AUG-JET cross-tokamak disruption predictor

G.A. Ratta¹, J. Vega¹, A. Murari², the EUROfusion MST1+ Team and JET Contributors*

EUROfusion Consortium, JET, Culham Science Centre, Abingdon, OX14 3DB, UK
¹Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain
²Consorzio RFX, Padua, Italy.

*See the author list of “Overview of progress in European Medium Sized Tokamaks towards an integrated plasma-edge/wall solution” by H. Meyer et al., to be published in Nuclear Fusion Special issue: Overview and Summary Reports from the 26th Fusion Energy Conference (Kyoto, Japan, 17-22 October 2016).

*See the author list of “Overview of the JET results in support to ITER” by X. Litaudon et al. to be published in Nuclear Fusion Special issue: overview and summary reports from the 26th Fusion Energy Conference (Kyoto, Japan, 17-22 October 2016)
Outline

• Introduction.

• The cross-tokamak idea for ML-based disruption predictors.
  • The low detection rates (2005) discontinued the research line.

• In this work the concept has been retaken using other ML tools and guidelines.
  • Database.
  • Support Vector Machines (SVM).
  • Genetic Algorithms (GAs).

• Results.

• Summary, conclusions and future work.
Introduction

Disruptions occur suddenly and up to now their appearance is inevitable.

• However, their harmful effects can be alleviated by starting mitigation (or even avoidance) actions.
  • For that, it is fundamental to predict them long enough in advance to provide time for activating these protection systems.
  • Therefore, the predictors not only need high accuracies but also the detection rates should consider the times required to intervene.

• Disruption predictors, before the inclusion of machine learning techniques, were mainly focused on setting a threshold in the locked mode amplitude signal, attaining poor detection rates (less than 65% with at least 10 ms of anticipation at JET*).

*G.A. Rattá et al. An advanced disruption predictor for JET tested in a simulated real-time environment. 2010 Nucl. Fusion 50 025005
In sights of ITER, more accurate systems are required.

- One way to obtain them is to develop models that also consider other plasma parameters.
  - This is not a simple task: the relationship among these parameters to predict disruptions is highly nonlinear and difficult to model.
    - Even more, it is not clear which is the optimal set of plasma measurements to include in the models!

Machine learning systems* can model accurate predictors.

- However, they “learn” from past experiments: they need an already stored DB.
  - This is a problem in sights of ITER (no database will be available at the beginning of its operation to create a machine learning-based DP).

The cross-tokamak approach

7 dimensionless parameters*

1. The edge safety factor, $P_e = \frac{\rho}{\rho_0}$. This is the value of the safety factor at the 95% flux surface. A low value ($\lesssim 3$) is indicative of disruptive behaviour.

2. Internal inductance, $P_i = \frac{L_i}{L_0}$. This is the ratio of the average value of the magnetic field over the average value of the field at the plasma edge. It is therefore also dimensionless. A high value of the internal inductance is known to increase the likelihood of disruption [2].

3. The normalized toroidal beta, $P_B = \frac{\beta_T}{\beta_T[\text{FP}]}$, with $\beta_T = \frac{2B_0^2}{\mu_0 R}$. $B_0$ is the magnetic field at the plasma edge and $\beta_T$ is the normalized toroidal beta. The predicted Sykes–Treyer $\beta$-limit [1].

4. The Greenwald density limit fraction, $P_n = \frac{n_n}{n_{\text{cr}}(\text{magnetic})}$. Here $n_n$ is the measured line average density, and $n_{\text{cr}}(\text{magnetic})$ is a generalized Greenwald density including the elongation $\kappa$. A high value of $P_n$ ($\gtrsim 1$) increases the probability of a disruption via nonturbulent collapse.

5. The radiated power fraction, $P_R = \frac{P_{\text{rad}}}{P_{\text{proc}}}$. Here $P_{\text{rad}}$ is the radiated power from the plasma determined from bolometer measurements, and $P_{\text{proc}}$ is the input power from external and ohmic heating. A high radiated power fraction ($\gtrsim 100\%$) is characteristic of disruptive activity.

6. The normalized energy confinement time, $P_\tau = \frac{\tau_e}{\tau_{\text{disrupt}}}$, The confinement time is calculated from the stored energy, divided by the total energy, less the time derivative of the energy. The confinement time is normalized according to the average value over the data set, and so this is not an absolute. A low normalized confinement time is suggestive of a disruptive plasma.

7. The normalized locked mode indicator, $P_{\text{lock}} = \frac{\ell_{\text{lock}}}{\ell_{max}}$. The signal is normalized to a maximum value of unity so that it switches rapidly from zero (no locked modes) to unity (locked mode present).

The parameters used here are based on quantities that are known to play a role in tokamak plasma disruptivity. They are based on the ones used in previous machine learning-based DP.

The cross-tokamak approach

7 dimensionless parameters

(1) The edge safety factor, \( p_e = q_w \). This is the value of the safety factor at the 95% flux surface. A low value \((\leq 3)\) is indicative of disruptive behavior.

(2) Internal inductance, \( I_0 = \frac{L}{\psi_0} \). This is the ratio of the average value of the magnetic field over the average value of the field at the plasma edge. It therefore is also dimensionless. A high value of the internal inductance is known to increase the likelihood of disruption [2].

(3) The normalized toroidal beta, \( B_0 = \frac{\beta_T}{\frac{1}{2} \int_0^L \psi_0 |B_0|^2 dr} \), with \( \beta_T = \frac{1}{2} \int_0^L \psi_0 |B_0|^2 dr \) and the normalization depending on the plasma current \( I_p \), the minor radius \( a \) and the toroidal field \( B_T \) arising from the predicted Sykes–Trinity \( E \)-limit [1]. Sawtoothing ELMy H-mode plasmas tend to become unstable if the parameter \( B_0 \) rises above about 3.5.

(4) The Greenwald density limit fraction, \( n_0 = n_{\text{cr}} = \frac{n_{\text{crit}}}{n_{\text{cr}(\text{tens})}} \). Here \( n_{\text{crit}} \) is the measured line average density, and \( n_{\text{cr}(\text{tens})} \) is a generalized Greenwald density including the elongation \( \kappa \). A high value of \( n_0 \) (\( \gtrsim 1 \)) increases the probability of a disruption via radiative collapse.

(5) The radiated power fraction, \( P_L = \frac{P_{\text{rad}}}{P_{\text{in}}} = \frac{P_{\text{rad}}}{P_{\text{in}}} \). Here \( P_{\text{rad}} \) is the radiated power from the plasma determined from bolometer measurements, \( P_{\text{in}} \) is the input power from external and ohmic heating. A high radiated power fraction \( \gtrsim 0.10 \) is characteristic of disruptive activity.

(6) The normalized energy confinement time, \( \tau_e = \frac{\tau_{\text{in}}} {\tau_{\text{in}}(\text{tens})} \). The confinement time is calculated from the stored energy, divided by the total energy, less the time derivative of the energy. The confinement time is normalized according to the average value over the data set, and so this is not an \( E \) factor. A low normalized confinement time is suggestive of a disruptive plasma.

(7) The normalized locked mode indicator, \( P_L = L_w \). The signal is normalized to a maximum value of unity so that it switches rapidly from zero (no locked mode) to unity (locked mode present).
The cross-tokamak approach
The new cross-tokamak predictor.

We have set new guidelines:

• Use different ML techniques (SVM instead of Neural Networks) to create the predictor.
  • Optimizing, with Genetic Algorithms, the selection of the input set of signals, signal features and internal SVM parameters of the predictor.

• Target a simpler and more pragmatic solution:
  • From smaller to larger tokamak (in sights of ITER).
  • Realistic testing in simulated real-time.
    • Wide testing database.
The first task was to gather a wide enough database from JET and AUG, taking care of validating the data disruption times.

1) the poloidal beta; 2) the line integrated plasma density; 3) the plasma elongation; 4) the plasma volume divide by the device minor radius; 5) the plasma current; 6) the plasma internal inductance; 7) the locked mode amplitude; 8) the plasma vertical centroid position; 9) the total input power; 10) the safety factor at 95%; 11) the total radiated power and 12) the time derivative of the stored diamagnetic energy.

**Signal features: std(FFT(last 32ms of the signal))**

Intentional disruptions and discharges with incomplete or unreliable measurements were omitted from the databases.

**AUG**
177 disruptive shots and 1391 non-disruptive shots (2012 and 2014).

Training/Validation (chronologically first 100 of disruptives and 1000 non-disruptives).

Testing
468 shots (77 dis/391 no dis).

**JET**
All the validated POG unintentional disruptions (699) and 4158 non-disruptive discharges (April 2012 to November 2016).

**TOTAL 4857**
Support Vector Machines.

To train the system, example feature vectors are provided.

Optimized solution

\[ w = \sum_{i=1}^{n} \alpha_i y_i x_i \]

\( n = \text{number of support vectors} \)

Non linearly separable problem

Higher dimensional feature space

Kernel

Support Vectors

Support Vector Machines.

2D example

\[ \overline{x}_i(a, b) \]

Training

\[ y_i \]

\[ 1 \]

\[ -1 \]
Support Vector Machines.

Validation/test

\[ x_i(a, b) \]

new

Trained system

\[ y \]

Belongs to class

• Extra information can be deduced with this classification method:

  • The distance between the objects and the hyper-plane can be calculated.
  • It can be considered, somehow, as a measure of the classification goodness.
  • If the distance between the new tested object and the hyper-plane is large, that means that the classification is more reliable.
To attain better prediction rates, the best combination of several parameters needs to be found.

- Signals and signal features to include in the SVM leaning system.

- Trade-off between precision (risk of overfitting) and flexibility (risk of imprecise learning) --> a parameters $C$ is should be adjusted (SVM).

- Solutions are normally highly non-linear. Which one of them is better? A gamma parameter must be tuned (SVM).

To attain the best combination, we could try an exhaustive analysis.

- $10$ $C$ values.
- $10$ gamma values.
- $12$ signals.
- $12$ signal features
- $(\text{std(FFT(last 32 ms ))})$.

\[ \sum_{i=1}^{n} C_i^n \]

\[ C_i^n = \frac{n!}{(n-i)!i!} \]

where

$n=$number of values $= 44$

$i=$possible groupings $= 1, 2, ..., 44$

\(~5\text{ seconds per training/test}\)

\(~55754\text{ years!}\)

\(~1\text{ day in a PC with Genetic Algorithms}\)
Genetic Algorithms

- Computational methods inspired in natural selection.

- In nature better adapted individuals have higher chances to reproduce and survive.
  - These individuals will transmit and combine their genetic material, creating children.
  - The ones not adapted will have lower chances to transmit their DNA/RNA.

- GA emulates this behavior:
  1. Codify a population of individuals (each individual represents a possible solution to a problem).
  2. Evaluation of each individual of the population according the objectives of the problem. This requires to have defined a metric to test how good each individual is solving the problem (i.e. the FF).
  3. Selection of parents (a higher probability to be chosen as parent is assigned to those individuals with higher FF values).
  4. Creation of children as a combination of parents’ genes (using genetic operators as crossover, mutation, reproductor).
  5. Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated.

- Instead of an exhaustive exploration of all the possible solutions, this method is based in combining promising solutions to create newer ones prone to be improved versions of the formers.
Genetic algorithms.

1- Codify a population of individuals (each individual represents a possible solution to a problem).

   • In nature, the instructions to create lifeforms is codified in the DNA or RNA.
   • A string can emulate this codification. In this case, bits would be the equivalent of genes (exchangeable parts).

0/1 \rightarrow \text{Not include/include the signal/signal feature assigned to that box.}

The first, and only the first population is created by a random assignment of values.

\( \text{STD(FFT(signal))} \)
Genetic algorithms.

1- Codify of a population of individuals (each individual represents a possible solution to a problem).
2- Evaluation of each individual of the population according the objectives of the problem. This requires to have defined a metric to test how good each individual is solving the problem (i.e. the FF).
3- Selection of parents (a higher probability to be chosen as parent is assigned to those individuals with higher FF values).
4- Creation of children as a combination of parents’ genes (using genetic operators as crossover, mutation, reproductor).
5- Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated.

Fitness Function (FF) is a score system created to evaluate each candidate:
• 5 points: disruption predicted with anticipation at least 5 ms and no more than 1 second before the disruption (to avoid premature alarms).
• 3 points: no alarm is triggered in a non-disruptive shot.

Train Evaluated using the AUG validation dataset
SVM Candidate Predictor 1

Training dataset

FF=870

1, 2, 3, … 50

2nd IAEA TM 2017. MIT, Cambridge, MA, USA.
Genetic algorithms.

Selection of parents: roulette method.

1- Codify of a population of individuals (each individual represents a possible solution to a problem).
2- Evaluation of each individual of the population according the objectives of the problem. This requires to have defined a metric to test how good each individual is solving the problem (i.e. the FF).
3- **Selection of parents** (a higher probability to be chosen as parent is assigned to those individuals with higher FF values).
4- Creation of children as a combination of parents’ genes (using genetic operators as crossover, mutation, reproductor).
5- Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated.
1- Codify a population of individuals (each individual represents a possible solution to a problem).

2- Evaluation of each individual of the population according the objectives of the problem. This requires to have defined a metric to test how good each individual is solving the problem (i.e. the FF).

3- Selection of parents (a higher probability to be chosen as parent is assigned to those individuals with higher FF values).

4- **Creation of children as a combination of parents’ genes (using genetic operators as crossover, mutation, reproductor).**

5- Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated.
Genetic algorithms.

![Graph showing the normalized score over generations/iterations for the mean score of the population and the best individual of the generation.](image)

2nd IAEA TM 2017. MIT, Cambridge, MA, USA.
Results: direct extrapolation.

Training/validation

AUG 100 disruptive
1000 non-disruptive

Predictor

Testing results

Very high rates of false and premature alarms.

468 shots (77 dis/391 no dis)
False = 2.81%

4158 shots test (699 dis/3274)
False= 31.61%
Prematures= 52.64%
Results: providing 1 disruptive and 1 non-disruptive discharges of the target tokamak.

AUG

Training
100 disruptive

1000 non-disruptive

JET

1 disruptive

1 non-disruptive

2nd predictor*

JET, first 500 test shots (51 dis). False: 14.85%

This one can be a first predictor for the target tokamak.

JET, total 4857 test shots (699 dis).
False: 27.03%

*Its calculation has been repeated in two independent GA runs to verify the reliability of the results.
Results. GA training using JET data

Training: JET shots previous to the missed alarms (51 disruptives).
A new GA optimized predictor is created.

Testing

Next 1000 shots (201 disruptives)
after missing the first disruption.
False alarms= 0.99%

All the rest (647 disruptives)
False alarms= 8.69%
Summary and conclusions.

- In this work the cross-tokamak idea has been retaken to provide a solution for disruption mitigation in ITER.

- New guidelines have been followed to develop it.

- In this case, to attain high detection rates, some information of the target tokamak was required.
  - It has been also demonstrated, that after this kick-start (1 disruptive shot of the target tokamak), a reliable GA/SVM predictor can be created.
  - This is one of the few realistic candidate disruption prediction strategies for ITER.

- This is, up to now, the only ML-based predictor tested in a tokamak different of the one it was used to train it (besides Windsor’s work and achieving considerably higher prediction rates).
Discussion and future work

• ML techniques can create reliable disruption predictors.
  • But they need a large stored database.

• In this work the historical data from AUG have been used to create a disruption prediction system for JET.

• A possible interpretation of the results is that, even if the system learns the main signatures of disruptions, it needs some scaling factor (or some notion of the target device) to be applied in a tokamak different to the one it was trained in.

• Then, a promising path of research to improve the results would be to look for machine-independent parameters to predict disruption or scaling factors to gain accuracy in the extrapolations.
Thanks for your attention!
Training: a subset of AUG data is given as input to the ML system to create the predictor.

Validation: the data is used to test the trained system. These test results can be used to decide which predictor (from all the developed ones) will be used for the final test.

---

Final test: just one predictor (the one with highest validation scores) is selected after 50 iterations of the GA (~1 day).

Each discharge was analyzed, simulating a real-time operational scenario, from the beginning of the PC plateau till the end.

Evaluations were performed every 8 ms according a previous research**.

---

'FFT_BP'  'IP'  'FFT_IP'  'FFT_LI'  'FFT_Loca'  'FFT_PVP'  'q95'
GAM = 1 C = 100

'IP'  'FFT_IP'  'FFT_LI'  'FFT_Loca'
GAM = 1 C = 1000