WPJET1-CPR(17) 17632

GA Ratta et al.

Viability assessment of a AUG-JET cross-tokamak disruption predictor

Preprint of Paper to be submitted for publication in Proceeding of 2nd IAEA Technical Meeting on Fusion Data Processing, Validation and Analysis

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

This document is intended for publication in the open literature. It is made available on the clear understanding that it may not be further circulated and extracts or references may not be published prior to publication of the original when applicable, or without the consent of the Publications Officer, EUROfusion Programme Management Unit, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK or e-mail Publications.Officer@euro-fusion.org

Enquiries about Copyright and reproduction should be addressed to the Publications Officer, EUROfusion Programme Management Unit, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK or e-mail Publications.Officer@euro-fusion.org

The contents of this preprint and all other EUROfusion Preprints, Reports and Conference Papers are available to view online free at http://www.euro-fusionscipub.org. This site has full search facilities and e-mail alert options. In the JET specific papers the diagrams contained within the PDFs on this site are hyperlinked

Viability assessment of a AUG-JET cross-tokamak disruption predictor

G.A.Rattá[1], J.Vega[1], A. Murari[2], the EUROfusion MST1 Team[+], JET Contributors* and the ASDEX Upgrade team

[1]*Laboratorio Nacional de Fusión. CIEMAT, Madrid, Spain.*

[2] *Consorzio RFX Corso Stati Uniti 4, 35127 Padova. 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).*

ψ

Abstract.

The application of Machine Learning (ML) techniques for disruption prediction has improved the detection rates and the warning times in JET [1] and other tokamaks. However, ML predictors learn from past events, which imply an already stored database to develop them. Therefore, a significant problem arises at the time of developing ML-based systems for ITER. In this work to tackle this problem, a Genetic Algorithms-optimized (GAs) predictor based on a previous study [1] was trained using, initially, only AUG data and tested with a wide database of JET. This "smaller to larger" tokamak approach is meant as a test for future extrapolation of this technique to ITER.

The outcomes of the direct application of the cross-predictor resulted in a 30.03% of false alarms and more than a 42% of premature alarms, which indicates the need of some information about the target device to achieve reasonable performance.

Then, in a second approach, a new predictor was trained with AUG database plus one disruptive and one non-disruptive pulses of JET. The final cross-predictions (over the chronologically first 564 shots after the training, 52 of them disruptive) reached a 100% of total detected disruptions (all of them with anticipation times higher to 10 ms). The false alarms in that period were 7,42%. However, the performances decay for new shots were tested. This ageing effect is a known phenomenon and it can be tackled by the periodic retraining of the system. As proof of principle, a final predictor was create as an adaptive approach, getting in the following 1000 pulses (52 of them disruptive) a 91,75% of detections with at least 10 ms of warning times and less than the 1% of false alarms.

Keywords. Disruption prediction, ITER, cross-tokamak.

I. Introduction

Disruptions [1][2] remain as a key problem to solve in nuclear fusion research. Their prediction, sufficiently in advance to perform mitigation [3][4] or even plasma recovery actions [5], has improved considerably since the application of Machine Learning (ML) techniques.

Nowadays, the main efforts are mainly focused on the future application of prediction systems to the next step tokamak, ITER. Since ML techniques learn from past data, a problem

arises: at the beginning of ITER's operation no database will be available. A further problem is the high detection rates required which demands accuracies only achieved by the use of ML methods.

In the last years, ML approaches have been explored using a minimum amount of past discharges or even none of them in "learning from scratch" methodologies [6][7]. These techniques can be a promising area of research.

Another approach was the one introduced by Windsor et al. in 2005 [8]. They proposed to train a ML predictor using the database of one tokamak and to apply the developed model for another one. The underlying aim is the extrapolation of models in sights of ITER. The best results obtained in that work could not exceeds the 70% of detections and the research line was not continued.

The recent and successful incorporation of Genetic Algorithms (GAs) to the prediction of disruptions [9][10][11][12] and the considerable escalation of computational power since 2005 made worth retaking this cross-tokamak strategy with newer tools and ideas. In this paper GAs were applied to select the most relevant set of signals and signal features and to adjust some required parameters of the ML model.

Having in mind that a pragmatic solution will demand an extrapolation from a 'smaller' tokamak (as JET) towards ITER, we have created and tested different predictors using, mainly (as it will be later explained) AUG data. Each model was finally tested using a wide dataset of 4857 JET shots (699 of them disruptive).

As it will be shown, even if the direct extrapolation of the model created using only AUG was not successful, some alternative approaches providing a pragmatic solution was carried out.

This article is organized as follows. The databases (from ASDEX-Upgrade and JET) and the applied signal processing are the subject of Section II. Section III is devoted to explaining the ML methods used in this work (Support Vector Machines and Genetic Algorithms). Also, the application of these methods to the disruption prediction problem is explained. In Section IV, the results are shown and finally summarized and discussed in Section V.

II. The database and the processing of the signals

The gathered database from JET and AUG was selected taking into account the extrapolation from one device to the other. Therefore, signals measured by equivalent diagnostic systems in both devices were selected. They are: 1) the poloidal beta; 2) the line integrated plasma density; 3) the plasma volume divided by the device minor radius; 4) the

plasma current; 5) the plasma internal inductance; 6) the locked mode amplitude; 7) the plasma vertical centroid position; 8) the total input power; 9) the safety factor at 95% flux surface (the flux surface that encloses the 95% of the toroidal flux); 10) the total radiated power and 11) the time derivative of the stored diamagnetic energy.

Discharges with incomplete or unreliable measurements were omitted from the databases. Also, intentionally produced disruptions have been discarded.

AUG total database consisted in 232 disruptive shots and 1173 non-disruptive shots produced during the years 2012 and 2014. This database has been split into the training/validation database (chronologically first 100 disruptive and 1000 non-disruptive discharges) and the remaining shots were saved for testing.

The total JET database included 699 disruptive and 4158 non-disruptive discharges generated between April 2012 (experimental campaign C30a) and November 2016 (C37), all of them after the installation of the metallic wall. JET database has been (almost, as it will be explained in later Sections) exclusively for testing purposes.

The data processing was performed having in mind real-time conditions, following the procedure of previous works [13][14]. Signals were normalized between (approximately) 0 and 1. This normalization is approximate. To be exactly bounded between 0 and 1 it would imply to know beforehand the maximum and minimum values of each magnitude for all the signals in the database. In a realistic situation this can be estimated (but not exactly known) for a future device. Then, the maximum and minimum values of the signals of the first discharge of each database (JET and AUG) have been used for the normalization of the rest of the shots.

Previous investigations demonstrated that not only a good combination of plasma measurements is necessary to create reliable detection systems but also some preprocessing of the signals (getting signal features) can enhance the results [15]. Then, the standard deviation of the Discrete Fourier Transform (discarding the zero frequency component) computed over the past 32 ms of each signal was computed to create an extra input feature. Therefore, a total of 22 inputs have been considered in this study: normalized amplitude values of 11 signals and the 11 above-mentioned features of these signals.

For both databases, and to allow its better computational treatment, the signals and signal features have been resampled at a frequency of 1 kHz (which is compatible with real-time conditions as it has been detailed in a previous publication [16]).

III. Machine Learning and optimization methods

III.A. Brief introduction

Machine Learning (ML) comprises a wide set of methods destined to automatically create models from data. One of the most common types of ML technique is the ones called "supervised" learning. Supervised implies to provide labeled objects to the ML system. The label of the inputs (e.g. "disruptive sample") makes the algorithm learn and compute a general rule based on that training data.

Probably the two most extendedly used supervised methods are Neural Networks [17] (the one employed in Windsor's et al. work and others related to disruption prediction at JET [18], AUG [19] and JT-60U [20]) and the one chosen in the present and other previous works [13][14], called Support Vector Machines (SVM) [21]. This technique is conceptually introduced in the following subsection *III.B.*.

To attain the best possible predictions, the optimal set of input variables (in this case, plasma raw and processed measurement) and some SVM parameters (as it is explained in the next subsection) must be selected and tuned. For that task, a well-known and very efficient optimization technique called Genetic Algorithms (GAs) has been used. This technique is introduced in the next subsection *III.C*.

For deeper details regarding SVM and GA, there is available abundant bibliography to be consulted [21][22][23].

III.B. Support Vector Machines

SVM [21], for classification, works in two stages. In the first one, called training, a set of known examples labeled with the class they belong to (in this case "pre-disruptive sample" or "non-disruptive sample") is provided to the system. These examples are represented by squares and circles of Figure 1.

During the training stage, SVM is programmed to calculate a function or hyper-plane able to separate these objects by maximizing in the process the margin (m) between the classes. Once the training has been completed, a model/classifier is created (i.e. the hyper-plane equation). The samples, in this work, are vectors containing the value of a set of plasma signals (raw and processed) extracted from different times windows of the training set of discharges.

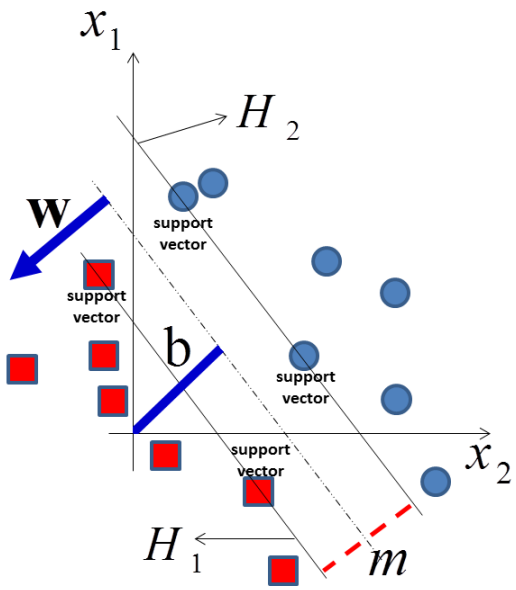

Figure 1. Bi-dimensional and bi-class conceptual example of a SVM classifier.

The time windows to extract the vectors were: 1) just before the occurrence of a disruption (to detect disruption precursors, e.g. squares in Figure 1); and 2) any time in a pulse that landed without a disruption (e.g. circles in Figure 1).

Performing this task for several samples of a wide number of discharges leads to a large set of vectors labeled as "pre-disruptive sample" or "safe sample". Once the training set is considered big enough, the hyper-plane (in this case, equivalent to the disruption predictor) can be created.

In the second stage, called testing, new examples are provided as inputs. Since the hyper-plane has been already calculated, the computation over the new inputs can be performed in the order of microseconds. Depending on which side of the hyper-plane the new samples get placed, the class they belong to (pre-disruptive or non-disruptive sample) is automatically determined. Notice that this methodology can be applied, from the beginning to the end, to each sample of a running discharge, obtaining by this way an on-line disruption predictor.

The conceptual example shown in Figure 1 is unrealistic for disruption prediction. On the one hand, the dimensional space where the hyper-plane is built in is normally >2 (in Figure 1, it is 2-dimensional).

On the other hand, normally, the set of training samples is non-linearly separable. So, the mathematical principles of SVM to calculate the separation between classes by maximizing the margin *m* can not be applied. To solve this issue, Kernel functions are used. Kernels transform the input space into another (normally higher dimensional) one. In that space, the

linear hyper-plane can be computed. The final solution (back into the input space) becomes non-linear in the case a non-linear Kernel is applied. Kernel functions can take any form and probably the most commonly used for its versatility and the one used in this work is the Radial Basis Function (RBF):

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \left\| \mathbf{x}_i, \mathbf{x}_j \right\|^2\right) \text{ RBF}$$

where $\left\| \mathbf{x}_i, \mathbf{x}_j \right\|^2$ is the squared Euclidean distance between two feature vectors.

As final comment, it is necessary to consider the importance of avoiding overfitting in order to get systems with enough generalization power (they must have good performances not only with the training samples but also at the time of testing it). For that, a parameter called slack variable (represented with the letter "C") can be added into the SVM optimization [21].

Before ending this Subsection, some important facts should be noticed:

1) If the RBF Kernel is used, the parameter γ must be predefined. This parameter defines the radius of influence of the support vectors. A very large γ will conduct to overfitting whereas a low γ value leads to a more linear hyper-plane.

2) The value of the slack variable C has to be predefined. This parameter trade-off misclassification of training examples (allowing some feature vectors to be positioned between the boundaries) against the simplicity of the decision surface. A low C value makes the decision hyperplane smoother.

3) The method creates the classifier using input examples (signal features). However, it does not select the optimal set of features.

The free licensed software used to work with SVM was Libsvm [24] adapted to run under MATLAB [25][26] environment.

III.C. Genetic Algorithms

The three points listed at the end of the previous Section are fundamental for creating reliable and accurate disruption predictors. However, they are not optimized by the SVM method itself.

To increase the probabilities of achieving better results, it is necessary a careful selection of the input parameters and the adjustment of internal variables (γ and C in SVM). At first sight, it would not represent a big problem. This could be solved by just performing an exhaustive analysis which means to look for all the possible combinations of signal, signal features (std(DFT(32 ms of signal))), γ and C parameters. Then, for each possible

combination of these input variables a disruption predictor can be trained and tested. Once this procedure is finished, the one with the highest score (the score can be, for instance, the total prediction rate over the testing discharges) can be selected.

However, the amount of possible combinations (without permutations) can be calculated with the following formula:

$$\sum_{i=1}^{n} C_i^n$$

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

where:

n=number of possible values and i=possible groupings

n can be calculated as the sum of: 1) 11 possible signal values; 2) 11 possible signal features values; 3) 10 possible values for γ; 4) 10 possible values for C. They complete a total of 42 values.

Then, n=42 and i= 1,2,3,…,42

To explore each one of the possible combinations (considering that each one needs a computational time of ~3 seconds), it would take more than 418383 years to finish the exhaustive analysis.

To solve this problem in affordable times, GAs (optimization methods inspired by biological evolution) [27][28] were applied.

In nature, the better adapted individuals have higher chances to survive, mate and transmit their characteristics to their progeny. Adaptation means, in this context, to survive and to reproduce. Individuals unable to accomplish these objectives will not pass their genes and therefore their configuration is destined to extinction.

GAs imitate that behavior by creating a population of solutions. Each solution is evaluated with a metric, called Fitness Function (FF). According the FF, higher possibility to be a parent is assigned to those solutions with better scores. Descendants are created as a combination of parents characteristics/genes. Finally, and since descendants are a combination of promising configurations, it is expected that newer generations outperform the former ones.

The GAs steps can be summarized as follows:

1- Creation of a population of possible solutions. The higher the population the higher the possibilities to achieve better results. However, the computational times increase exponentially with the population size. In this study 50 individuals per generation were chosen as a good trade-off between results quality and computational times.

Each individual of the population represents a set of instructions to create a SVM-based disruption predictor *candidate*. For the correct evolution of the algorithm all the information of each individual needs to be codified. For that, a simplified computational version of the DNA is used. For that, strings are used (see Figure 2). In this case, the codification consisted of assigning 'ones' and 'zeros' randomly only in the first iteration (in the following ones through the GAs

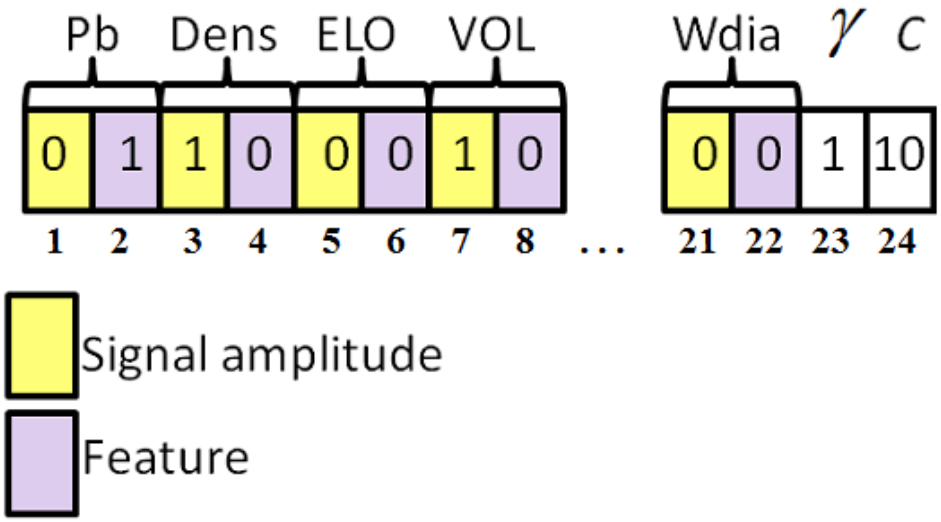

Figure 2. Example of an individual codification. The exemplified predictor uses the Poloidal beta feature, the amplitude of the line integrated density and the amplitude of the plasma volume. It has a γ value equal to 1 and a C value equal to 10.

evolution) to the strings boxes 1 to 22. They correspond to 11 signal amplitudes and to 11 signal features (the $std_{FFT(signal)}$) as it is represented in Figure 2. In the codification, the 'ones' indicate that the signal or the processed signal feature in that box must be included in the predictor development; the 'zeros' imply they must be not. Positions 23 and 24 correspond to the value of the γ and C parameters (mentioned in the previous Section). The selected possible γ values were: {0.00001; 0.0001; 0.001; 0.005; 0.01; 0.05; 0.1; 0.5; 1; 10} and the possible C values: {0.01; 0.5; 1; 5; 10; 50; 100; 1000; 10000; 100000}.

Notice that the main instructions to train each individual are enclosed in short strings. The next step is to follow these instructions and train (using the training database) a SVM candidate model. This training process is analogous to the ones explained in a previous paper [12]. The data is extracted from the AUG training database. From there, 1000 non disruptive and 50 pre-disruptive samples are randomly selected and the candidate predictor is built with them. Then, the candidate is tested over 100 extra validation shots (50 of them that ended in a

disruption) to evaluate its performance. The procedure is repeated once, swapping training and validation groups, in a 2-fold cross-validation method, to get more robust results.

2- The 2 results achieved in the abovementioned 2-fold cross validation are averaged, getting by this way the Fitness Function (FF) score. The procedure is repeated for each individual of the population, getting as result 50 FF values. In this study, criteria to define the scores was similar to the one adopted in a previous publication: for each predictor candidate 5 points were assigned per disruption predicted with warning times between 5 ms and 1 seconds (to avoid premature alarms) and 3 points in case, correctly, no alarm is triggered in a non-disruptive shot.

3- The next step consists of selecting the parents. For that, the roulette method was applied: *i)* all the individuals are sorted according theirs FF scores; *ii)* FF scores are normalized between [0; 1]; *iii)* a random value between 0 and 1 is chosen; *iv)* all individuals whose normalized fitness is over the random value are added to the bag of selected parents; *v)* the process is repeated from step *i* till the number of parents is equal to the population number (50 in this case).

4- Children are created as a combination of parents' genes. To do it, the 2 points cross-over operation, depicted in Figure 3, was conducted. It consists of randomly selecting 2 points in parent's strings. Then, the sections determined by these points are interchanged between parents to create offspring. Also, the mutation probability (0.05%) of randomly changing the value of a bit in an offspring has been included.

5- Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated. In this work, the ending condition was triggered after 50 iterations since they were enough to get good results in affordable computational times.

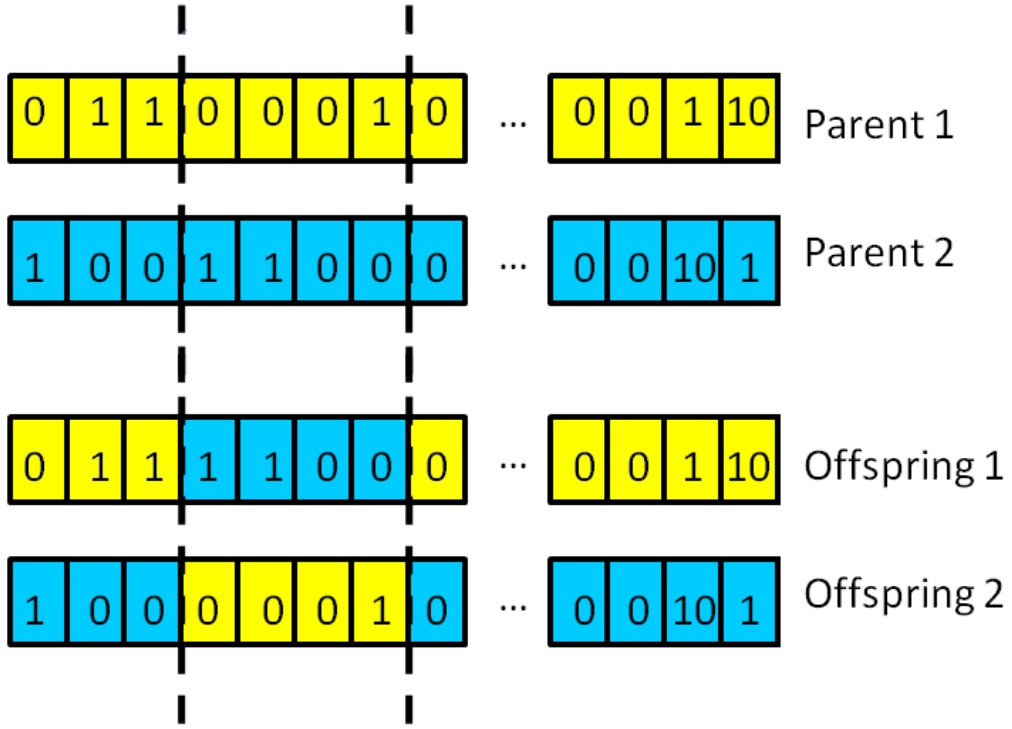

Figure 3. Example of the two points cross-over operation. Two sections are randomly selected from parents. The segments enclosed by these limits are interchanged to create two offspring.

As the final comment of this Section, it is important to highlight that only the first population is created by a random assignation of values. In posterior iterations, the new individuals are driven by the heuristic of the GAs.

IV. Results

IV.A. Direct AUG/JET extrapolation

Going for the main objective of this work, a direct AUG to JET extrapolation was attempted. Following the methodology detailed in Section III, SVM and GAs were combined to get an optimized predictor. This predictor was built using only AUG training data and it was tested over the whole JET testing dataset (4158 non disruptive and 699 disruptive pulses).

Even it was beyond the scope of this work, an extra test was performed over an independent AUG set of 77 disruptive and 391 non-disruptive discharges produced after (chronologically) the ones used for training.

To understand the results summarized in Figure 4, some basic definitions should be mentioned. The vertical lines are drawn to highlight relevant times. The ones at 5 ms and 10 ms of warning times are set according to period required for mitigation in AUG and JET respectively. The lines at 1 second and at 1,5 seconds delimits which alarms are triggered too

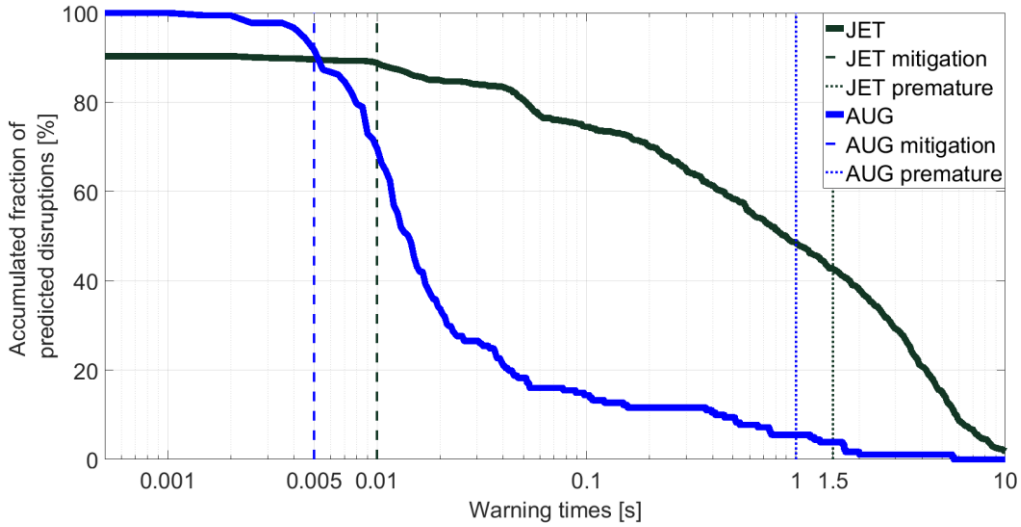

Figure 4. Response of the predictor trained using only AUG data. The good performances over the AUG testing database suggest an adequate training. However, its lack of generalization capabilities is unveiled at the time of assessing JET data. There, the false and premature alarms are excessively high, overpassing the 30% and 40% respectively.

early before the disruption and therefore they can be considered as "premature". These plots are built only using disruptive discharges. Therefore, the analysis of the predictor over the non-disruptive shots are usually summarized by just providing the false alarms rates.

The results reveal that the performances are as good as expected over AUG testing database (all the disruptions are predicted, and ~97% of them foreseen with warning times of at least 5ms). However, over the target tokamak (JET) the landscape changes and unacceptable rates of false alarms (30,03%) and premature alarms (42,78%) are attained. The feature vector chosen by the GAs were: the Poloidal beta, the Plasma density *feature* (the features were extracted as the std[FFT[32 ms of signal]]), the plasma volume (normalized by the device radius, the Plasma current feature, the plasma internal inductance feature, the locked mode amplitude, the plasma vertical centroid position feature, the radiated power and the time derivative of the stored diamagnetic energy).

The most probable cause of these poor outcomes of the extrapolation is the lack of information of the target device during the training.

IV.B. Adding 1 disruptive and 1 non-disruptive shots from the target tokamak (JET)

Since the direct extrapolation was not successful, a second attempt, including just 2 discharges of the target tokamak during the validation, was performed. Having always in mind

a practical solution towards ITER, the chronologically first disruptive and non-disruptive discharges from JET were the ones selected, mixing them with the validation pool of AUG discharges. For the better understanding of this procedure, the reader might need a more precise indication of the difference between training, validation and testing groups. The training group contains the dataset (only from AUG in this case) given to the SVM machine learning system to build the disruption predictor (i.e. to compute the hyper-plane). Several (thousands) predictors are developed using only with training data. Each one is evaluated and optimized by GAs. A database (different than the training set) is required to identify if each of the predictors is performing well. The validation is part of the system development and refinement stage whereas the testing must be done only once a final model has been already selected.

To provide an example involving disruptions prediction, it is not the same to create one million models using AUG data, to test all of them over a JET database and, *after* the testing, to pick up the one that obtained the best rates. This *a posteriori* selection is not suitable in cases the objective is a realistic application in newer devices (as ITER). The correct way to proceed is to define a validation subset and to use it to choose the best predictor before (and not after) the testing. This difference between validation and testing datasets is fundamental to have realistic outcomes.

The point of this training (using AUG) data and validation (using AUG data plus 2 JET pulses) is to take advantage of a wide and already available database and simultaneously to force the system to learn how to scale the model to a different tokamak. During the validation, a very high score (1000 points) was assigned each time the 2 JET shots were correctly evaluated (no alarm in the disruptive shot and alarm between 10 ms and 1,5 second in the disruptive). This large reward was necessary; otherwise the 2 JET validation discharges would be treated with the same relevance than each one of the other 50 AUG validation pulses.

The testing results, after adding the 2 extra discharges, can be seen in Figure 5. The selected signals and signal features were: the plasma current, the plasma internal inductance signal and its feature, the locked mode amplitude feature, the plasma vertical centroid position feature, the safety factor at 95% signal and its feature. The outcomes over the first 564 experiments are shown. Notice that in this set all the disruptions (52) are correctly recognized, with a false alarms rate of 7,42%. After that (starting at the following testing discharge, the 565^{th}) the system misses for the first time a disruption and it begins to show evidence of ageing [28], as depicted Figure 6. This "ageing" refers to the deterioration of a predictor performance after newer configurations than the ones used for training are engaged during the

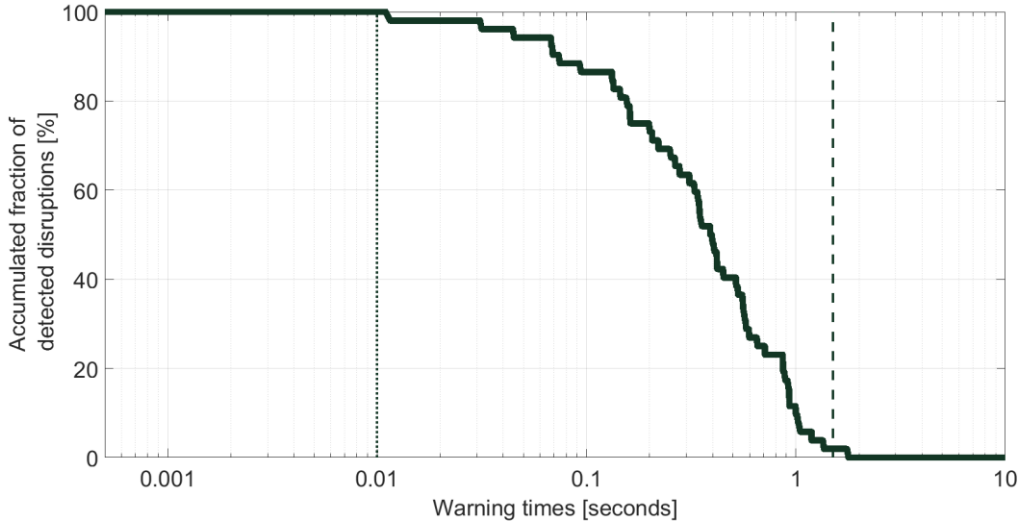

Figure 5. This second predictor was trained with AUG data and 2 JET validation discharges. These are outcomes over the first 564 pulses. All (52) disruptions are predicted with at least 10 ms of anticipation. The false alarms rate is a 7,42%.

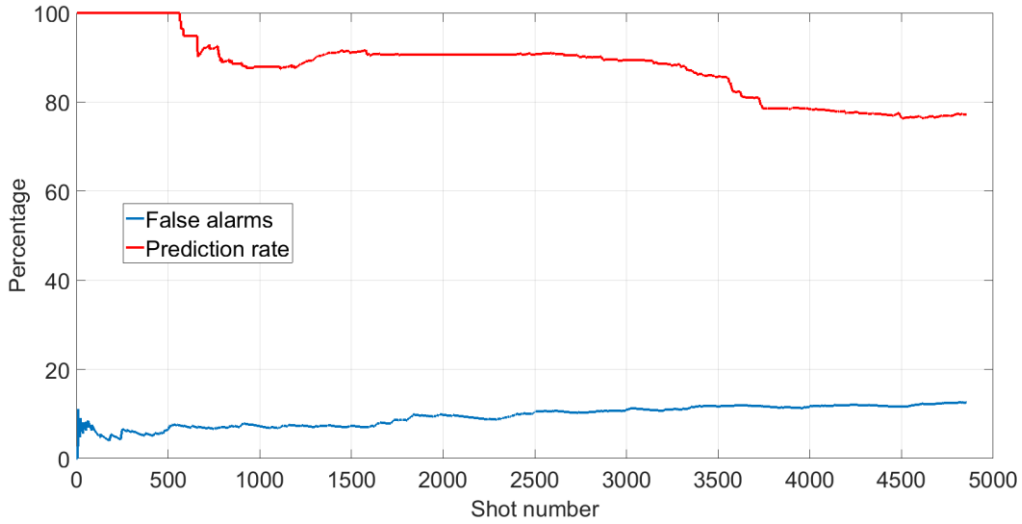

Figure 6. An ageing effect of the predictor is evidenced. As new shots are produced, the false alarms increase and the accuracy in anticipating disruptions decay.

testing. This effect can be avoided by the continuous retraining of the system to force it learn from the more demanding experiments.

IV.C. Facing the ageing effect by retraining the predictor, after missing the first disruption, with the data gathered before the missed alarm.

The strategy of adding 2 discharges considerably improved the results. The ageing effect, as it was stated at the end of the last subsection, can be easily patched by the continuous retraining of the system. Just to provide a practical example (that also evidences that the retraining does not really need to be so frequent) a third and final predictor was developed. This last system uses only the JET testing discharges that were produced before missing the first disruption. This can be a pragmatic solution towards a future device: a system can be created using an already existing database from a smaller tokamak plus 2 shots from the target tokamak. Then, it can be used until it fails predicting a disruption. Finally, all the gathered information until the failure can be used to create a patched predictor.

This retrained system was again developed under the guidelines detailed in Section III, but using the 564 pulses (52 of them disruptive) of JET instead the ones from AUG for training and validation. In Figure 7 its performance over the following 1000 discharges (194 of them disruptive) after the failure can be observed. Even trained with a reduced database, the rates are on the top of what nowadays can be achieved for disruption prediction, with a 91,75% of recognitions with warning times of at least 10ms and less than a 1% of false alarms. The statistics, extended to all the discharges after the failure (4293, 564 disruptive) are also depicted. In this case the false alarms reach the 8,69%. In this case, the GAs selected only the locked model amplitude signal and its feature and the safety factor signal.

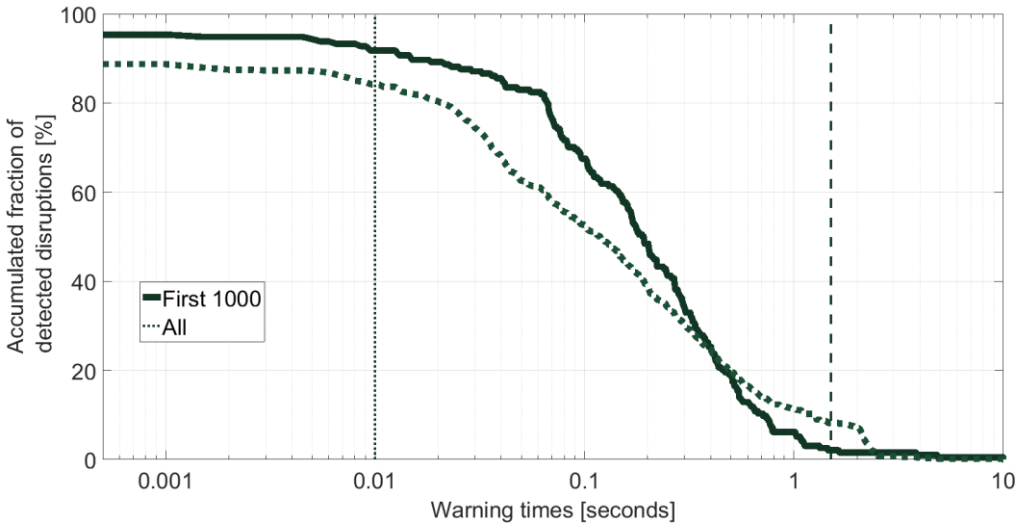

Figure 7. Outcomes of the last predictor over the 1000 shots (52 of them disruptive) after the failure, and all of the remaining pulses (4293, 564 disruptive) after missing the first disruption. The false alarms, respectively, are 0,99% and 8,69%.

V. Summary and discussion

The potential hazard of disruptions demands to investigate all possible paths to reach a solution applicable to ITER. Predicting them sufficiently in advance would allow coordinating mitigation or even avoidance actions to alleviate or dodge their harmful effects. The prediction of disruptions has significantly improved in the last decade since ML-based systems have been applied to tackle them. However, the dependency of these techniques on already stored databases represents a problem for future devices at the beginning of their operation.

The main effort introduced in this research has been focused on the extrapolation of a predictor developed with a smaller tokamak database (in this case AUG) and to apply it to a larger target device (JET). This direct application was not successful, mainly because of the unacceptably large rate of false alarms triggered during the testing. An attempt to improve this behavior (not detailed in this paper because it was also unfruitful, attaining even worst results) was to use dimensionless inputs, as in Windsor's paper [8], instead of the signal and signal features detailed in Section II. The most obvious conclusion is that this procedure can create an accurate predictor based on the training information. Even more, the model can be flexible enough to distinguish disruptive (or not) patterns in pulses quite different to the ones used during the training stage, as it has been proven in the testing over an independent set of AUG discharges. However, at the time of applying it to a different device it seems evident that some scaling factor is needed. This is the reason why in a second attempt two discharges (1 disruptive and one non disruptive) from the target tokamak, JET, were added into the validation bag. In this way, and following the same GAs optimization procedure, a second predictor was created. In this case, a considerable improvement of the prediction rates was achieved (100% of the 52 chronologically first disruptions were anticipated) and a simultaneous decay of the false alarms (to a 7,42% in the first 564 testing shots) was obtained. The rates after these first 564 testing pulses decay, showing an ageing effect that can be easily overcome by the periodic retraining of the predictor. To proof that this can be a pragmatic strategy in sights of future devices, a third model was created. In this last case, only the data of JET produced before the missed alarm (564 pulses, 52 disruptive) was used. The rates obtained over the following 1000 shots (after the missed alarm) are among the best ever obtained to predict disruptions: 91,75% of detections with at least 10 ms of warning times and less than a 1% of false alarms.

It is clear that the system here presented is not ideal, mainly for the need of some information of the target tokamak to get a reliable enough model. However, nowadays, it is one of the few pragmatic solutions to face the prediction of disruptions in ITER.

A possible research direction to overcome the main problem detected here might be to study possible machine-independent variables or robust scaling variables to adjust a predictor developed with "smaller" tokamaks data to make it work acceptably in larger ones.

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] F.C. SCHULLER. "Disruption in tokamaks". Plasma Phys. Control. Fusion 37. A135–62. 1995.

[2] A.H. BOOZER. "Theory of tokamak disruptions". Physics of Plasmas.19, 058101. 2012.

[3] M. LEHNEN et al. *Impact and mitigation of disruptions with the ITER-like wall in JET*. Nucl. Fusion 53 093007 doi:10.1088/0029-5515/53/9/093007. 2013.

[4] L.R. BAYLOR et al. *Pellet fuelling, ELM pacing and disruption mitigation technology development for ITER*. Nucl. Fusion 49 085013 doi:10.1088/0029-5515/49/8/085013. 2009.

[5] G. PAUTASSO et al. *Assimilation of impurities during massive gas injection in ASDEX Upgrade.* Nucl. Fusion 55 033015. 2015.

[6] S. DORMIDO-CANTO et al. *Development of an efficient real-time disruption predictor from scratch on JET and implications for ITER.* Nucl. Fusion **53** 113001 2013.

[7] J. VEGA et al. *Adaptive high learning rate probabilistic disruption predictors from scratch for the next generation of tokamaks.* Nucl. Fusion **54** 123001 2014.

[8] C.G. WINDSOR et al. *A cross-tokamak neural network disruption predictor for the JET and ASDEX Upgrade tokamaks.* Nucl. Fusion 45 337. 2005.

[9] G.A. RATTÁ et al. *Improved feature selection based on genetic algorithms for real time disruption prediction on JET*. Fusion Engineering and Design. Volume 87, Issue 9. Pages 1670–1678 2012.

[10] G.A. RATTÁ et al. *Simulation and real-time replacement of missing plasma signals for disruption prediction: an implementation with APODIS.* Plasma Phys. Control. Fusion 56 114004. 2014.

[11] A. PEREIRA et al. *Feature selection for disruption prediction from scratch in JET by using genetic algorithms and probabilistic predictors.* Fusion Engineering and Design. Volumes 96–97. Pages 907–911. Proceedings of the 28th Symposium On Fusion Technology (SOFT-28). 2015.

[12] G.A. RATTÁ et al. *Global optimization driven by genetic algorithms for disruption predictors based on APODIS architecture.* Fusion Engineering and Design 112, 1014-1018. 2016.

[13] G.A. RATTÁ et al. *An advanced disruption predictor for JET tested in a simulated real-time environment.* Nucl. Fusion 50 025005. 2010.

[14] J. VEGA et al. *Results of the JET real-time disruption predictor in the ITER-like wall campaigns.* Fusion Engineering and Design. Volume 88, Issues 6–8. Pages 1228–1231. 2013.

[15] G.A. RATTÁ et al. *Feature extraction for improved disruption prediction analysis at JET.* Rev. Sci. Instrum. 79, 10F328 http://dx.doi.org/10.1063/1.2965020 2008.

[16] J.M. LÓPEZ et al. *Implementation of the Disruption Predictor APODIS in JET's Real-Time Network Using the MARTe Framework.* Nuclear Science, IEEE Transactions on (Volume:61 , Issue: 2) pp 741 - 744. 2014.

[17] J. HERTZ et al. *Introduction to the theory of neural computation*. Addison-Wesley. 1991.

[18] B. CANNAS et al. *Disruption forecasting at JET using neural networks*. P. Nucl. Fusion 44 68. 2004.

[19] G. PAUTASSO et al. *An adaptive real-time disruption predictor for ASDEX Upgrade.* Nucl. Fusion 50 075004. 2010.

[20] R. YOSHINO. *Neural-net disruption predictor in JT-60U*. Nucl. Fusion 43 1771. 2003.

[21] C. CORTES AND V. VAPNIK. *Support-vector networks*. Mach. Learn. 20 273–97. 1995.

[22] M. MOHRI, A. ROSTAMIZADEHAND A. TALWALKAR. *Foundations of ML.* Cambridge, MA: MIT Press. 2012.

[23] S. THEODORIDIS AND K. KOUTROUMBAS. *Pattern Recognition*. 4th edn (London: Academic). 2009.

[24] T. JOACHIMS. *Advances in Kernel Methods - Support Vector Learning*. MIT-Press. 1999.

[25] T. BRIGGS. MATLAB/MEX Interface to SVMlight. http://sourceforge.net/projects/mex-svm/

[26] MATLAB and Statistics Toolbox. The MathWorks, Inc., Natick, Massachusetts, United States

[27] K.A. DE JONG ET AL. *Parallel Problem Solving from Nature*. LNCS, vol. 496, pp. 38-47. Springer, Heidelberg. 1991.

[28] J. VEGA, ET AL. *Disruption Precursor Detection: Combining the Time and Frequency Domains*. SOFE program. 26th Symposium on Fusion Engineering (SOFE 2015). Austin (TX), USA. 2015.