PAPER

Interpretation of machine-learning-based disruption models for plasma control

IOP Publishing Plasma Physics and Controlled Fusion
Plasma Phys. Control. Fusion **59** (2017) 085001 (5pp) https://doi.org/10.1088/1361-6587/aa72a3

Interpretation of machine-learning-based disruption models for plasma control

Matthew S Parsons

Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States of America

E-mail: parsons6@illinois.edu

Received 2 February 2017, revised 30 March 2017
Accepted for publication 11 May 2017
Published 5 June 2017

Abstract

While machine learning techniques have been applied within the context of fusion for predicting plasma disruptions in tokamaks, they are typically interpreted with a simple 'yes/no' prediction or perhaps a probability forecast. These techniques take input signals, which could be real-time signals from machine diagnostics, to make a prediction of whether a transient event will occur. A major criticism of these methods is that, due to the nature of machine learning, there is no clear correlation between the input signals and the output prediction result. Here is proposed a simple method that could be applied to any existing prediction model to determine how sensitive the state of a plasma is at any given time with respect to the input signals. This is accomplished by computing the gradient of the decision function, which effectively identifies the quickest path away from a disruption as a function of the input signals and therefore could be used in a plasma control setting to avoid them. A numerical example is provided for illustration based on a support vector machine model, and the application to real data is left as an open opportunity.

Keywords: plasma disruptions, disruption prediction, disruption avoidance, plasma control, machine learning

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

1. Introduction

In a tokamak, it is possible for the plasma to suddenly escape its confinement in an event known as a disruption [1, 2]. This sudden loss of confinement results in thermal and magnetic loads on the walls, and potentially the formation of electron currents with relativistic energies, which can seriously damage the machine. In order for the tokamak to be a viable design for a fusion power plant, it must be possible to avoid, or at least reliably mitigate, the effects of disruptions. Even before a power plant, the need for minimizing disruptions in ITER is severe [3]. Since avoidance and mitigation techniques are inherently more effective when more warning time is given for them to be enacted, a great deal of attention has been given to the prediction of disruptions.

There are a large variety of phenomena which lead up to disruptions, which makes their prediction a significant challenge. However, there is a wealth of experimental data that has been accumulated on disruptions from experiments around the world, and this data provides a place to search for answers. To help make sense of this wealth of disruption data, one can employ the tools of machine learning to search for patterns and construct models to predict the occurrence of disruptions.

A lot of promising work has been done in applying machine learning techniques to the problem of disruption prediction. This work is based on the idea that disruptive discharges must exhibit some characteristics, just before the disruption occurs, which distinguish them from non-disruptive discharges. Using diagnostic measurements taken from both disruptive and non-disruptive plasmas, a machine learning algorithm constructs a model which separates the disruptive and non-disruptive plasma conditions as clearly as possible. The model will then give a prediction, either in terms of a probability or a simple yes/no response, of whether a disruption will occur for a given set of input variables. In this decade, this has been demonstrated using Logistic Regression on AUG [4], Multilayer Perceptron Neural Networks on AUG [5, 6] and separately on J-TEXT [7], Support Vector Machines on JET [8–11], and Adaptive Venn

Predictors on JET [12]. In testing these machine-learning-based disruption models on other archived and even real-time data, prediction success rates have been demonstrated near 90%, with just a few percent of false alarms.

A major criticism of these approaches is that they fail to make a clear correlation between their inputs and outputs, as is inherently the nature of machine learning models. This criticism has continued to dissuade a more serious pursuit to develop these techniques, which are often viewed as being out-of-place in the plasma physics community. While precise scaling-law-type relationships are likely not to be recoverable from these models, it still is possible to extract some information about how the input parameters affect the output. Here is described a new way to interpret machine-learning-based disruption prediction models, and some implications of this interpretation for plasma control applications. While the focus will be on disruptions, it is worth noting that this type of analysis could be applied to any situation where different plasma states are tracked, such as ELM regime classification [13].

2. Classification model interpretation

To begin, suppose that a disruption prediction model exists which can identify whether a plasma will disrupt or not based on the input of n features describing the state of the plasma. These features, which may include quantities like the plasma current and density, combine to form a so-called feature vector $\mathbf{x} \in \mathcal{D} \subset \Re^n$. For any feature vector $\mathbf{x}$ in the domain $\mathcal{D}$, a decision function $f(\mathbf{x})$ can determine where $\mathbf{x}$ is with respect to the boundary between the two classes.

As a simple toy model, consider a system of two classes A and B (or disruptive and non-disruptive) represented by two features $x_1 \in [0, 1]$ and $x_2 \in [0, 1]$. The number of dimensions used in disruption prediction work is typically something like 14, though two will provide a more tangible illustration. Here, a decision function is produced with a support vector machine algorithm [14], which takes a set of 'training' points and pulls out the ones which best separate the two classes. The set of points forming the boundary between the two classes are called support vectors, and these are used by the decision function to determine the class of any new point. A set of Support Vectors separating classes A and B can be seen in figure 1.

During a plasma discharge, the state of the plasma will evolve through the feature space. Since the features are always normalized to represent their individual statistical ranges, the trajectory will be reasonably smooth with respect to the entire domain. To simulate this motion through the feature space, consider moving in the 2D plane with an attractive $1/\|\mathbf{x}\|^2$ force at the origin. By 'launching' different trajectories through the feature space, one can examine different scenarios of approaching the boundary. The decision function can then be exploited to better understand what is happening along the trajectory and how one might use that information to make control decisions.

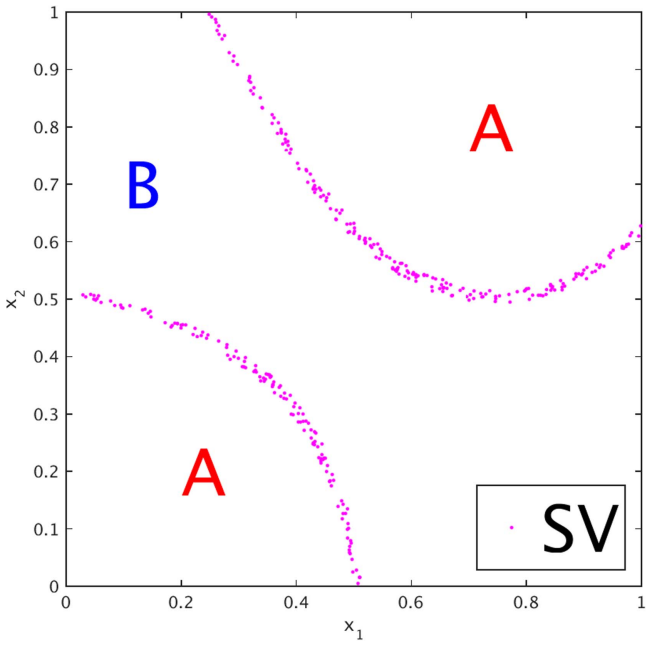

Figure 1. A set of 10 000 training points is randomly generated on a 2D grid spanning 0 to 1, and the points are assigned to one of two classes, A (red) or B (blue), based on their location. An SVM model with a gaussian kernel is trained on this data, and the points chosen as support vectors (purple) mark the noisy boundary between the two classes. In practice, the noisiness of the boundary depends entirely on the particular features chosen, and ideally the boundary could be drawn as a distinct line.

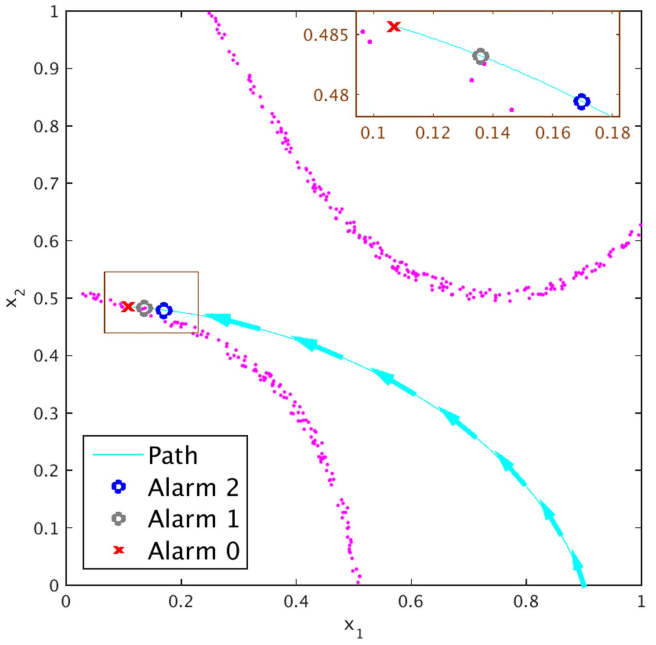

Figure 2. Trajectory 1 through feature space results in a collision with the boundary after several warning alarms.

As a first example, consider Trajectory 1 as depicted in figure 2, which collides with the boundary after triggering two warning alarms. Alarm 1 is the maximal distance of any support vector from the boundary, and Alarm 2 is twice this value. There is a significant uncertainty in the ability of the

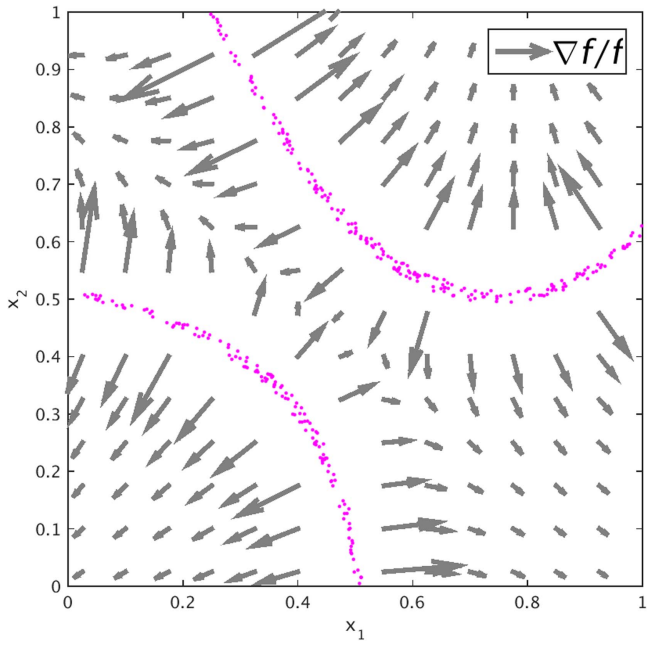

Figure 3. Normalized gradients of the decision function, calculated by first-order central differences, indicate which direction to take to avoid the boundary at any given point.

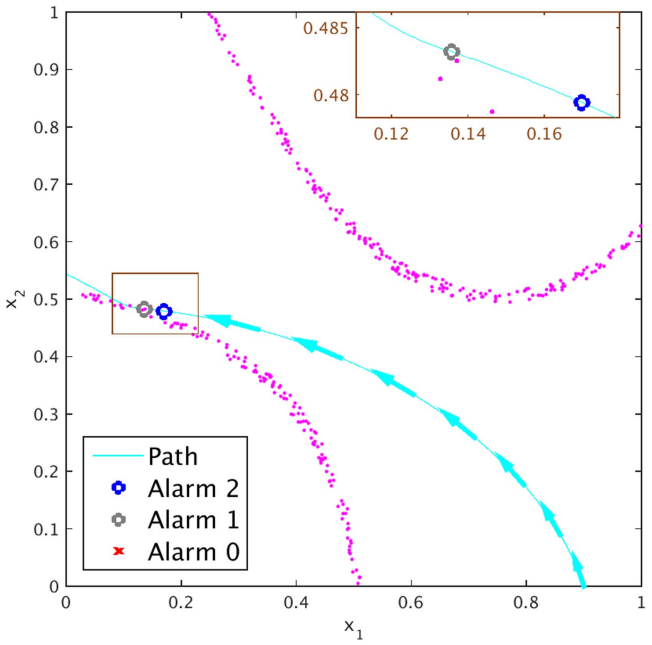

Figure 4. Trajectory 2, with identical initial conditions to Trajectory 1, avoids colliding with the boundary by adding an additional force term when it gets too close. This force term is proportional to the normalized gradient of the decision function.

decision function to correctly classify points inside the margin of support vectors, so Alarm 1 marks the edge of a grey region for prediction. If a collision with the boundary is to be avoided, this would be the last opportunity to take an action.

Everywhere along this trajectory, the decision function outputs a measurement of how far a given point is from the boundary. Whether this distance is positive or negative identifies which side of the boundary it is on, and hence which class it belongs to. Disruption prediction research typically only utilizes the class information, but there is much to be gained by considering the distance measurement itself. It is important to note that this distance is not a distance in the feature space, but is a measure of how far away the decision function perceives the boundary to be. In practice, the precise location of the boundary would not be known or a statistical model would not be needed in the first place, so it is not ambiguous to continue to refer to this simply as the distance from the boundary.

Using the decision function, normalized gradients $\nabla f(x)/f(x)$ can be computed using a first-order central difference formula, and the resulting vector field is shown in figure 3. The gradient $\nabla f(\mathbf{x})$ of course describes how sensitive $f(\mathbf{x})$ is to a change in any of the features x_n. Moreover, the $1/f(\mathbf{x})$ normalization gives a higher weighting when $\mathbf{x}$ is close to the boundary. In application, the components of this normalized gradient would describe which input features should be modified to steer the plasma away from this boundary between disruptive and non-disruptive states, weighted with a sense of urgency. If the chosen features are things which can be controlled, decomposing this gradient with respect to the combined effects of the available actuators would identify clearly what action to take to best avoid the boundary.

Trajectory 2, as seen in figure 4, demonstrates how this normalized gradient information could be used to help avoid a collision with the boundary. Here, a feedback control is simulated by adding a force term proportional to $\nabla f(\mathbf{x})/f(\mathbf{x})$ into the equations of motion when the trajectory is within the Alarm 2 distance. This allows the trajectory to be safely steered away from the boundary.

In addition to having an alarm based on the distance to the boundary, it may be useful to have an alarm based on the time to reach the boundary. The ability to navigate the trajectory away from a boundary will depend on the response time of the actuators to modify the trajectory, so an estimation of how much time they have to react would be useful. To do this, consider the time evolution of the distance from the boundary. By taking simple backwards differences, one can compute the instantaneous velocity v and acceleration a toward the boundary. These quantities, in combination with the current distance from the boundary d, provide a quadratic equation which can be solved to estimate the time to reach the boundary.

For Trajectory 1, the root $(-v + \sqrt{v^2 - 2ad})/a$ is plotted as a function of time in figure 5. Here the geometry of the problem works out so that only this root is needed to describe the system, but in general the important root is the one with the smallest magnitude. In the middle of this trajectory there is a period where the quadratic root is complex, also indicated in figure 6, meaning that the trajectory is not expected to collide with the boundary. In effect, the real root suggests how close the boundary is, the imaginary part suggests whether to expect a collision with the boundary, and the norm suggests a combined risk of colliding with the boundary.

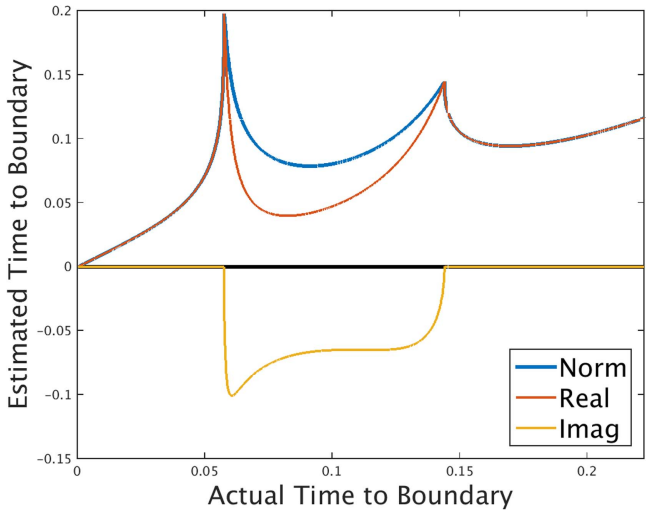

Figure 5. For Trajectory 1, a root of the quadratic equation predicting the collision with the boundary is plotted in terms of its real (orange) and imaginary (gold) components, and their norm (blue).

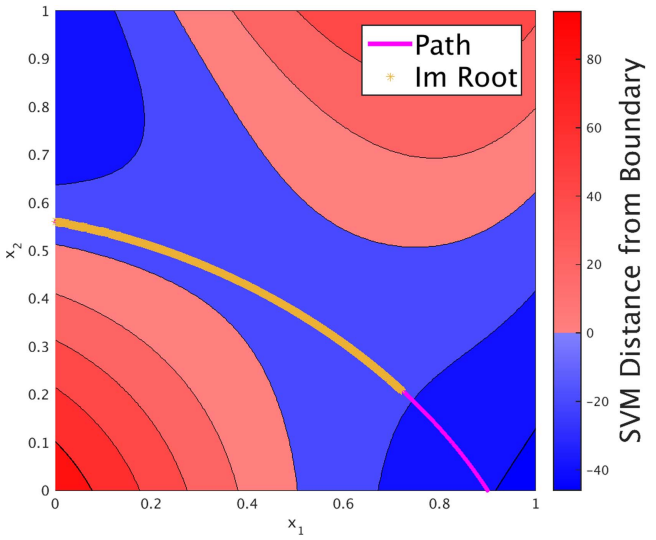

Figure 7. Trajectory 3 (magenta) is shown on top of a contour plot indicating the distance to the boundary. The presence of an imaginary component in the estimated time to collision suggests that a collision is not expected during part of the trajectory (gold).

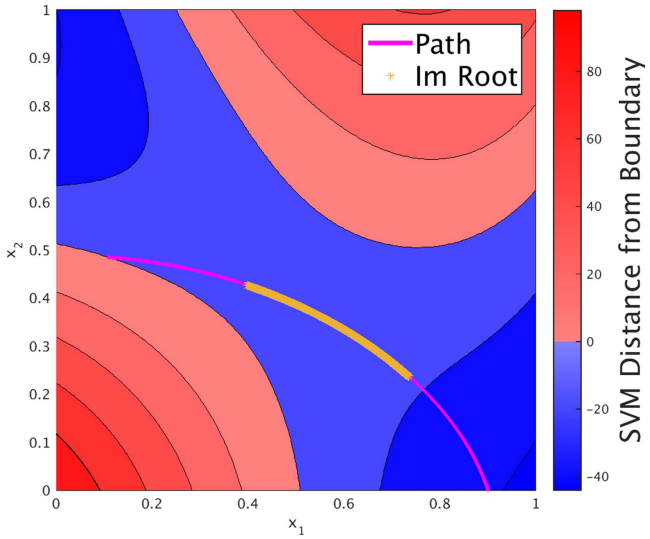

Figure 6. Trajectory 1 (magenta) is shown on top of a contour plot indicating the distance to the boundary. The presence of an imaginary component in the estimated time to collision suggests that a collision is not expected during part of the trajectory (gold).

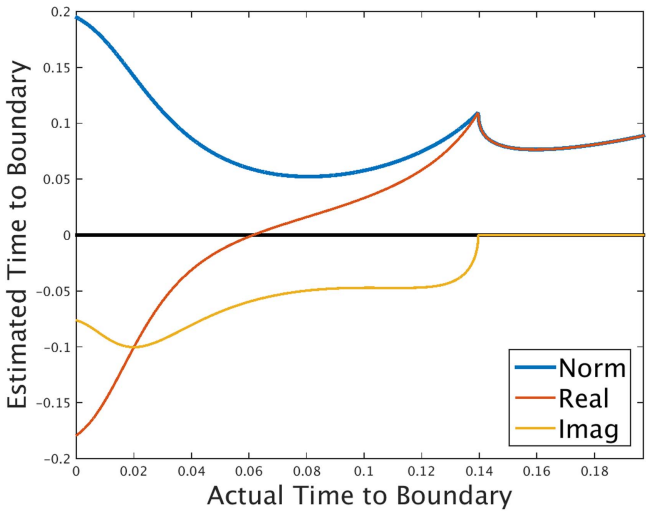

Figure 8. For Trajectory 3, a root of the quadratic equation predicting the collision with the boundary is plotted in terms of its real (orange) and imaginary (gold) components, and their norm (blue). Since this trajectory does not actually collide with the boundary, the *x*-axis represents the time until the trajectory leaves the domain where the problem is defined.

By tuning the initial velocity, Trajectory 3 is chosen to pass very close to the boundary without a collision, as shown in figure 7. An imaginary component of the collision time exists for a large portion of the trajectory, as shown in figure 8, suggesting that the trajectory is not expected to collide with the boundary. However, the real part of the estimated time decreases toward zero as the trajectory gets closer to the boundary. Although it is not expected that the trajectory would meet the boundary, it is clear that a slight, unexpected alteration could quickly send it there. As such, in making control decisions it would be prudent to compare the time given by the real root to the response time of the machine actuators.

3. Discussion

If machine learning can be used to produce a model which accurately predicts disruptions, such as is attempted in the works described previously, those models could be deployed in a way that actually makes it possible to avoid disruptions rather than simply mitigate them. Instead of just accepting a probability forecast or simple yes/no prediction, these models can be exploited for more information. By tracking the decision function over time and taking simple backward differences to determine the instantaneous velocity and acceleration toward the disruption boundary, one can calculate an estimated time to disruption. By solving the quadratic

equation to calculate this disruption time, one can get a sense of whether the plasma is headed toward a disruption, how imminent a disruption is, and how far away a disruption is waiting if the discharge changes its course.

This kind of disruption time estimation is invaluable when it comes to making a decision about whether to fire a disruption mitigation system or to take an avoidance action. If the estimated time to disruption is less than the response time of any control actuators, clearly the mitigation system must be used immediately. A decision to trigger a mitigation system might also be made if the distance to the disruption is too small, regardless of the time. However, if the plasma is not so close to the disruption and there is enough time to take an avoidance action, the proper one can be determined by looking at the normalized gradient of the decision function. These normalized gradients describe, weighted with a sense of urgency, which parameters of the plasma should be modified to avoid the disruption. If actuators are in place to modify these parameters, they could potentially be used to avoid the disruption.

Previous work on using an SVM model to monitor a discharge's proximity to disruption found that, for JET data, a non-disruptive plasma maintains a trajectory parallel to the disruption boundary [15]. This suggests that an approach like the one described here to interpret the prediction outputs may indeed be practical in a plasma control setting, since there is a high confidence that if the plasma is on a trajectory toward the boundary it truly is headed toward a disruption. Further work needs to be done to identify the timescales during which the plasma evolves from this non-disruptive trajectory to one approaching the disruption boundary. If there is sufficient warning time for avoidance actions to be taken, the gradient interpretation of the prediction model could be a powerful tool for plasma control.

Acknowledgments

This work, conducted at ITER, is supported by a grant from the 2016–2017 Fulbright US Student Program, administered by the Franco-American Fulbright Commission in France. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

ORCID

Matthew S Parsons https://orcid.org/0000-0002-2428-3215

References

[1] Schuller F C 1995 Disruptions in tokamaks *Plasma Phys. Control. Fusion* **37** A135
[2] Hender T C *et al* 2007 Mhd stability, operational limits and disruptions *Nucl. Fusion* **47** S128
[3] de Vries P C, Pautasso G, Humphreys D, Lehnen M, Maruyama S, Snipes J A, Vergara A and Zabeo L 2016 Requirements for triggering the iter disruption mitigation system *Fusion Sci. Technol.* **69** 471–84
[4] Aledda R *et al* 2015 Improvements in disruption prediction at asdex upgrade *Fusion Eng. Des.* **96** 698–702
[5] Cannas B *et al* 2011 Disruption prediction with adaptive neural networks for asdex upgrade *Fusion Eng. Des.* **86** 1039–44
[6] Cannas B, Fanni A, Pautasso G, Sias G and Sonato P 2010 An adaptive real-time disruption predictor for asdex upgrade *Nucl. Fusion* **50** 075004
[7] Wang S Y, Chen Z Y, Huang D W, Tong R H, Yan W, Wei Y N, Ma T K, Zhang M and Zhuang G 2016 Prediction of density limit disruptions on the j-text tokamak *Plasma Phys. Control. Fusion* **58** 055014
[8] Moreno R, Vega J, Dormido-Canto S, Pereira A, Murari A and (JET Contributors) 2016 Disruption prediction on jet during the ilw experimental campaigns *Fusion Sci. Technol.* **69** 485–94
[9] Vega J *et al* (JET-EFDA Contributors) 2013 Results of the jet real-time disruption predictor in the iter-like wall campaigns *Fusion Eng. Des.* **88** 1228–31
[10] Dormido-Canto S *et al* (JET-EFDA Contributors) 2013 Development of an efficient real-time disruption predictor from scratch on jet and implications for iter *Nucl. Fusion* **53** 113001
[11] Odstrcil M, Murari A and Mlynar J 2013 Comparison of advanced machine learning tools for disruption prediction and disruption studies *IEEE Trans. Plasma Sci.* **41** 1751–9
[12] Vega J *et al* (JET-EFDA Contributors) 2014 Adaptive high learning rate probabilistic disruption predictors from scratch for the next generation of tokamaks *Nucl. Fusion* **54** 123001
[13] Shabbir A, Verdoolaege G, Vega J and Murari A 2015 Elm regime classification by conformal prediction on an information manifold *IEEE Trans. Plasma Sci.* **43** 4190–9
[14] Press W H 2007 *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge: Cambridge University Press)
[15] Vega J, Moreno R, Pereira A, Murari A, Dormido-Canto S and (JET Contributors) 2015 Investigation of plasma dynamics to detect the approach to the disruption boundaries *1st IAEA TM on Fusion Data Processing, Validation and Analysis (Nice, France, 1st–3rd, June 2015)*