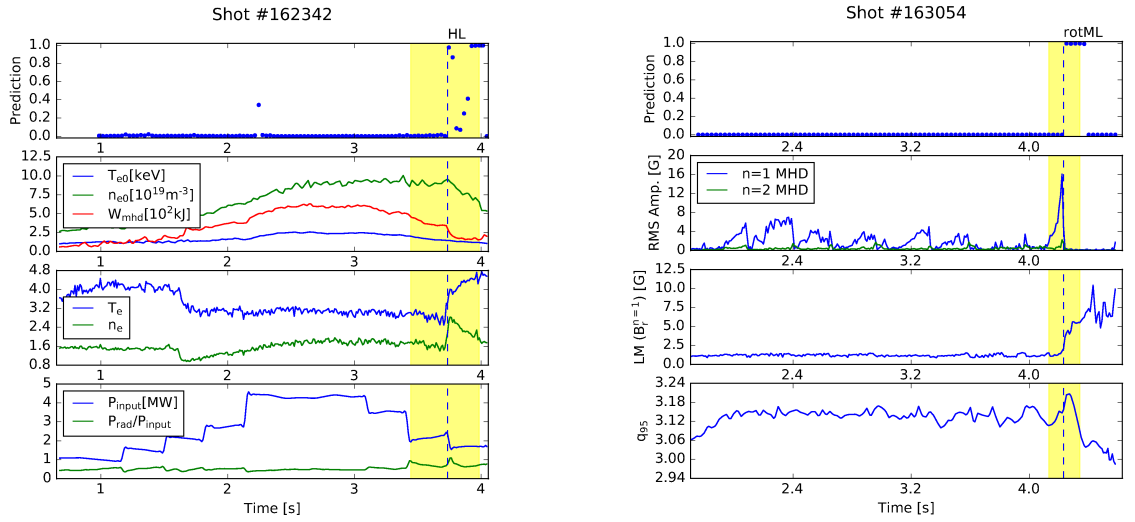
To demonstrate a generalizable event detector, label spreading was used to separately search for three distinct events in the dataset. The same algorithm was used to detect each event, so that the only differences amongst applications are the underlying feature space used to describe the data and the initially labeled examples. Since each of these events typically evolves over a few hundred milliseconds, the same sequence properties discussed in Section 2 were used for each case. For wide applicability to the different multi-dimensional spaces that describe each event, a radial basis function given by

$$f(\vec{x}_i, \vec{x}_j) = \exp(-\gamma D_M(\vec{x}_i, \vec{x}_j)^2) \tag{3}$$

is used as the kernel, where

$$D_M(\vec{x}_i, \vec{x}_j) = \sqrt{(\vec{x}_i - \vec{x}_j)^{\mathrm{T}} \Sigma^{-1} (\vec{x}_i - \vec{x}_j)} \tag{4}$$

is the Mahalanobis distance and Σ is the sample covariance matrix of the standardized sequence distribution. For each event in this section, $\gamma = \frac{1}{2}$ is used so that f is a Gaussian of standard deviation $\sigma = 1$ that more heavily weights neighboring nodes within a Mahalanobis unit. Finally, only 3 initially labeled shots are used for each case, representing only $\sim 1\%$ of the shots in the dataset.



(a) (b)

Figure 2: Detections of an H-L back transition (**a**) and initially rotating locked mode (**b**). The top subpanel shows the label spreading prediction, whereas the others show the signals used as inputs to the algorithm. Manually identified event times are marked by vertical dashed lines, and automatic detection regions are highlighted.

The label spreading algorithm was applied to detect three distinct events in the dataset: H-L back transitions (HL), initially rotating locked modes (rotML), and core radiative collapses (coreRC). Occurrences of the first two events were briefly discussed

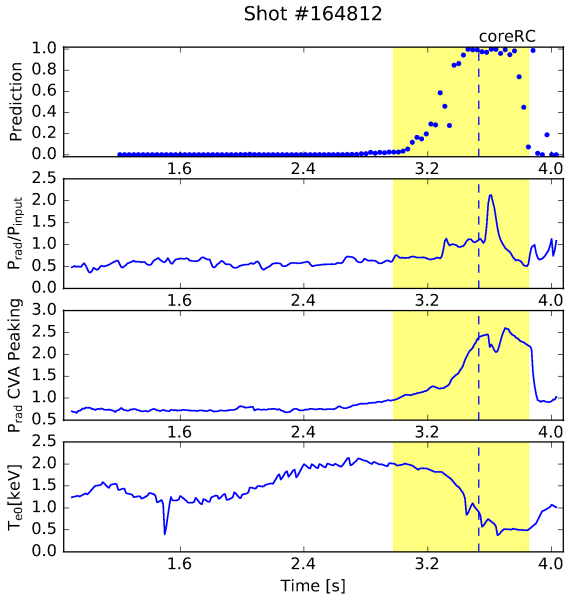


Figure 3: A detection of a core radiative collapse following a period of impurity accumulation in the core plasma.

in Section 2 and shown in Figure 1(a), and both were highly prevalent in the dataset (occurred in 76% and 64% of shots, respectively). For the H-L back transition, $N = 7$ signals were chosen to describe the event dynamics. They consist of the core electron temperature T_{e0} and density n_{e0}, the magnetic stored energy W_{mhd}, the electron temperature and density profile peaking factors, and the input power P_{input} and radiated power fraction P_{rad}/P_{input}. For the initially rotating locked mode, only $N = 4$ signals were used: the locked mode proxy signal ($n = 1$ Fourier component of the radial magnetic field), the $n = 1$ and $n = 2$ RMS MHD amplitudes, and q_{95}. These signal sets are shown for two example event detections in Figure 2. For each detection, the output probability Y_i from each sequence $\vec{x}_i$ is plotted in the top subpanel, where each point marks the endpoint of the sequence it represents. To aid the viewer, the time period overlapping with positive sequence predictions is highlighted in yellow and referred to as the *detection region*.

Unlike the first two events, the core radiative collapse is a low prevalence event as it only occurs in $\sim 8\%$ of shots. It is characterized by radiated power $P_{rad} > P_{input}$ predominantly from the core region of the plasma and a corresponding drop in core temperature T_{e0} as energy is lost from the core. As such, only $N = 3$ signals are chosen to identify it: the radiated power fraction, electron core temperature, and a core radiation peaking factor [23]. These signals are shown with an example core radiative collapse detection in Figure 3.

The examples shown in Figures 2 and 3 are characteristic of the majority of the shot predictions in the dataset. Most of the prediction values are near 0 when there is no event present, and they rise to values near 1 when the event of interest is near.

Table 1: Properties and performance statistics [true positive rate (TPR) and false positive rate (FPR)] for each event detection application in Section 3 when only 3 shots ($\sim 1\%$) are initially labeled.

Event	Sequence Duration	Prevalence	TPR|FPR	Input Signals
HL	300 ms	76%	88%|19%	T_{e0}, n_{e0}, W_{mhd}, P_{input}, $\frac{P_{rad}}{P_{input}}$, T_e & n_e peaking
rotML	150 ms	64%	84%|2%	LM proxy, $n = 1$ & 2 MHD, q_{95}
coreRC	300 ms	8%	75%|21%	$\frac{P_{rad}}{P_{input}}$, P_{rad} core peaking, T_{e0}

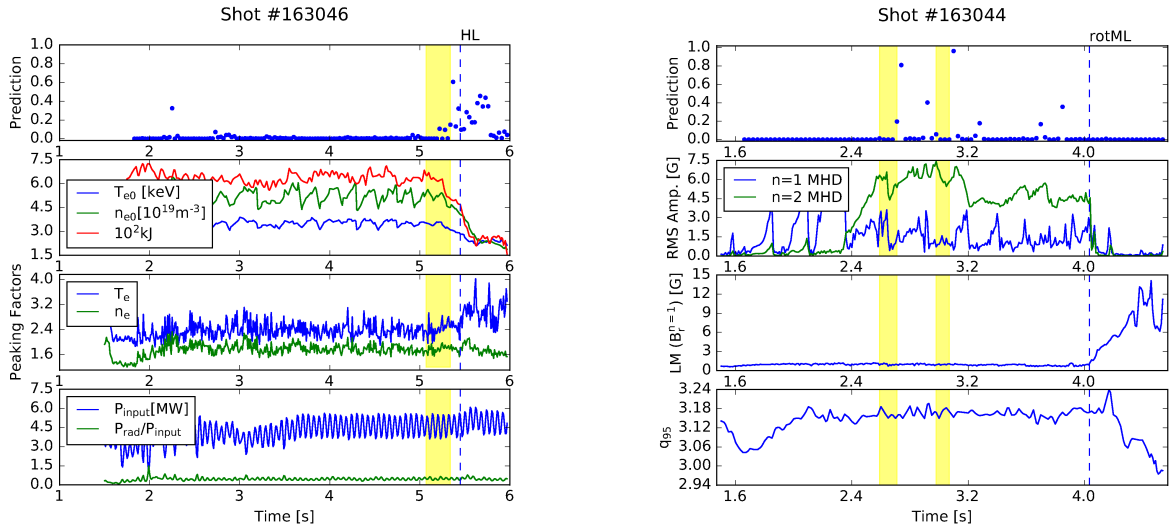
This rise is generally smooth for longer timescale events (Figure 3), but can fluctuate when shorter timescale phenomena are important. To gauge the overall performance of the algorithm, each shot with a detection is classified as a true positive if the manually identified event time lies within the detection region, and as a false positive if no event occurs for that shot. Dividing the number of true and false positives by the number of initially unlabeled shots with and without an event, respectively, yields the true (TPR) and false (FPR) positive rates. These are reported along with other relevant parameters in Table 1 to summarize this section.

3.1. Failure Modes

Exploratory analysis of false positive and negative cases reveals that the metrics reported above (and similarly in Sections 4 and 5) are conservative estimates of performance. One reason for this is that, for all 3 events, a fraction of the false negatives under the criteria used had detection regions that were near the event time, but did not include it. An example is shown in Figure 4(a), where an H-L back transition is detected by the algorithm before the time at which the start of the event was manually recorded. It is clear that the algorithm is responsive to the event after this manually recorded time as well, but the prediction level does not exceed 50% confidence and thus does not trigger a detection. This problem in particular was found to significantly impact the true positive rate of the core radiative collapse case, and accounting for early and late detections alone increases the true positive rate for this case to up to 91%.

Other false negative cases are harder to understand without more detailed analysis. An example is shown in Figure 4(b), where the algorithm fails to detect the rotating mode that locks just after 4.0 s, yet detects spurious phenomena early on in the discharge. Perhaps the algorithm is picking up on the coinciding drops in the $n = 2$ amplitude at those times, but it is unclear why the actual locking event is missed later on.

Finally, there are cases where false positive detections coincide with relevant signal behavior, even though these events do not satisfy the event definitions given above. This is an important factor in the core radiative collapse case, as many of the false positives



(a) (b)

Figure 4: False negative cases for the H-L back transition (**a**) and initially rotating locked mode (**b**).

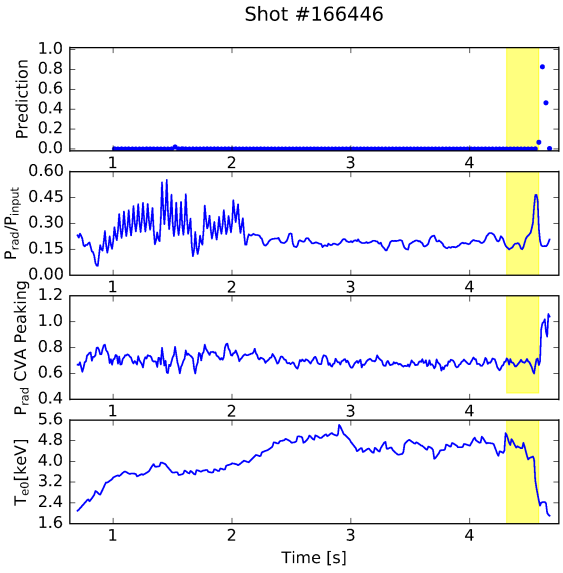


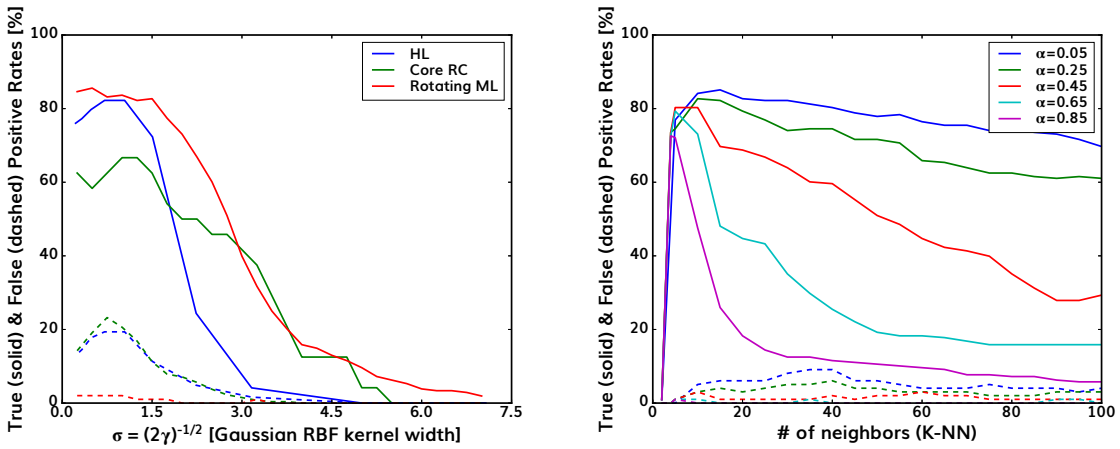
Figure 5: A false positive core radiative collapse detection; although T_{e0} decreases as radiation peaks in the core, the radiated power only reaches half of the input power.

are cases where the algorithm detects significant core radiation. An example is shown in Figure 5, which is classified as a false positive because the coreRC definition is not satisfied since the radiated power did not exceed the input power during the 'collapse'. Accounting for these types of cases alone could lower the coreRC false positive rate to as low as 11%. This reflects the difficulty and somewhat subjective nature of defining a precise event time and type for each event occurrence, which could be considered as one of the shortcomings of this approach.

4. Sensitivity to Hyperparameters

To obtain the event detections described in Section 3, the clamping factor (α), number of initial labels, kernel function type (f), and kernel parameterization (γ) were all fixed a priori. The extent to which the algorithm's predictive performance is sensitive to each of these parameters can reveal something about the robustness of the results, and potential applications to new events. To begin to explore this, we can compare the performances obtained when varying the kernel and clamping factor. We will then briefly discuss the impact of changes in the initial labels provided.

Broadly, the algorithm's performance is sensitive to the width, or scale, of the kernel. As shown in Figure 6(a), increasing γ in the Mahalanobis radial basis function (see Equation 3), and thereby decreasing the width of the kernel, results in a large gain of true positives for each event. However, this comes at a cost of false positives, and eventually hits diminishing returns near $\sigma = 1$ ($\gamma = \frac{1}{2}$) as chosen a priori. A similar pattern is observed for the simpler k-nearest neighbors (k-NN) kernel, for which



(a) (b)

Figure 6: Performance sensitivity to kernel width for (a) all events using the Mahalanobis kernel (Equation 3) and (b) the initially rotating locked mode event using the k-NN kernel with varying clamping factor, α.

$f(\vec{x}_i, \vec{x}_j) = 1$ for the k sequences $\vec{x}_j$ closest to $\vec{x}_i$ (using Euclidean distance) and is 0 otherwise. As shown in Figure 6(b), performance for the detection of the initially rotating locked mode event increases as the number k of nearest neighbors decreases, but reaches a maximum around $k = 10$ before sharply dropping. In general, then, the evidence suggests that there is some optimum kernel width for event detection. It must be small enough to maintain a non-linear decision function, but large enough to avoid overfitting.

Another trend worth noting is the sensitivity to the clamping factor, α. This sensitivity is low when using the Mahalanobis kernel in Equation 3, yet increases for other kernel types like the k-NN kernel. In all cases observed, though, it appears that performance is generally better for lower values of α, corresponding to a harder clamping effect and a greater retention of initially labeled information. This is expected, as the initially labeled times were manually verified and should be mostly retained. An example of this sensitivity is shown in Figure 6(b). Performance varies little for number of neighbors $k < 10$, but diverges widely as k increases, showing a trend of increasing performance as α decreases. Though not shown, similar trends are observed for the HL and coreRC events with optimal performance near $k = 10$ and an inverse correlation between clamping factor and performance.

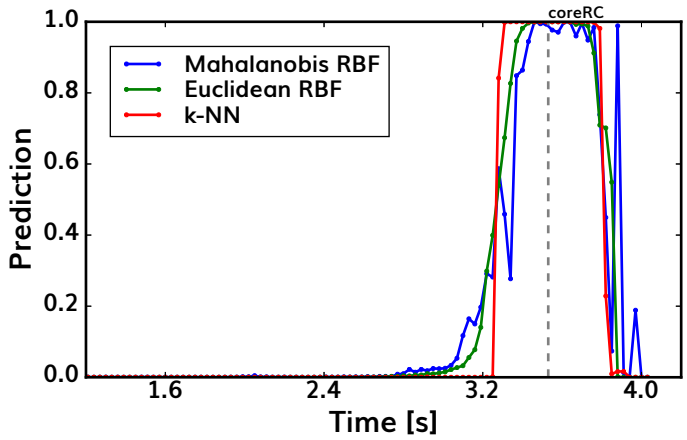
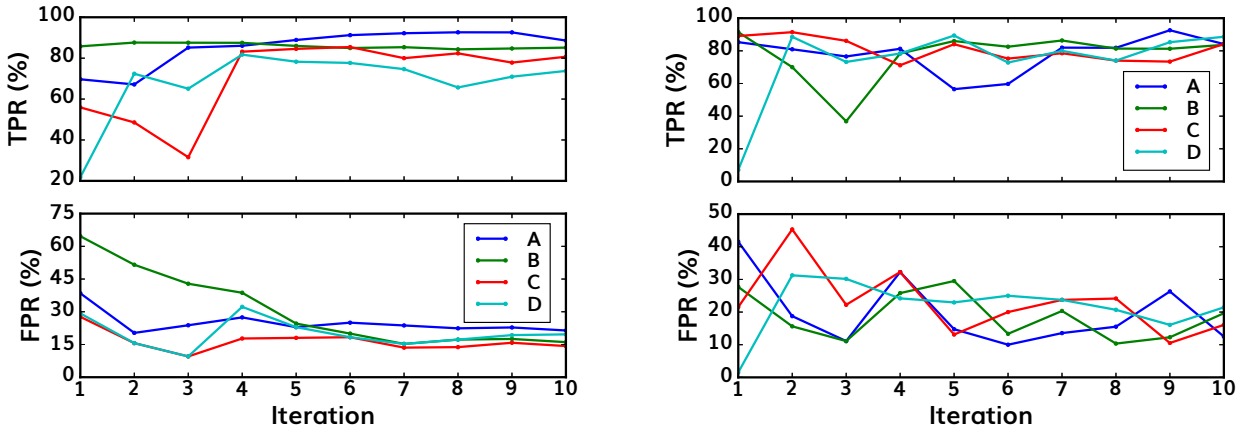


Figure 7: Comparison of core radiative collapse predictions for shot 164812 (see Figure 3) using the k-NN and RBF (with Mahalanobis and Euclidean distance) kernels.

The type of kernel function used also has a significant effect on the quality of individual predictions, even if those predictions agree to a certain extent. As an example, a comparison of predictions using three different kernels is shown in Figure 7 for the shot with a radiative collapse first shown in Figure 3. Note that the k-NN kernel has a sharper, almost binary transition near the event time. However, the Mahalanobis and Euclidean radial basis function kernels yield smoother predictions. Although the detection regions for each kernel are similar in this case, a smoother prediction may be useful in some contexts (for example, smooth predictions make it easier to search

for marginal detections, discussed in Section 5). This comparison of varying kernel smoothness was also observed for each event on multiple shots.



(a) (b)

Figure 8: True (TPR) and false (FPR) positive rates for H-L event detection when number of initially labeled shots increases by 2 iteratively. In (**a**), each iteration randomly adds shots to the previous iteration's set. In (**b**), all initially labeled shots are randomly selected each iteration. Each case (A-D) uses a different random seed.

Recall that all applications discussed so far have used the same sets of 3 initially labeled shots corresponding to each event. Of course, performance results are sensitive to the initially labeled shots and where the corresponding labeled sequences $\vec{x}_i$ lie in the feature space $\mathbb{R}^{N \cdot S}$. This sensitivity should be dependent on the structure of the manifold in $\mathbb{R}^{N \cdot S}$ describing the set of positive sequences, how well that structure is resolved by the sequences available in X, and which parts of that structure are accessible via the initially labeled sequences. In practice, this is dependent on the nature of the event of interest, the signals and samples chosen to describe that event, and the types of examples that are initially labeled. This is illustrated in Figure 8, which shows performance results for changing initial conditions. In each case (A-D) a single discharge with an H-L back transition is randomly chosen as the initial label, and the corresponding true and false positive rates are reported for iteration 1. On the second iteration in Figure 8(a), a randomly chosen pair of shots (one with an H-L event and one without) is added to the set of initially labeled shots from the previous iteration, and this is repeated for future iterations. Note that at first, when the number of initial labels is small, the cases vary widely in performance. Perhaps this is because only part of the manifold is accessible using the initially labeled sequences from the first shot. However, this variance decreases and the performance statistics begin to converge (albeit inefficiently) as initially labeled shots are randomly added and the manifold is better sampled. The same phenomenon is observed in Figure 8(b) when all initially labeled shots chosen on each iteration do not depend on those of the previous iteration. The increase in performance to some

early saturation level shown in Figure 8 is also a general observation, as it is observed for the rotML and coreRC event cases as well.

5. Applications

In Section 3, it was demonstrated that label spreading can be applied to detect three distinct events of varying prevalence using the same algorithmic parameters. Section 4 showed that this choice of parameters is robust, yielding reasonable performance regardless of the event type. Therefore, it is reasonable to expect that this algorithm can be applied to detect any arbitrary event, given that the event is described well by a set of signals robustly available on many shots. Typically, the user will not have access to a large dataset of labeled events like the one used in this study, but may have a few manually identified examples. With this in mind, methods for exploratory analysis of a large set of shots are suggested in this section.

5.1. Labeling From Scratch

Suppose one starts with a single example of a new event and wants to use it with the label spreading algorithm to construct a database of many examples of this event. One could train a label spreader using this single shot as the initial label, but it would be difficult to gauge its overall performance without manually identified events to compare with the predictions. However, one can always access the distribution of all sequence predictions in the dataset, which normally has two large peaks near 0 and 1 representing negative and positive predictions. For a perfect predictor, the relative magnitudes of these two peaks can be calculated if the event prevalence is known. The probability that the prediction $Y_i = 1$ for a perfect predictor and randomly selected sequence $\vec{x}_i$ is

$$p(Y_i = 1) \approx Prevalence \times \frac{Event\ Duration}{Mean\ Shot\ Duration} \tag{5}$$

if the event occurs at most once per shot. This is true for most shots in the dataset, so Equation 5 is a reasonable approximation (to be more exact, the mean shot duration can be subsumed into the prevalence, which would then be in units of events per shot per second). Taking the prevalence from Table 1, the mean event duration as the sequence duration (300 ms), and the mean shot duration in the dataset as $\approx 2.7s$, we expect $p(Y_i = 1) \approx 8\%$ and $p(Y_i = 0) \approx 92\%$ for a perfect detector of H-L back transitions (see Figure 9 for a comparison with actual prediction distributions). If an estimate of the prevalence can be made, then one may be able to estimate the expected performance of an imperfect predictor by using its distribution to compare the observed peak values to their corresponding 'perfect' values.

In practice, the prevalence may not be known and a significant fraction of intermediate predictions $0 < Y_i < 1$ exist. In particular, the intermediate values near ~ 0.5 correspond to nodes that have a roughly equal number of positive and negative neighboring nodes at the final iteration. Two examples of these distributions from label spreaders with comparable true positive rates, each trained on a different initially labeled

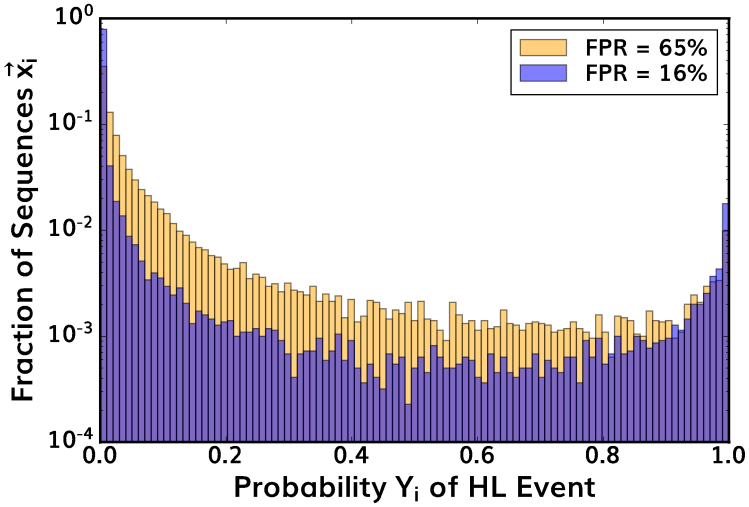
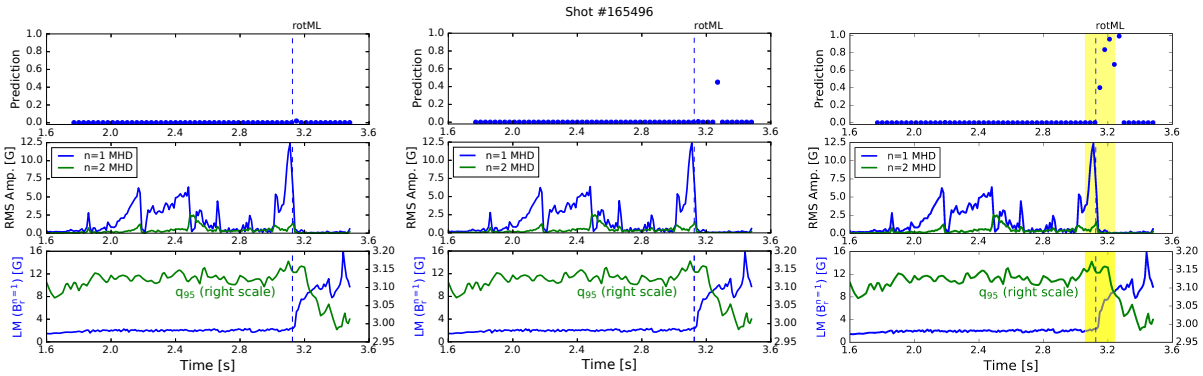


Figure 9: Prediction distributions for two identical label spreaders with different initial labels (from iterations 1 and 10 of Figure 8(a), case B). The less confident predictor (with a higher intermediate value fraction) has a higher overall false positive rate.



(a) (b) (c)

Figure 10: Initially rotating locked mode prediction for a single unlabeled shot after the 1st (a), 2nd (b), and 3rd (c) iterations of the 'from scratch' labeling method. The prediction in (b) is a typical example of a 'marginal' detection.

shot, are shown in Figure 9. As illustrated in the figure, the detector with a higher false positive rate has a significantly larger fraction of intermediate predictions. In general, an inverse correlation between the fraction of intermediate sequence predictions and the overall performance has been observed. This suggests that the degree of prediction confidence likely reflects the degree to which the target manifold is resolved, and that it can thus be used as a rough proxy for relative performance in the absence of manually verified ground truth labels.

The above observation motivates an iterative method for constructing an event

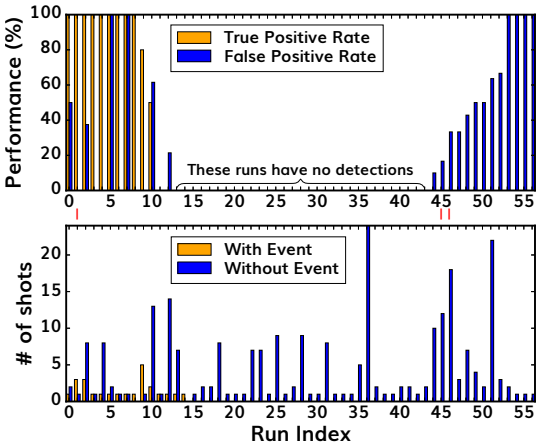
database from scratch, when only a single example of the event is given. On the first iteration, a label spreader is trained using initially labeled sequences all from a single manually analyzed shot. Then, the prediction distribution is used to search for a 'marginal' detection, or a shot where predictions $Y_i \sim 0.5$ so that an event was barely detected or barely missed (see Figure 10(b) for an example). The user can then manually analyze this shot to validate whether or not an event occurred. Afterward, this shot is added to the set of initial labels and another label spreader is trained. This can be repeated iteratively until the user has confidence in the performance, using the prediction distribution and the quality of individual predictions as a guide. This method was implemented to detect initially rotating locked modes, starting with the single shot shown in Figure 1(a). Using this single shot, a true positive rate of 50% was achieved with no false positives. By the 3rd iteration, the true positive rate on unlabeled discharges was increased to 85% with just a 2% cost in false positives. As shown in Figure 10, the quality of individual predictions also increased.

5.2. Experimental Search Engine

Another approach to exploratory analysis involves correlating the detections with a database of experiments, or 'runs', describing the operational plan for each set of shots. This was done using the label spreading detection results for core radiative collapses described in Section 3 and the 'summaries' SQL database available at DIII-D. The shots in the dataset used for core radiative collapse detection are distributed amongst almost 60 different runs at DIII-D. In Figure 11, each of these runs is assigned an index and the true and false positive rates for each run are plotted in order of decreasing performance along with the corresponding number of shots both with and without an event. Three red vertical lines are placed between these two plots at the runs from which the initially labeled discharges were drawn, and run indices with positive predictions are shown with their corresponding names in the table at the right.

A few observations may be made at first glance. Firstly, roughly half of the runs have no core radiative collapse event and no positive detections. This indicates that the detector has appropriately narrowed the range of experiments for which there may be a core radiative collapse, increasing the likelihood that the user may discover the event of interest. Indeed, if the set of shots is restricted to just those with positive detections, the information in Table 1 indicates that the prevalence increases by a factor of 3. Secondly, the majority of true positive event detections occurred in runs other than those from which initial labels were drawn. This suggests that the detector is not overfitting to one specific type of regime or operational sequence.

Upon further inspection of the runs with predictions, one may notice some common themes. For example, runs 2, 4, and 10 (highlighted in blue) all occurred during the 'metal ring' campaign on DIII-D in 2016 [26–28]. During this campaign, a tungsten ring was placed near the divertor, and it spawned a few events in which impurities accumulated in the plasma and led to a core radiative collapse. A similar mechanism



	Run Days with Positive Predictions
0	Optimizing Radiative Divertor at High Power Density in High Beta Hybrids
1	Isotope scaling L-H and H-L power thresholds in hydrogen and deuterium plasmas
2	Impact of High Power AT Operation on Divertor and Core using W-tiles
3	ELM Suppression in Steady State Hybrid Plasmas
4	End of Metal tiles campaign tasks
5	High Frequency D2 Pellet ELM Pacing
6	Plasma Startup and Systems Checkout - Day 2
7	Detachment onset at the inner and outer divertor
8	Plasma Startup and Systems Checkout - Day 7
9	Zonal Flow Damping in L-H Transitions
10	Effect of RMP ELM Control on W Divertor Erosion and SOL Transport

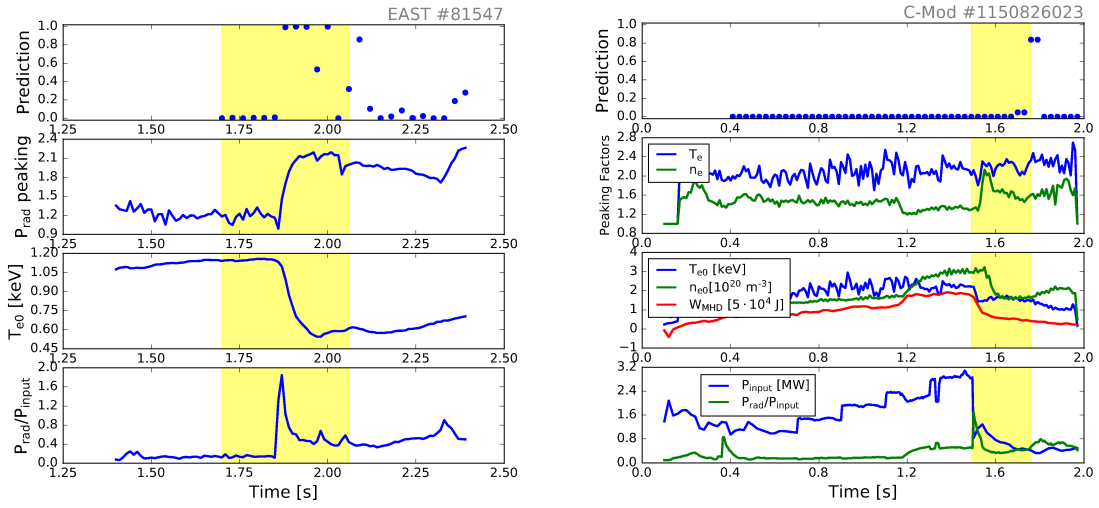
(a) (b)

Figure 11: (**a**) Performance and number of shots for each run, or experiment, in the dataset [run 36 has 40 shots without an event, off scale]; (**b**) the 11 run indices and corresponding run names with positive predictions.

led to this event in run 0, which used N_2 impurity seeding [29]. Another mechanism leading to radiative collapse is uncovered in runs 3 and 5 (highlighted in orange), in which regular ELMs that acted to flush impurities out of the core were suppressed [30]. This is connected to the event chains in runs 1 and 9 (highlighted in green), which saw a change in ELM type after deliberate power stepdowns during power threshold experiments [31]. The remaining runs 6-8 each have one shot with a radiative collapse linked to the dynamics described above. In summary, the 3 initially labeled shots were used to search for a set of regimes and dynamics that correlate with radiative collapses. This uncovered a correlation between the event and impurity accumulation, all without using any signals relating to impurities.

5.3. Extension to Other Tokamaks

In the same way that events from different experiments and run days can be detected, one may also reasonably expect to detect events on other tokamaks if the event can be described by comparable signals on multiple devices. To demonstrate that this is feasible, single shots with a radiative collapse on EAST and an H-L back transition on the Alcator C-Mod tokamak were added to the corresponding datasets from Section 3 and the same label propagation algorithm was retrained. Since each input signal used was either dimensionless or had a similar range to the DIII-D operational space, all data was standardized using the same method discussed in Section 2. Despite the different characteristics amongst the devices, the events in both discharges were successfully detected, as shown in Figure 12. Of course, a more rigorous statistical study would be needed to evaluate changes in performance when extrapolating to different tokamaks.



(a) (b)

Figure 12: Detections of a core radiative collapse on EAST (a) and an H-L back transition on Alcator C-Mod (b) are shown, using the same input signals from Table 1 for each case and the same algorithmic parameters discussed in Section 3

However, this observation motivates the potential use of label spreading to detect events on one tokamak given few initial examples of the event on another.

5.4. Discussion

In order to share this work with the larger fusion community and encourage the construction of events databases, a new module has been developed in OMFIT [32]. The module guides the user through the data collection and preprocessing steps necessary to create a usable dataset for a new event, providing MDSplus [33] support for multiple tokamaks via the SCOPE module. It then allows the user to iteratively train label spreading algorithms on that dataset and manually verify their predictions. The workflow is supported by a graphical user interface meant to guide the user through the steps outlined in this paper, and the original datasets used in this study are included so that this work can be reproduced.

Although the label spreading algorithm can in principle be applied to any arbitrary event described by a robustly available set of 0D signals, in practice it becomes intractable for an unmanageably large number of samples in the dataset. Given n sequence samples, label propagation can be shown to have a computational cost of $O(n^3 + 2n^2)$, though that cost can be significantly reduced with slight modifications to the algorithm which use bounds on the node values during each iteration to prune unnecessary computations [34]. Therefore, caution should be taken when extending this to a substantially larger set of shots, a set of shots with much longer duration, or to a set

of sequences sampled at a much higher rate. Using a lightweight kernel like the $k-NN$ kernel introduced in Figure 6(b) may also help reduce computational cost, as this will produce a sparse transition matrix (see Equation 2) relative to the more computationally expensive radial basis function kernel in Equation 3. All label spreaders in this study were trained on a set of $\approx 2 \cdot 10^4$ time sequences representing ~ 300 shots, and each training instance typically required a few minutes of wall clock time on a dual processor.

A related problem arises when the number of input signals or sequence steps is increased, known as the curse of dimensionality. Recall the algorithm's assumption that sequences with the same label lie near each other on some appropriately smooth manifold in the high-dimensional space. The manifold in question could grow in complexity with the number of dimensions used to describe it, thereby requiring more data to effectively resolve its curvature [35]. There is precedent for using this algorithm with > 8000 features for applications like text classification [21], but caution should be taken as the dimensionality of $\mathbb{R}^{N \cdot S}$ is expanded.

6. Conclusions

The label spreading algorithm has been applied to three distinct disruption precursor events: H-L back transitions, locked modes with rotating precursors, and core radiative collapses. Its results have been compared with manually verified event occurrences in a DIII-D dataset of over 300 disruptive discharges. The results show a reliable detection ability when the user starts with little initial information, as few as one example discharge. This stands in contrast with the bulk of machine-learning literature in the fusion community, which typically either rely on large labeled datasets, or do not use labeled data at all to constrain solutions.

A trend of initially increasing predictive performance has been demonstrated as more initial information is added for the algorithm to use. These observations hold for each of the events studied in this work, and are shown to apply for a robust choice of algorithmic parameters. In particular, it is shown that the scale of the kernel must be large enough to avoid overfitting to the initially labeled region, yet small enough to capture the non-linearity of the target manifold. Additionally, a harder clamping effect (small α) is generally desired, as this retains more of the initially labeled information and tends to result in better performance.

Motivated by these results, methods for exploratory analysis were introduced. These can potentially be used to build events databases from scratch and search large sets of shots for particular dynamics. It is shown that one can expect this analysis to naturally extend to other types of events and be applied to $\sim 10^3$ shots, as long as these events can be represented well in a reasonably-sized feature space. These applications can be readily implemented with a user-friendly module developed as part of this study. For future work, it can be applied to construct events databases which may contribute to understanding of disruptive event chains and the development of disruption avoidance algorithms.

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-SC0010492 and DE-FC02-04ER54698. Part of the data analysis was performed using the OMFIT integrated modeling framework [32]. DIII-D data shown in this paper can be obtained in digital format by following the links at https://fusion.gat.com/global/D3D_DMP.

Disclaimer

References

[1] Martin Y R and Takizuka T 2008 *Journal of Physics: Conference Series* **123** 012033 ISSN 17426596 URL https://doi.org/10.1088/1742-6596/123/1/012033

[2] Klevarová V, Zohm H, Pautasso G, Tardini G, Mcdermott R, Verdoolaege G, Snipes J, Vries P C and Lehnen M 2020 *Plasma Physics and Controlled Fusion* **62** 025024 ISSN 13616587 URL https://doi.org/10.1088/1361-6587/ab5c41

[3] De Vries P C, Johnson M F, Alper B, Buratti P, Hender T C, Koslowski H R and Riccardo V 2011 *Nuclear Fusion* **51** 053018 ISSN 00295515 URL https://doi.org/10.1088/0029-5515/51/5/053018

[4] Berkery J W, Sabbagh S A, Bell R E, Gerhardt S P and LeBlanc B P 2017 *Physics of Plasmas* **24** 056103 ISSN 10897674 URL https://doi.org/10.1063/1.4977464

[5] Strait E J, Barr J L, Baruzzo M, Berkery J W, Buttery R J, De Vries P C, Eidietis N W, Granetz R S, Hanson J M, Holcomb C T, Humphreys D A, Kim J H, Kolemen E, Kong M, Lanctot M J, Lehnen M, Lerche E, Logan N C, Maraschek M, Okabayashi M, Park J K, Pau A, Pautasso G, Poli F M, Rea C, Sabbagh S A, Sauter O, Schuster E, Sheikh U A, Sozzi C, Turco F, Turnbull A D, Wang Z R, Wehner W P and Zeng L 2019 *Nuclear Fusion* **59** 112012 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab15de

[6] Kaye S M, Battaglia D J, Baver D, Belova E, Berkery J W, Duarte V N, Ferraro N, Fredrickson E, Gorelenkov N, Guttenfelder W, Hao G Z, Heidbrink W, Izacard O, Kim D, Krebs I, La Haye R, Lestz J, Liu D, Morton L A, Myra J, Pfefferle D, Podesta M, Ren Y, Riquezes J, Sabbagh S A, Schneller M, Scotti F, Soukhanovskii V, Zweben S J, Ahn J W, Allain J P, Barchfeld R, Bedoya F, Bell R E, Bertelli N, Bhattacharjee A, Boyer M D, Brennan D, Canal G, Canik J, Crocker N,

Darrow D, Delgado-Aparicio L, Diallo A, Domier C, Ebrahimi F, Evans T, Fonck R, Frerichs H, Gan K, Gerhardt S, Gray T, Jarboe T, Jardin S, Jaworski M A, Kaita R, Koel B, Kolemen E, Kriete D M, Kubota S, Leblanc B P, Levinton F, Luhmann N, Lunsford R, Maingi R, Maqueda R, Menard J E, Mueller D, Myers C E, Ono M, Park J K, Perkins R, Poli F, Raman R, Reinke M, Rhodes T, Rowley C, Russell D, Schuster E, Schmitz O, Sechrest Y, Skinner C H, Smith D R, Stotzfus-Dueck T, Stratton B, Taylor G, Tritz K, Wang W, Wang Z, Waters I and Wirth B 2019 *Nuclear Fusion* **59** 112007 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab023a

[7] Piccione A, Berkery J W, Sabbagh S A and Andreopoulos Y 2020 *Nuclear Fusion* **60** 046033 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab7597

[8] Pau A, Fanni A, Cannas B, Carcangiu S, Pisano G, Sias G, Sparapani P, Baruzzo M, Murari A, Rimini F, Tsalas M and de Vries P C 2018 *IEEE Transactions on Plasma Science* **46** 2691–2698 ISSN 0093-3813 URL https://doi.org/10.1109/TPS.2018.2841394

[9] Pau A, Fanni A, Carcangiu S, Cannas B, Sias G, Murari A and Rimini F 2019 *Nuclear Fusion* **59** 106017 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab2ea9

[10] Montes K J, Rea C, Granetz R S, Tinguely R A, Eidietis N, Meneghini O M, Chen D L, Shen B, Xiao B J, Erickson K and Boyer M D 2019 *Nuclear Fusion* **59** 096015 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab1df4

[11] Rea C, Montes K J, Erickson K G, Granetz R S and Tinguely R A 2019 *Nuclear Fusion* **59** 096016 ISSN 17414326 URL https://doi.org/10.1088/1741-4326/ab28bf

[12] Fu Y, Eldon D, Erickson K, Kleijwegt K, Lupin-Jimenez L, Boyer M D, Eidietis N, Barbour N, Izacard O and Kolemen E 2020 *Physics of Plasmas* **27** 022501 ISSN 10897674 URL https://doi.org/10.1063/1.5125581

[13] Vega J, Dormido-Canto S, López J M, Murari A, Ramírez J M, Moreno R, Ruiz M, Alves D and Felton R 2013 *Fusion Engineering and Design* **88** 1228–1231 ISSN 09203796 URL https://doi.org/10.1016/j.fusengdes.2013.03.003

[14] Rattá G A, Vega J and Murari A 2018 *Fusion Science and Technology* **74** 13–22 ISSN 1536-1055 URL https://doi.org/10.1080/15361055.2017.1390390

[15] Kates-Harbeck J, Svyatkovskiy A and Tang W 2019 *Nature* **568** 526–531 ISSN 14764687 URL https://doi.org/10.1038/s41586-019-1116-4

[16] Zheng W, Hu F, Zhang M, Chen Z, Zhao X, Wang X, Shi P, Zhang X, Zhang X, Zhou Y, Wei Y and Pan Y 2018 *Nuclear Fusion* **58** 056016 ISSN 0029-5515 URL https://doi.org/10.1088/1741-4326/aaad17

[17] Windsor C, Pautasso G, Tichmann C, Buttery R, Hender T, Contributors J E and Team t A U 2005 *Nuclear Fusion* **45** 337 ISSN 0029-5515 URL https://doi.org/10.1088/0029-5515/45/5/004

[18] Zhu J, Rea C, Montes K J, Granetz R, Sweeney R and Tinguely R A 2020 *Nuclear Fusion* URL https://doi.org/10.1088/1741-4326/abc664

[19] Murari A, Vega J, Rattá G, Vagliasindi G, Johnson M and Hong S 2009 *Nuclear Fusion* **49** 055028 ISSN 0029-5515 URL https://doi.org/10.1088/0029-5515/49/5/055028

[20] Aledda R, Cannas B, Fanni A, Sias G and Pautasso G 2012 *International Journal of Applied Electromagnetics and Mechanics* **39** 43–49 ISSN 13835416 URL http://doi.org/10.3233/JAE-2012-1441

[21] Zhou D, Bousquet O, Lal T N, Weston J and Bernhard S 2004 Learning with local and global consistency *Advances in In Neural Information Processing Systems* vol 16 pp 321–328 URL https://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf

[22] Zoidi O, Fotiadou E, Nikolaidis N and Pitas I 2015 *ACM Comput. Surv.* **47** 48 ISSN 0360-0300 URL https://doi.org/10.1145/2700381

[23] Rea C, Montes K J, Pau A, Granetz R S and Sauter O 2020 *Fusion Science and Technology* URL https://doi.org/10.1080/15361055.2020.1798589

[24] Zhu X and Ghahramani Z 2002 Learning from Labeled and Unlabeled Data with Label Propagation Tech. Rep. CMU-CALD-02-107 Carnegie Mellon University Pittsburgh, PA, USA URL http://www.cs.cmu.edu/~zhuxj/pub/CMU-CALD-02-107.pdf

[25] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 *Journal of Machine Learning Research* **12** 2825–2830 ISSN 2375-0529 URL http://jmlr.org/papers/v12/pedregosa11a.html

[26] Bykov I, Chrobak C P, Abrams T, Rudakov D L, Unterberg E A, Wampler W R, Hollmann E M, Moyer R A, Boedo J A, Stahl B, Hinson E T, Yu J H, Lasnier C J, Makowski M and McLean A G 2017 *Physica Scripta* **T170** 014034 URL https://doi.org/10.1088/1402-4896/aa8e34

[27] Abrams T, Unterberg E A, Rudakov D L, Leonard A W, Schmitz O, Shiraki D, Baylor L R, Stangeby P C, Thomas D M and Wang H Q 2019 *Physics of Plasmas* **26** 062504 URL https://doi.org/10.1063/1.5089895

[28] Barton J L, Nygren R E, Unterberg E A, Watkins J G, Makowski M A, Moser A, Rudakov D L and Buchenauer D 2017 *Physica Scripta* **T170** 014007 URL https://doi.org/10.1088/1402-4896/aa878a

[29] Petrie T, Osborne T, Fenstermacher M, Ferron J, Groebner R, Grierson B, Holcomb C, Lasnier C, Leonard A, Luce T, Makowski M, Turco F, Solomon W, Victor B and Watkins J 2017 *Nuclear Fusion* **57** 086004 URL https://doi.org/10.1088/1741-4326/aa7399

[30] Petty C, Nazikian R, Park J, Turco F, Chen X, Cui L, Evans T, Ferraro N, Ferron J, Garofalo A, Grierson B, Holcomb C, Hyatt A, Kolemen E, Haye R L, Lasnier C, Logan N, Luce T, McKee G, Orlov D, Osborne T, Pace D, Paz-Soldan C, Petrie T, Snyder P, Solomon W, Taylor N, Thome K, Zeeland M V and Zhu Y 2017 *Nuclear Fusion* **57** 116057 URL https://doi.org/10.1088/1741-4326/aa80ab

[31] Yan Z, Mckee G R, Gohil P, Schmitz L, Holland C, Haskey S R, Grierson B A, Ke R, Rhodes T and Petty C 2019 *Physics of Plasmas* **26** 062507 URL https://doi.org/10.1063/1.5091701

[32] Meneghini O, Smith S, Lao L, Izacard O, Ren Q, Park J, Candy J, Wang Z, Luna C, Izzo V, Grierson B, Snyder P, Holland C, Penna J, Lu G, Raum P, McCubbin A, Orlov D, Belli E, Ferraro N, Prater R, Osborne T, Turnbull A and Staebler G 2015 *Nuclear Fusion* **55** 083008 ISSN 0029-5515 URL https://doi.org/10.1088/0029-5515/55/8/083008

[33] Fredian T, Stillerman J, Manduchi G, Rigoni A, Erickson K and Schrder T 2018 *Fusion Engineering and Design* **127** 106 – 110 ISSN 0920-3796 URL https://doi.org/10.1016/j.fusengdes.2017.12.010

[34] Fujiwara Y and Irie G 2014 Efficient label propagation *Proceedings of the 31st International Conference on Machine Learning* (*PMLR* vol 32) (Journal of Machine Learning Research) pp 784–792 ISBN 9781634393973 URL http://proceedings.mlr.press/v32/fujiwara14.pdf

[35] Delalleau O, Bengio Y and Le Roux N 2006 11 Label Propagation and Quadratic Criterion *Semi-Supervised Learning* ed Chapelle O, Scholkopf B and Zien A (MIT Press) chap 11, pp 35–58 ISBN 9780262033589