Disruption Prediction by Support Vector Machine and Neural Network with Exhaustive Search*)

Tatsuya YOKOYAMA, Takamitsu SUEYOSHI, Yuya MIYOSHI[1)], Ryoji HIWATARI[1)], Yasuhiko IGARASHI[2)], Masato OKADA and Yuichi OGAWA

Graduate School of Frontier Science, The University of Tokyo, Kashiwa 277-8561, Japan
[1)]*Rokkasho Fusion Institute, QST, Rokkasho 039-3212, Japan*
[2)]*Japan Science and Technology Agency, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan*

(Received 27 December 2017 / Accepted 19 February 2018)

A disruption is an event in which the plasma current suddenly shuts down in a tokamak reactor. Establishing methods to predict, mitigate, and avoid disruptions may be indispensable for realizing a tokamak reactor. In the present study, we have used the large dataset of high-beta experiments at JT-60U to develop a method for predicting the occurrence of disruptions. The method is based on sparse modeling that exploits the inherent sparseness common to all high-dimensional data, and it enables us to extract the maximum amount of information from the data efficiently. To carry out the sparse modeling, we have used exhaustive searches with a support vector machine and a neural network. In this research, we repeated the training and evaluation of the predictor while changing the combination of plasma parameters. As a result of the exhaustive search, we found $|B_r^{n=1}|$ and $\mathrm{d}|B_r^{n=1}|/\mathrm{d}t$ to be the dominant parameters for disruption predictions. This is not surprising, because MHD instabilities are considered to be the direct triggers of disruption. In addition, we have succeeded in identifying several important parameters that may also be strongly related to disruptions, i.e., β_N, β_P, q_{95}, δ, f_{GW}, and f_{rad}.

Keywords: tokamak, disruption, data-driven science, machine learning, sparse modeling, ES-K

DOI: 10.1585/pfr.13.3405021

1. Introduction

Plasma disruption is a critical phenomenon in a tokamak fusion reactor. To develop an operational nuclear fusion reactor, it is necessary to understand and control this phenomenon. Establishing methods to predict, mitigate, and avoid disruptions may thus be indispensable for realizing a tokamak fusion reactor, and for this reason continuous studies of these methods have been carried out [1, 2]. Nowadays, since the physical mechanism that causes a disruption has not yet been clearly identified, some studies are seeking to predict the occurrence of disruptions on the basis of experimental data. These are a form of "data-driven science", meaning that their goal is to extract scientific knowledge from a large amount of data. Data-driven science has been attracting attention in recent years, and machine learning is often used to assist it.

At the Joint European Torus (JET), a real-time disruption predictor has been constructed by Rattá *et al.* [3]. This predictor system, called APODIS, predicts disruptions using a support vector machine (SVM). Another research group has recently developed a predictor based on the automatic recognition of changes (called "anomaly detections") in data streams, without training a model using a huge amount of data [4]. In Japan, Yoshino has used a neural network (NN) to predict disruptions based on experimental data from the Japan Torus-60 Upgrade (JT-60U) [5]. The variables used for these predictions have either been selected by genetic algorithms [6] or directly by the researchers.

In the present study, we have used the large amount of experimental data collected at JT-60U, to construct a method for predicting the occurrences of disruptions based on "sparse modeling". This approach exploits the inherent sparseness common to all high-dimensional data and enables us to extract the maximum amount of information from the data efficiently. For the sparse modeling, we have used exhaustive searches with the SVM and a deep neural network (DNN), assuming that the optimal combination of explanatory variables is K-sparse [7]. We denoted these methods as "ES-K-SVM" and "ES-K-DNN", respectively. After extracting the explanatory variables, we have compared the result between the two methods.

This article is organized as follows. In Sec. 2, we describe the construction of a dataset to train and test the models. The methods of machine learning and sparse modeling, including the ES-K method, are described in Sec. 3. The results from ES-K are described in Sec. 4, where we discuss and compare the results from both SVM and DNN. The conclusions drawn from this study are presented in Sec. 5.

author's e-mail: yokoyama.tatsuya17@ae.k.u-tokyo.ac.jp
*) This article is based on the presentation at the 26th International Toki Conference (ITC26).

2. Dataset

We used data from high-beta plasma experiments at JT-60U to create the dataset that we used to train and test the machine learning models. First, we divided the experimental shots into two cases, that is, non-disruptive and disruptive cases, respectively. The non-disruptive cases include non-disruptive shots that lasted more than one second before plasma disruption. The disruptive cases include shots in which the plasma current shut down clearly and quickly. We used 69 non-disruptive cases and 54 disruptive cases.

Secondly, we established the base time for each shot. For the non-disruptive cases, we chose the base time to be the moment when the normalized beta is the highest while for the disruptive cases, we selected the moment when the current quench started. For each shot, we obtained diagnostic data and results from equilibrium calculations. We selected the 17 plasma parameters listed in Table 1 as potential variables for predicting disruptions. We used the values of these plasma parameters at multiple 5 ms intervals before the base time. The time range of the data used for training and testing the models are listed in Table 2. The distributions of the data for the eight main plasma parameters, i.e., β_N, l_i, q_{95}, δ, f_{GW}, f_{rad}, $|B_r^{n=1}|$, and $d|B_r^{n=1}|/dt$ are shown in Fig. 1.

Table 1 List of the plasma parameters obtained from each shot.

Name of parameter	Expression
Plasma current [MA]	I_p
Normalized beta	β_N
Poloidal beta	β_P
Plasma internal inductance	l_i
Safety factor at 95% of minor radius	q_{95}
Plasma triangularity	δ
Plasma elongation	κ
Mode lock amplitude ($n = 1$) [mT]	$\lvert B_r^{n=1}\rvert$
The ratio of the plasma density to the Greenwald density limit	f_{GW}
The ratio of the radiated power to the total input power	f_{rad}
Normalized beta time derivative	$d\beta_N/dt$
Poloidal beta time derivative	$d\beta_P/dt$
Plasma internal inductance time derivative	dl_i/dt
Safety factor at 95% of minor radius time derivative	dq_{95}/dt
Plasma elongation time derivative	$d\kappa/dt$
Mode lock amplitude ($n = 1$) time derivative	$d\lvert B_r^{n=1}\rvert/dt$
The ratio of the plasma density to the Greenwald density limit time derivative	df_{GW}/dt

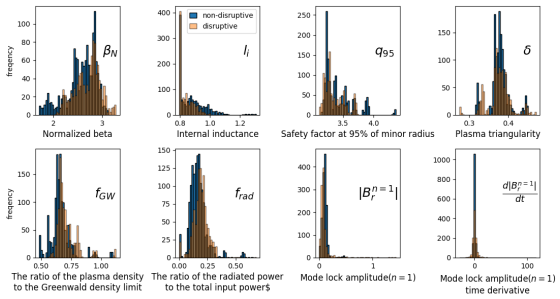

Fig. 1 Histograms of the primary plasma parameters. For each non-disruptive (blue bars) and disruptive (red bars) case, the distribution of data included in the time range used for training is shown.

3. Method

3.1 Method of machine learning

3.1.1 Support vector machine

A support vector machine (SVM) is a supervised machine learning technique [8], where the term "supervised" means that these tools work with sets of labeled objects. In the present research, we used SVM as a basic 2-class classifier, which means that it is used to classify events as non-disruptive or disruptive. The conceptual basis of the SVM is to find a hyperplane that divides data distributed in a multidimensional space into labeled sub-spaces. The coefficients of the equation representing the hyperplane are selected to maximize the distance between the data closest to the hyperplane and the hyperplane in the multidimensional space. The distance between the hyperplane and the closest data is called the "margin".

3.1.2 Neural network

A neural network (NN) is a network consisting of units that imitate neurons [9]. Each unit has multiple inputs and one output. As a 2-class classifier, we used a simple network called a "feedforward neural network" (FNN) in which the units in a given layer are connected only to the units in the adjacent layers. In an FNN, the information propagates only in one direction, from the input to the output layer. Layers between the input and the output layers are called "hidden layers". We used a network which has 2 hidden layers consisting of 64 units each. An FNN with multiple hidden layers is called a "deep neural network" (DNN).

The total input received by a unit u_j is the sum of the outputs x_i of the previous layer multiplied by different weights w_{ji} and with added a bias value b_j. The output z_j of the unit is a function f of the total input. In summary, these relations can be written as follows.

Table 2 The range of times before the base time used in training and testing the models.

Case	For training [ms]	For testing [ms]
non-disruptive	5 – 100	5 – 100
disruptive	30 – 125	5 – 125

$$u_j = \sum_{i=1}^{I} w_{ji} x_i + b_j,\ z_j = f(u_j). \quad (1)$$

The weights and biases are given random initial values, and they are repetitively modified through the training process of the NN.

3.2 Evaluation of disruption predictor

A disruption predictor is required to predict disruptions as rapidly and accurately as possible and not to issue an alarm accidentally if there is no disruption. To measure the performance of a predictor, we defined the prediction success rate and the false alarm rate are defined in this paper as follows.

Prediction Success Rate (PSR)

$$= \frac{\textit{Number of shots correctly judged as disruptive}}{\textit{Total number of disruptive shots}}, \quad (2)$$

False Alarm Rate (FAR)

$$= \frac{\textit{Number of shots incorrectly judged as disruptive}}{\textit{Total number of non-disruptive shots}}. \quad (3)$$

To avoid overfitting, meaning that the model matches the training data well, but it does not match the unknown data, we used a "k-fold cross validation" method. In this method, the data shots are divided into k pieces, of which $(k-1)$ are used to train the model and the remaining one is used to test it. The results obtained for all k patterns are averaged to provide the final result.

We tested the trained model as follows. For each test shot, the determinations were made from the data in order from the earliest time until the shot is judged to be disruptive. For each PSR and FAR, the model is evaluated based on cumulative results between the moment of evaluation and the start time of the dataset.

3.3 Realizing sparse modeling, ES-K

In an "exhaustive search" (ES) method, the models are trained and evaluated for all possible combinations of variables to obtain an optimal combination of variables. The reason for searching exhaustively is because we expect the information to be included in the combination of variables. In an ES method, all $2^N - 1 = {}_N C_1 + {}_N C_2 + \cdots + {}_N C_N$ states, which are combinations of the N variables, are exhaustively searched.

However, there is the risk of a "combination explosion" in an ES method. To overcome this problem, we used a "K-sparse exhaustive search" (ES-K) method [7]. The ES-K method is based on the assumption that the optimal combination of explanatory variables is K-sparse, i.e., that K of the N components are explanatory variables. In this research, the explanatory variables correspond to the plasma parameters shown in Table 1. We also expect that we can extract the structure presented in the optimal combination of variables by examining the results in order from small K up to N using the ES-K method. We denoted the application of ES-K to SVM or DNN as "ES-K-SVM" or "ES-K-DNN", respectively.

4. Results and Discussions

We conducted ES-K-SVM and ES-K-DNN searches for values of K from 1 to 4 using the dataset described in Sec. 2. Figure 2 shows the results of (a) ES-4-SVM and (b) ES-4-DNN at 30 ms prior to a disruption. In order to compare the performance of the two predictors, we determined that the smaller the distance from the upper left in the 2D histogram [Figs. 2 (a1) and (b1)], the better the performance. The 2D histogram has the PSR on the vertical axis and FAR on the horizontal axis, so the distance is calculated as follows.

$$distance = \sqrt{(100 - PSR)^2 + FAR^2}. \quad (4)$$

Combinations that show high performance in ES-4 are shown in Figs. 2 (a2) and (b2). These indicator diagrams show the top 20 combinations of parameters. Figures 2 (a3) and (b3) shows the histograms of variables included in these combinations that provide higher predictive performance than the performance of using all variables or that is in the top 10%.

From these diagrams and histograms, we find that certain variables are included in many combinations that show good predictive performance. The dotted lines in the variable histograms represent the case for which each variable appears equally in the combination of objects. The maximum value of the x axis corresponds to the number of possible combinations of objects. For the SVM, we find that $\mathrm{d}|B_r^{n=1}|/\mathrm{d}t$ is the most frequently cited variable. In addition, $|B_r^{n=1}|$, f_{GW}, and f_{rad} occur more frequently than in the case where all variables are assumed to be equivalent. Also, the indicator diagram shows that almost all combinations displaying the highest performance contain both $|B_r^{n=1}|$ and $\mathrm{d}|B_r^{n=1}|/\mathrm{d}t$. In contrast, for DNN, $|B_r^{n=1}|$ is the most fre-

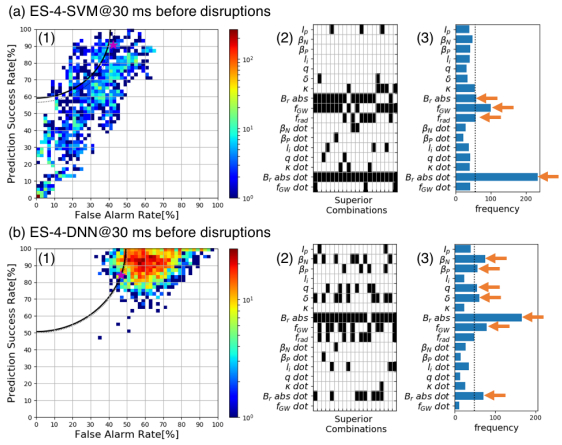

Fig. 2 (1) 2D histogram, (2) indicator diagram and (3) variable histogram showing the results of (a) ES-4-SVM and (b) ES-4-DNN. The arrows in the variable histograms indicate the variables that appear more frequently than when assuming that all variables appear equally.

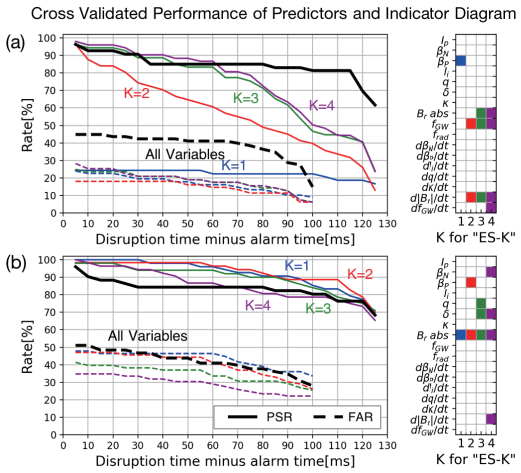

Fig. 3 Cross validated prediction success rate (solid lines) and false alarm rate (dashed lines) for the combinations that show the best performance for each K vs. time prior to disruption, together with the correspond in indicator diagrams. The figures (a) and (b) show the results from ES-K-SVM and ES-K-DNN, respectively.

quent variable while β_N, β_P, q_{95}, δ, f_{GW}, and $d|B_r^{n=1}|/dt$ occur more frequently than in the case where all variables are assumed to be equivalent. The parameters $|B_r^{n=1}|$, and $d|B_r^{n=1}|/dt$ seem thus to be central variables for predicting disruptions, because they appeared frequently in both the SVM and NN searches. Because MHD instabilities are considered to be the direct causes of disruptions [1], the result obtained with $|B_r^{n=1}|$ and its time derivative, selected by both ES-K-SVM and ES-K-DNN, seems entirely reasonable physically. On the other hand, several other variables, such as β_N, β_P, q_{95}, δ, f_{GW}, and f_{rad}, appear relatively frequently as well. These variables are known to be variables related to disruption, and they seem to be extracted as factors that affect disruptions indirectly. From this result, we expect not only that there will be limiting values for individual variables prior to disruptions, but also that there will be limiting values for regions that span multiple variables. These variables may therefore be useful for identifying possible parameter areas that lead to disruption. On the other hand, the plasma internal inductance l_i was extracted neither by ES-K-SVM nor ES-K-DNN. We have no clear explanation for this result. The histogram of l_i in Fig. 1 shows that there are no major differences between the two distributions but the distribution of disruptive case slightly concentrate on a certain value.

For those combinations that showed the highest predictive performance for each K at 30 ms prior to disruption, the time change of the results is shown in Fig. 3. For ES-K-SVM, the PSR improves as K increases, but there is no significant change in the FAR from $K = 2$ to 4. However, around 100 ms, the PSR for $K = 1$ to 4 are inferior to the PSR obtained from all the variables. On the other hand, the FAR improves as K increases in ES-K-DNN while the PSR is generally higher than that with all variables. The PSR for $K = 4$ seems to be inferior to others before 40 ms. Figure 4

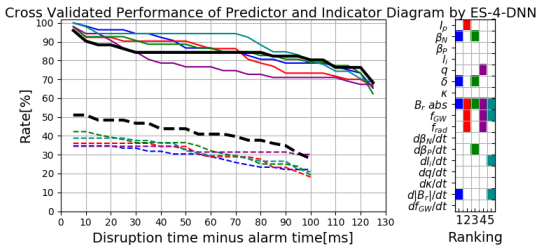

Fig. 4 Cross validated prediction success rate (solid lines) and false alarm rate (dashed lines) for the combinations showing the best performance from ES-4-DNN at 30 ms prior to the disruption vs. time prior to disruption together with the corresponding indicator diagram.

shows that some combinations have higher values for PSR and slightly lower values for FAR before 40 ms. These facts indicate that the extraction of variables is important for improving the performance of disruption predictors.

5. Conclusions

We have constructed disruption predictors using the SVM and the DNN based on experimental data from JT-60U, and we have extracted explanatory plasma parameters using the ES-K method. We have shown that the variables related to disruption can be extracted using the concept of "sparse modeling". The parameters $|B_r^{n=1}|$ and $d|B_r^{n=1}|/dt$ were selected as the most frequent variables by both the SVM and DNN, and these plasma parameters are considered as the direct causes of disruption. The parameters β_N, β_P, q_{95}, δ, f_{GW}, and f_{rad} were also found to be important factors that affect disruption. This result indicates that a parameter space can be constructed using these plasma parameters to define the conditions in which a plasma may be in danger of disruption.

Acknowledgments

We acknowledge Drs. N. Oyama, G. Matsunaga, S. Ide, and Y. Kamada in QST for fruitful discussion on disruption phenomenon, in addition to the preparation of JT-60U experimental data. This work was supported by JSPS KAKENHI Grant Number JP 17H03508.

[1] T.C. Hender *et al.*, Nucl. Fusion **47**, S128 (2007).
[2] A.H. Boozer, Phys. Plasmas **19**, 058101 (2012).
[3] G.A. Rattá, J. Vega, A. Murari *et al.*, Nucl. Fusion **50**, 025005 (2010).
[4] J. Vega, R. Moreno, A. Pereira *et al.*, In 1st EPS Conf. on Plasma Diagnostics, 2015.
[5] R. Yoshino, Nucl. Fusion **45**, 1232 (2005).
[6] G.A. Rattá, J. Vega and A. Murari, Fusion Eng. Des. **87**, 1670 (2012).
[7] Y. Igarashi, H. Takenaka, Y. Nakanishi-Ohno *et al.*, IEICE Technical Report, 116 (300): 313-320, 2016.
[8] I. Takeuchi and M. Karasuyama, *Sapoto Bekutoru Mashin (Support Vector Machine)*, (Kodansha, 2015).
[9] T. Okatani, *Shinsogakushu (Deep Learning)*, (Kodansha, 2015).