EUROFUSION WPJET1-PR(14) 11949

GA Ratta et al.

Simulation and real-time replacement of missing plasma signals for disruption prediction: an implementation with APODIS

Preprint of Paper to be submitted for publication in
Plasma Physics and Controlled Fusion

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.

Simulation and Real-Time Replacement of Missing Plasma Signals for Disruption Prediction: An Implementation with APODIS

G.A. Rattá[1], A. Murari[2], J. Vega[1],
and JET EFDA contributors*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

[1]Laboratorio Nacional de Fusión. CIEMAT, Madrid, Spain
[2]Consorzio RFX, Associazione EURATOM/ENEA per la Fusione, Padua, Italy
** See annex of F. Romanelli et al, "Overview of JET Results",*
(24th IAEA Fusion Energy Conference, San Diego, USA (2012)).

Preprint of Paper to be submitted for publication in
Plasma Physics and Controlled Fusion

ABSTRACT

So far, the best results for real-time disruption prediction on JET (Joint European Torus) have been achieved with APODIS (Advanced Predictor Of DISruptions). APODIS is a data-driven system whose last version has been implemented in JET's real time-data network. It has been designed for the real-time analysis of features (mean and frequency values) corresponding to 7 plasma signals in order to foresee incoming disruptions.

In this article, non-linear regression techniques are applied to create (off-line) signal models. The models are able to generate (in real-time) "synthetic" signals. Therefore, these "synthetic" signals can be used to replace the original ones in the case they are in error or missing.

Under these conditions APODIS has been tested emulating real-time operation. The simulation results demonstrate that once a signal in error is replaced by the generated "synthetic" one, APODIS performances are considerably improved. The development of the regression models and the implications of the results are detailed and discussed in this paper.

1. INTRODUCTION

The prediction of disruptions [1] is a priority research subject in nuclear fusion. Their early identification would allow performing avoidance or mitigation actions in order to minimize the harm they can inflict. So far, the best online results obtained in JET (in terms of detection rates and prediction times) have been achieved with APODIS. APODIS is a multilayer architecture of machine learning classifiers trained to foresee disruptions before their occurrence.

The first version of APODIS was developed in 2009 [2] simulating a real-time scenario. Since then, it has been subjected to upgrades. Finally, its last version has been installed in the JET-EFDA real-time data network to be operated with the new metallic wall [3].

To perform predictions, every 32ms APODIS computes and analyses 2 different features extracted from 7 plasma signals. The features are the mean value and the standard deviation of the Fast Fourier Transform (after removing the DC component). The 7 signals are: plasma current, mode lock amplitude, plasma internal inductance, time derivative of the stored diamagnetic energy, total radiated power and total input power.

All the signals are necessary for the proper operation of the predictor. However, a recent study [4] has proven that the mode lock amplitude and the plasma internal inductance are essential for APODIS. Very low performances are achieved if any of these two specific signals is in error or missing. Besides, a recent failure in the stored diamagnetic energy time derivative signal at the beginning of the campaign C31 of JET induced APODIS to fail in the detection of several incoming disruptions.

Once a signal is known to be in error, APODIS and any other control or safety system may be compromised. An immediate practical solution is required in these circumstances to allow the systems proper operation.

In the present study, the developed solution is to create (off-line) Signal Feature Models (SFMs)

using two different non-linear regression techniques. The models are meant to estimate the time evolution of the signal features required by APODIS. This paper simulates both the real-time generation and the use of the "synthetic" signal estimations in replacement of the original features that APODIS requires, just to overcome the lack of one of them.

The article is structured as follows: section 2 briefly introduces the signal features APODIS requires to operate. Section 3 describes the database gathered from the JET device and used in this study. Section 4 is devoted to explaining the idea of the SFMs and to summarize the mathematical basis of the applied regression techniques. The overall testing procedure, that includes the application of the SFMs to estimate the signal features (under a simulated real-time scenario) and their application to the APODIS system, is the topic of section 5. Section 6 reports the results of the simulations. The discussion of the obtained results and their implications is the subject of section 7.

2. APODIS INPUTS

The development of data-driven systems to predict disruptions started around two decades ago as alternative to the physics-based models. Since these systems learn from data, computational treatment of the information plays a fundamental role to achieve good results. It has been proved that the proper process of the signals in order to extract from them the most relevant information can improve the early detection of disruptions [3][5]. The process that leads to identify these adequate signatures in signals, called feature extraction, allows discarding the redundant or useless data retaining only the problem-related information.

The features are computed in real-time and applied as inputs to the APODIS system. Notice that the inputs to the disruption predictor are not the signals themselves but the features extracted from them.

In the specific case of APODIS, the feature extraction performed over the signals is aimed to condense and simplify the data associated to disruptions.

In the last years, it has been determined that 2 specific features are particularly useful for APODIS: 1) the mean value and 2) the standard deviation of the discrete Fourier Transform (after removing the DC component) [2,3]. For the 7 signals APODIS analyses, the features are computed over 32 ms time windows after normalization:

$$\text{Normalized signal} = \frac{\text{Signal value} - \min}{\max - \min} \tag{1}$$

where min and max represent the minimum and maximum value of the signal in the training dataset.

A very relevant point arises here. APODIS inputs are not the signals themselves, but the 2 features extracted from them. Therefore, the inputs it requires are the signal *features*. In the graphical representation of Figure 1 for one of the signals, the stored diamagnetic energy time derivative (dW/dt), this idea is shown graphically .

Therefore, from the APODIS point of view, the missing/in error signals problem can be addressed

from two different angles:

1) Creating (off-line) *signal models*. Once signal models are created, they can estimate (in real-time) the values of the signal in error. Thus, the 2 features that APODIS requires can be calculated from the estimated "synthetic" signal. Finally, the 2 extracted features can be input to APODIS.
2) Creating (off-line) *SFMs*. The SFMs can estimate (in real-time) the *features* that APODIS requires. Those "synthetic" features can be directly used as inputs for APODIS.

Signal features are derived from the signals and, as it can be seen in Figure 1, they present a lower complexity. Approximating a simpler function is more feasible. For that reason, in this study we embraced the second option (creating SFMs instead of developing signal models).

3. DATABASE

The database has been extracted from JET, the largest operational magnetic confinement nuclear fusion device in the world and the one where APODIS is currently in operation. The collected database (see Table 1) contains all the discharges produced in the device from July 2011 to July 2013. The period includes the shots from the beginning of the metallic ITER-like wall campaigns (campaigns C28) to the beginning of campaign C31. The replacement of the device wall represented a challenge for APODIS, a system trained with carbon-wall discharges and unmodified (i.e. no retraining) since its installation in the real-time data network.

The first (in chronological order) 586 discharges were used as training dataset. With these shots the SFMs are created off-line.

The remaining 660 discharges (belonging to the period between C28-C30) were saved for testing. This dataset has two purposes. First, to quantify the goodness of fit of the SFMs. Second (and more relevant), to test APODIS with the modelled signal features. The testing is accomplished under a realistic emulation of real-time conditions.

In the first 132 discharges of the C31 campaign the BetaLi [6] (a real-time code running at JET and devoted to reconstruct, among others, the value of the stored diamagnetic energy time derivative signal dW/dt), was in error. Therefore, the dW/dt was unavailable in real-time and APODIS was not able to predict any of the 15 unintentional disruptions that occurred in that period. This is exactly the operational situation this study tries to tackle and provides a good opportunity to test, in a true operational scenario, what would happen in case of having SFMs to replace the features of the signal in error. Therefore, the first 132 discharges of campaign C31 (July 2013) were added to the database to test the SFMs for the dW/dt (a summary of the database is provided in Table 1).

Having in mind a realistic operational scenario, it is convenient to create the models using a reduced number of signals, and of course all of them ought to be available in real-time. Thus, in this work only the 7 real-time signals required by APODIS, plus 3 extra ones, have been considered . The 3 extra signals satisfy the condition of real-time availability. Also, they have been used in previous studies related to the prediction of disruptions [2,7]. The 3 signals are: the toroidal magnetic

field, the poloidal Beta and the plasma vertical position. Adding these 3 quantities to the 7 signals APODIS requires, it makes a total of 10 signals gathered for this study (listed in Table 2).

The first 7 parameters are the ones that APODIS requires. The extra 3 parameters at the bottom of the table have been also included to create the SFMs.

4. SFMS AND THE APPLIED REGRESSION TECHNIQUES

So far, it has been explained that the idea of this paper, to overcome the sporadic lack of real-time signals that APODIS requires to operate, is to replace the predictor missing inputs with "synthetic" ones. More accurately speaking, the goal is not to generate "synthetic" signals but "synthetic" signal *features* (specifically, the mean values and the standard deviation of the fast Fourier transform, computed every 30ms).

To generate "synthetic" signal features under realistic operational conditions, they have to be inferred only from signals available in real-time. Regression analysis is the technique devoted to create models as relationships among variables and therefore the most suitable for the aimed purposes.

Two non-parametric and non-linear regression techniques that have earned popularity in the last years are Support Vector Regression (SVR) [8] and Symbolic Regression [9] (based on Genetic Programming). These methods are suitable for the regression problem here under study which involves complex and non-linear relationships among the plasma parameters.

SVR is based on kernel methods and it has been conceived as an extension of a classification method called "Support Vector Machines". It normally provides good results in short computational times. However, it requires the tuning of some internal parameters to get better regression models. Its mathematical principles and the creation of SFMs through it are explained in the next Subsection 4.1.

Symbolic regression is an even more recent method and tackles the regression problem with Genetic Programming. It involves an iterative optimization method based on the evolution of living organisms. Even if this approach does not guarantee reaching an optimal solution, in practice it normally reaches excellent results. The basis of Genetic Programming and Symbolic regression are the subject of Subsection 4.2.

4.1. SUPPORT VECTOR REGRESSION

Recapitulating, the main goal of in this study is to create a regression function able to estimate the value of a signal feature. For instance, in the specific case of the mean value feature for the plasma internal inductance (LI) signal (let us call it $\hat{m}\_LI$) the expected function should have the form:

$$\hat{m}\_LI = f(\mathbf{x}^m{}_i) + \text{constant} \tag{2}$$

Here $\mathbf{x}_i^m$ is a vector with the remaining 9 signal *mean* features (in this case, mean features of signals 2 to 10 in Table 2) for the sample i.

Similarly, for the standard deviation of the Fast Fourier Transform (removing the DC component) feature, the expected model should have the form:

$$\hat{std(FFT)}\_LI = f(\mathbf{x}^{std(FFT)}{}_i) + \text{constant} \tag{3}$$

For other SFMs, for instance the ones corresponding to the mode lock amplitude (ML) and the dW/dt, analogous functions are expected. Of course, in those cases the LI features will be part of the remaining 9 features at the right hand side of equations (2) and (3).

In nuclear fusion devices, the relationships among the different plasma parameters usually are complex and non-linear. To estimate a signal feature value using classic linear or polynomial regressions leads to poor approximations. To tackle the problem the SVR technique has been applied [8]. SVR is a non-linear method based on statistical learning theory [10]. It is conceived to "learn" from data in a so called "training procedure" attempting to reach results with high generalization capabilities.

To introduce the mathematical basis of SVR, let us consider a set of training data $\{(\mathbf{x}_i, y_i),...., (\mathbf{x}_l, y_l)\}$ $\mathbf{x}_i \in R''$ and $y_i \in R$ for $i = 1,..., l$ with $l =$ number of training samples.
Then, y_1 is the mean LI value for sample 1, and $\mathbf{x}_1$ is a vector containing the mean values of the remaining signals.

The main form of the regression function using SVR is:

$$f(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) + b \tag{4}$$

where $\mathbf{w} \in R''$, $b \in R$ and Φ denotes a non-linear transformation from R'' to a higher dimensional space. SVR is based on the structural minimization principle, so two objectives have to be satisfied to find the values of $\mathbf{w}$ and b: first, the function $f(\mathbf{x})$ should fit properly the "true" values of the set of output values y; second, in order to guarantee generalization, it is necessary to avoid overfitting and, therefore, $f(\mathbf{x})$ should be as smooth as possible.
To measure the quality of the fit, the ε-insensitive loss function is introduced. The idea is to quantify the absolute error between the true values and the fitting with a tolerance ε By convention, the loss is linear with an insensitive zone ε. Then, if $f(\mathbf{x}_i)$ is inside of the bounds of tolerance $y_i \pm \varepsilon$, no loss is considered:

$$L_\varepsilon = \begin{cases} 0, & for \quad |y - f(\mathbf{x})| \le \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & otherwise \end{cases} \tag{5}$$

It is necessary to minimise the empirical error on the training set and the parameter norm at the same time. For that, slack variables ζ_i and ζ_i^* that measure the distance between samples and ε are introduced. Also, Schölkopf [11,12] added a term ν to solve the primal problem in the "Nu" version of SVR:

$$\min \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C\left[\upsilon\varepsilon + \frac{1}{l}\sum_{i=1}^{l}\left(\zeta_i + \zeta^*{}_i\right)\right] \tag{6}$$

$$\begin{aligned} &\left(\left\langle \mathbf{w}, \Phi\left(\mathbf{x}_i\right)\right\rangle + b\right) - y_i \le \varepsilon + \zeta_i \\ &y_i - \left(\left\langle \mathbf{w}, \Phi\left(\mathbf{x}_i\right)\right\rangle + b\right) \le \varepsilon + \zeta_i^* \$ \\ &\varepsilon, \zeta_i, \zeta_i^* \ge 0, \qquad i = 1,...,l \end{aligned} \tag{7}$$

The regularization parameter *C* determines the trade-off between the empirical error and the parameter norm, $0 \le \nu \le 1$. A large *C* results in higher penalties for errors and a small *C* assigns smaller ones (i.e. a higher generalization capability). Summarizing, if *C* goes to infinitely large, SVR would not allow the occurrence of any error and would result in a complex and probably overfitted model; whereas a model with C = 0 tolerates a large amount of errors and would be "smoother".

The equations and are solved applying multipliers α, α* in order to build a Lagrange function. Here, the partial derivatives of the function with respect to the primal variables (**w**, *b*, ε, ζ*) have to vanish for optimality (saddle point condition). Replacing the partial derivatives in and yields to the dual problem of Nu-SVR:

$$\min \frac{1}{2}\sum_{i,j}^{l}\left(\alpha^*{}_i - \alpha_i\right) H_{ij}\left(\alpha^*{}_j - \alpha_j\right) - \sum_{i=1}^{l} y_i\left(\alpha^*{}_i - \alpha_i\right) \tag{8}$$

Subject to:

$$\begin{aligned} &\sum_{i,j=1}^{l}\left(\alpha^*{}_i - \alpha_i\right) = 0 \\ &0 \le \sum_{i,j=1}^{l}\left(\alpha^*{}_i + \alpha_i\right) \le C\upsilon \\ &0 \le \alpha^*{}_i, \alpha_i \le \frac{C}{l}, i = 1,...,l \end{aligned} \tag{9}$$

where $H_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$in denotes the matrix of kernel functions. Kernel functions enable the dot product to be performed in high-dimensional feature space using low dimensional space data input without knowing the transformation Φ. All kernel functions must satisfy Mercer's condition that corresponds to the inner product of some feature space. There are several types of Kernels. In this study the better accuracies were achieved using the Radial Basis Function (RBF):

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \left\| \mathbf{x}_i, \mathbf{x}_j \right\|^2\right), \ \gamma > 0 \tag{10}$$

Solving the dual problem in yields the regression function:

$$\hat{m}\_LI = f(\mathbf{x}) = \sum_{k=1}^{\sup vectors} \alpha_k^{(*)} K\left(\mathbf{x}_k, \mathbf{x}_{\sup vectors}\right) + b \tag{11}$$

In the regression function only a reduced number of the Lagrange multipliers are non-zero. Data samples associated with the non-zero coefficients are called support vectors and the final function only depends on those data samples.

The final function is extremely easy to implement in real-time. $\mathbf{x}_{\text{sup } vectors}$ is a fixed matrix with input samples whose Lagrange coefficients are different from zero. $\alpha_k^{(*)}$ and b are calculated in the training process.

To estimate the value for a new sample it is only necessary to introduce the input vector corresponding that sample ($\mathbf{x}_k$) into equation .

To reach the solution expressed in using an RBF Kernel, some parameters have to be predefined: γ,C, ε and ν. The free licenced software implemented to train the regression models is LIBSVM [13] for MATLAB [14]. It automatically sets values for ε and ν. as a trade-off based on the training data. The tuning of the remaining parameters γ and C plays a fundamental role to obtain good models. Their selection is explained at the beginning of next Subsection 4.1.2.

4.1.2. Parameters adjustment, goodness of fit and results.

The most straightforward method to get the appropriate values for γ and C is to fit the data with different combinations of these parameters and then to choose the best result. For that, it is necessary to quantify the goodness of fit of a model given a certain combination of γ and C.

The coefficient of determination r^2 is an effective indicator of the fitting quality. It is based on the quadratic distance between the original signal features and the ones estimated with the regression model. To calculate it, it is necessary to compute the summed square of residuals (SSE), a well-known formula to measure the total deviation of the target values from estimated values:

$$SSE = \sum_{i=1}^{l} (y_i - \hat{y}_i)^2 \tag{12}$$

for $i = 1,...,l$ with $l =$ number of testing samples.

Also it is necessary to calculate the so called sum of squares about the mean (SST):

$$SST = \sum_{i=1}^{l} (y_i - \overline{y}_i)^2 \tag{13}$$

Finally, the r^2 is:

$$r^2 = 1 - \frac{SSE}{SST} \tag{14}$$

The scan was performed for each signal feature and for all the possible combinations of γ (values {0,001; 0,01; 0,05; 0,1; 0,5; 1}) and C (values {0,1; 1; 10; 100; 1000}).

Henceforth, the goodness of fit has been defined as r^2 times 100 (in order to represent its percentage values).

SFMs with the highest Goodness of fit values are selected and listed in Table 3 for each signal feature with each corresponding γ and C values.

4.2. GENETIC PROGRAMMING

In nature, better adapted individuals have higher chances to survive, attract possible partners and breed descendants. Those ones will transmit to the next generation their genes (and therefore their intrinsic characteristics) that allow them to survive until reproduction. Subsequently, their progeny will inherit a combination of these "well-adapted" genes. On the contrary, individuals unable to reproduce will not pass to other generations their intrinsic characteristics.

Genetic Programming (GP) comprises a serial of computational algorithms inspired by natural selection [15,16,17]. Given a problem, a population of possible solutions (i.e. individuals, each one with its own intrinsic characteristics) is created. In order to quantify which individuals are better adapted, a metric called Fitness Function (FF), that measures how accurately each individual solves the problem, is applied. GP assigns a higher possibility to have descendants to individuals with better FF values. Since descendants are created as a combination of "well-fitted" genes, it is expected that newer generations will outperform the former ones.

The iterative process of GP can be summarised in the following steps:

1. Creation of a population of individuals.
2. Fitness evaluation of each individual of the population (using a predefined FF).
3. Selection of parents (a higher selection probability is assigned to those individuals with better FF values).
4. Creation of children as a combination of parents' genes (using genetic operators as *crossover* and *mutation*).
5. Unless an ending condition is satisfied, iterate from step 2, where the new population (created in step 4) is evaluated.

Two genetic operators are normally used to create new individuals (children) in step 4. The *crossover* mixes the genes of the selected parents to create children. Regarding *mutations*, they are unlikely events that occur in nature. In GP these mutations are useful to skip local minima and consist of modifying a gene value in children (with a very low probability of occurrence).

Notice that the whole procedure requires encoding the solutions (individuals) as a set of interchangeable characteristics (genes). GP may adopt an extremely wide number of configurations. For instance, some algorithms take "real-world" approaches, considering several different populations (that interact with their neighbours) instead a single population (as it has been explained above, in the summary of the GP steps). In addition to *crossover* and *mutation*, some GP include extra genetic operators as the *reproductor* that allow direct copies of an individual with promising characteristics into the subsequent generation. The genetic operation *crossover* can be performed by splitting parents' genes in one point (selected randomly or not) or in several points. Also, many formulas and studies have been published regarding the appropriate sizes of the populations and

the number of generations required to reach suitable solutions [18,19,20,21].

Details regarding the state of the art and the extremely wide range of possibilities and variations GP includes are beyond the purposes of this work. Consequently, the selected method, operators and conditions set in this study will be stated in the next Subsection 4.2.1, whereas the explanation of Symbolic Regression must be understood merely as an introduction to a much wider topic of research.

4.2.1. Symbolic regression

When the problem to be solved using GP relies on finding a function that fits a set of data points without making any assumptions about the structure of that function, the genetic programming is often known as Symbolic Regression (SR) [22,23].

In SR, individuals take a tree structure. The leaves of the trees are called nodes. Two classes of nodes exist: operator nodes (that imply an arithmetic operation or mathematical function) and variable nodes (that include the parameters to be considered).

An example tree structure can be observed in Figure 2. The relationship of trees' operators leads to the construction of a function. The trees are read bottom-up and from the left to the right. In the example, at the bottom of the tree, two variables ($mean_{dw/dt}$ and $std(FFT)_{ne}$) are joined together by the operator node division (/). The relationship among these variables is therefore defined as a division ($\frac{mean_{dw/dt}}{std(FFT)_{ne}}$). Following the same procedure, the tree can be read to complete the function: $\frac{mean_{dw/dt}}{std(FFT)_{ne}} - mean_{lp} + e^{mean_{Li}}$.

Individuals of a population (each tree) are created initially randomly. After that, each one of them is evaluated with the FF. For this study the FF has been the r^2 test (in agreement with the goodness of fit determined for SVM-based SFMs). The algorithm might converge into a complex function that accurately fits the training samples. However, it will not necessarily perform adequately with newer data. To control this possible overfitting, it is necessary to evaluate the fitness of each individual also considering its generalization capabilities. For that, a 5-fold cross validation method was applied. The method splits the dataset (training dataset, in this case) randomly into 5 equal size subsamples. Of the 5 subsamples, one is retained for testing the individual, and the remaining 4 subsamples are used as training data. The process is repeated 5 times, with each of the 5 subsamples used once for testing. The 5 results are averaged to get the final fitness. Higher probabilities to be selected as parents are assigned to individuals with better fitness values. Once the selection of the parents has been performed, the genetic operators *crossover* and *mutation* (the latter with a 0,05 % of occurrence probability) are applied. The crossover genetic operation is exemplified in Figure 3. There, sections of Parents' structures are removed. Childs are the result of the combination of randomly selected Parents' structures. Following this methodology, a new population (in which each individual is a combination of Parents with promising FF values) is created after each iteration.

To attain satisfactory tree representations, the closure property must be established. It is a

restriction which implements the protection of the function against inadmissible argument values, *e.g*. negative square roots or divisions by zero. This and other restrictions are included in the open source SR MATLAB toolbox (GPTIPS [23]) that has been used in this study.
Some basic settings must be predefined before launching the SR. They are the number of individuals per population (NIP) and the amounts of generations/iterations (AGI) to evolve. The automatic determination of these parameters is the subject of a wide number of publications [18,19,20,21]. However, in practice, the NIP and iterations usually are problem-dependent. For this study, good results have been obtained setting the NIP = 300 and the AGI = 500.

4.2.2. Results.
In the case of SVR (Subsection 4.1.), to create a SFM for *e.g*. a *mean* signal feature, the 9 *mean* features from the remaining 9 signals (let us call them "regression inputs") are used. Alternatively, SR does not lead *only* to the consecution of a regression model but it also selects the relevant "regression inputs" to include in such model. For instance, if after several iterations of the algorithm most of individuals of a generation perform better (i.e. they have a higher fitness function value) without using any feature coming from the mode lock signal, these signal features will be unlikely to be inherit by the following generations and consequently they will be probably unused at the end. This is the mechanism SR has to eliminate uninformative sources of data and to reduce the complexity of the models. Also, it means that for SR, the SFM (for instance for the $mean_{LI}$) will not be a combination of all the remaining 9 *mean* features ("regression inputs") as in the SVR. Instead, it will be the best combination of all the "regression inputs" it has been fed with, discarding those ones that are unnecessary. Then, it is pointless to restrict so tightly the quantity of "regression inputs" as in SVR since the method automatically discard the uninformative ones.
Therefore, for each one of the SFM based on SR (for instance the $mean_{LI}$) all the remaining signal features (mean *and* std(FFT), instead of only *mean* features in this case for SVR) are initially given as "regression inputs". From them, the ones that are not useful to create a good SFM (according the SR algorithm) are automatically discarded.

To create each SFM for each signal feature one independent run of the SR algorithm is executed. It iterates using the *training dataset* from the creation of the first population (each population set with NIP = 300 individuals) till it reaches the established ending condition (number of generations = 500). The best individual (in terms of fitness function values) from the 500 generations is selected as SFM.

Once this best performing individual has been selected (i.e. the final SFM for a given signal feature), the r^2 test is performed over it (using the *testing dataset*) to get the Goodness of fit values in an analogous way it is calculated for the SVR-based SFMs.
The summary of the results for each signal feature is listed in Table 4. Notice the Goodness of fit values for the LI and dW/dt SFMs are slightly higher than the ones obtained using SVR (see Table 3).

5. IMPLEMENTATION WITH APODIS

Once the SFMs are trained (using either SVR or SR) they can estimate the signal features. In accordance to the real-time scenario simulation, models get as inputs the real-time available signals. Then, their "synthetic" signal features estimations are used as APODIS inputs in replacement of the original ones.

A scheme of the real-time simulated scenario is shown in Figure 4, exemplifying that the LI signal is missing for the SVR-based SFMs. In this case, the SVR-based SFMs models $\hat{m}\_LI$ and the $\hat{std}(FFT)\_LI$ computes the "synthetic" signal features emulating each discharge as it is being generated in real-time. Notice in Figure 4 that APODIS inputs are the features coming from 6 of the 7 usual signals. Instead of applying signal features from LI, they are estimated (using also the 3 extra parameters signal features) and the "synthetic" features are input to APODIS. An analogous procedure has been followed for the SR-based SFMs.

All the discharges of the *testing dataset* have been simulated under these realistic operational conditions. The procedure is completely compatible with a real-time operation and allows verifying how would be the performance of APODIS using the "synthetic" signal features in replacement of the original ones in case they are known to be in error or missing.

6. RESULTS

The results attained with "synthetic" signal features for the LI, Mode Lock and dW/dt in the simulated real-time scenario explained above have been compared with two reference cases: with APODIS operating under standard conditions (*i.e.* using all the signals) and with APODIS simulating the case in which one of the signals is unavailable (*i.e.* without any reliable value during the discharge). The statistics have been computed using the testing dataset that includes a total of 660 JET discharges (see Table 1).

The performances of disruption predictors are usually summarized in plots similar to the ones depicted in Figures 5a, 5b, 5c and 6. These representations are especially informative since they show the disruption prediction rates in relation with the warning time. The warning time is the difference between the disruption time and the alarm time. Therefore, it defines the temporal margin actuators have available to perform avoidance or mitigation actions during the execution of a pulse once an incoming disruption has been foreseen. Points on the curves plotted in Figures 5a, 5b, 5c and 6 describe the accumulated percentage of recognized unintentional disruptions with a warning time equal or higher than the specified in the corresponding logarithmic x-axis.

In addition, Table 5 summarises the false alarms and success rates for APODIS using all the required signals, without one of the signals under study and replacing one of the signals by the "synthetic" estimations carried out by the SVR-based and SR-based SFMs.

SFMs for the LI are accurate in terms of goodness of fit (see Table 3 and Table 4) for both SVR-based and SR-based SFMs. Therefore, it is expected that the prediction results with the original signals and the "synthetic" signal features would be similar. Figures 5a (on the left, they represent

the SVR-based SFMs and the SR-based SFMs on the right) confirm such assumption. Even more, with the "synthetic" signal features used as inputs, APODIS predictions are slightly better (red line) than the ones it attains with the original signals (blue line). This fact is due to statistical fluctuations but confirms that very reliable approximations of LI signal features can be reached. In this case, APODIS results using the SVR-based models trigger less false alarms compared with the SR-based "synthetic" signal features (5,43% *versus* 6,40%, see Table 5). This difference, even being narrow, is expected since the goodness of fit for the LI signal features were higher for the SVR-based SFMs (as it can be seen at Table 3 and Table 4). The relevance of the LI signal can be also noticed in Figures 5a. The black dotted line shows that without it, APODIS rates drop to less than 6% of successful detections.

In agreement with a previous study [4], the evidence that suggested that the Mode Lock amplitude signal is very important for the APODIS predictor has been reinforced. Under the lack of this signal (and therefore its features), the system is not able to operate correctly and it does not trigger any alarm (see the flat black dotted line over zero at the bottom of Figures 5b). For this signal the goodness of fit of the SFMs are significantly lower (see Table 3 and Table 4). APODIS rates with the synthetic features are not as good as the ones with the original signals. They decay from more than 90% (blue lines) to ~59% for the SVR-based SFMs and to ~68% for the SR-based SFMs (red lines). Therefore, detections rates are higher in this case for the SR-based models but with the detrimental factor of triggering a larger amount of false alarms (7,37% vs 3,88% for SVR-based SFMs, see Table 5).

An analogous procedure has been performed for the dW/dt signal. There, the results are excellent even with poor SFMs models (in the sense of goodness of fit, see Tables 3 and 4). From the results shown in Figures 5c, it can be deduced that this signal is important for APODIS but not indispensable for its operation, and even with unreliable synthetic features as inputs (in terms of goodness of fit), APODIS is able to operate as well as under normal conditions.

Finally, the extra testing dataset has been performed in order to know what would have happened in the case of having a SFM for the dW/dt signal during the first 132 discharges of campaign C31. Figures 6 do not show results for APODIS working in normal conditions since in that period the dW/dt was in error and therefore the "true" value of the dW/dt is unknown. With the "synthetic" signal features, the system attains rates over the 92 % of successful detections with a reduced number of false alarms. In general, its performances are similar to the historical ones of the predictor working in standard conditions.

7. DISCUSSION

Due to several reasons, diagnostic systems may fail or be unavailable during experimental campaigns in nuclear fusion devices. When that happens, control and safety systems that depend on the acquired signals may be compromised. In the specific case of APODIS, some signals have proven to be essential for its operation. Without them, APODIS is not able to predict disruptions correctly.

In this article, a potential real-time solution has been developed and tested. It would allow APODIS to operate even in the case that a signal is missing or in error. SFMs have been created off-line and these models are ready to estimate in real-time the value of a signal in error. In this study the SFMs are tested under the assumption of a signal that has been *already* detected to be faulty, as it was the case of the dW/dt at the beginning of campaign C31.

The results have proven that regression models are viable and immediately available patches to overcome problems related to the unavailability of signals. APODIS performs almost perfectly in the case that the signal in failure is either the LI or the dW/dt. For the Mode Lock amplitude, the synthetic signal features improve drastically APODIS responses (prediction rates around between ~59% and ~68%) compared with the ones it obtains under the lack of that parameter (0 % of detected disruptions). However, it could be insufficient having in mind the high standards required for the next step devices. The inclusion of a wider set of real-time plasma parameters could provide a more accurate estimation of the mode lock amplitude. Also, a wider set of optimization techniques that may improve the estimate of the SFMs for that signal could be explored.

For this specific application, the regression models have been developed targeting APODIS requirements. However, the methodology here presented can be similarly followed to estimate the evolution of other plasma parameters (or signal features) if they are considered relevant for control or safety systems in nuclear fusion devices.

ACKNOWLEDGMENTS

This work was partially funded by the Spanish Ministry of Economy and Competitiveness under the Projects No ENE2012-38970-C04-01. This work, supported by the European Communities under the contract of Association between EURATOM/CIEMAT, was carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

[1]. Schuller F.C. 1995. "Disruption in tokamaks". Plasma Physics and Controlled Fusion **37**. A135–62.

[2]. G. A. Rattá, J. Vega, A. Murari, G. Vagliasindi, M.F. Johnson, P.C. de Vries and JET-EFDA Contributors. "An Advanced Disruption Predictor for JET tested in a simulated Real Time Environment" Nuclear Fusion. **50** (2010) 025005 (10pp).

[3]. J. Vega, S. Dormido-Canto, J.M. López, A. Murari, J.M. Ramírez, R. Moreno, M. Ruiz, D. Alves, R. Felton and JET-EFDA Contributors. "Results of the JET real-time disruption predictor in the ITER-like wall campaigns". Fusion Engineering and Design **88** (2013) 1228-1231.

[4]. R. Moreno, J. Vega, A. Murari, S. Dormido-Canto, J.M. López, J.M. Ramírez and JET EFDA Contributors. "Robustness of JET Advanced Predictor of Disruptions (APODIS)". 8th Workshop on Fusion Data Processing, Validation and Analysis. November 4-6, 2013. Ghent, Belgium.

[5]. Rattá G.A., Vega J, Murari A. and Johnson M. 2008. "Feature extraction for improved disruption prediction analysis at JET". Review of Scientific Instruments **79** 10F328

[6]. O. Barana, E. Joffrin, A. Murari and F. Sartori. "Real-time determination of confinement parameters in JET". Fusion Engineering and Design. Volumes 66–68, September 2003, Pages 697–701.

[7]. B. Cannas, A. Fanni, M.K. Zedda, P. Sonato, JET EFDA contributors. "A prediction tool for real-time application in the disruption protection system at JET". Nuclear Fusion, **47** (2007), pp. 1559–1569

[8]. Cortes C. and Vapnik V. 1995. "Support-vector networks". Mach. Learn. 20 273–97

[9]. Gandomi, AH & Alavi, AH, A new multi-gene genetic programming approach to non-linear system modeling. Part II: geotechnical and earthquake engineering problems, Neural Comput & Applic, Springer. 2011.

[10]. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012). "Foundations of Machine Learning". The MIT Press ISBN 9780262018258

[11]. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett. "New support vector algorithms". Neural Computation, **12**, 2000, 1207-1245.

[12]. B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. "Estimating the support of a high-dimensional distribution". Neural Computation, **13**, 2001, 1443-1471.

[13]. Chih-Chung Chang and Chih-Jen Lin. "LIBSVM : a library for support vector machines". ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

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

[15]. Koza J.R. "Genetic Programming", MIT Press, ISBN 0-262-11189-6, 1998.

[16]. Langdon, W.B., Poli, R. (2002), Foundations of Genetic Programming, Springer-Verlag ISBN 3-540-42451-2.

[17]. G.A. Rattá, J. Vega, A. Murari, JET-EFDA Contributors. "Improved feature selection based on genetic algorithms for real time disruption prediction on JET". Fusion Engineering and Design (2012) http://dx.doi.org/10.1016/j.fusengdes.2012.07.002

[18]. Alander, J.T. "On optimal population size of genetic algorithms". Proceedings CompEuro, Computer Systems and Software Engineering, 6th Annual European Computer Conference, 1992, pp.65–70.

[19]. Chan, K.Y.; Aydin, M.E; Fogarty, T.C. "An empirical study on the performance of factorial design based crossover on parametrical problems", Evolutionary Computation, 2004. CEC2004. Congress on, On page(s): 620 - 627 Vol.1 Volume: 1, 19-23 June 2004

[20]. Jun Zhang et al. "Adaptive probabilities of crossover and mutation in genetic algorithms based on clustering technique", Evolutionary Computation, 2004. CEC2004. Congress. 2280 - 2287 Vol.2 Volume: 2, 19-23 June 2004.

[21]. K. A. De Jong, W.M. Spears. "An analysis of the interacting roles of population size and crossover in genetic algorithms". Parallel Problem Solving from Nature. Lecture Notes in Computer Science Volume **496**, 1991, pp 38-47.

[22]. A Murari, I Lupelli, M Gelfusa, P Gaudio. "Non-power law scaling for access to the H-mode in tokamaks via symbolic regression". Nuclear Fusion **53** 043001 doi:10.1088/0029-5515/53/4/043001 2013

[23]. Searson, D.P., Leahy, D.E. & Willis, M.J. "GPTIPS: an open source genetic programming toolbox for multigene symbolic regression". Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 (IMECS 2010), Hong Kong, 17-19 March, 2010.

Period	Number of discharges	Description
~First half of C28-C30	586	Training database
~Second half of C28-C30	660	Testing database
Beginning of C31	132	Extra testing database

Table 1: Description of the database.

Signal number	**Signal**	**Units**
1	Plasma internal inductance	*A*
2	Mode lock amplitude	*T*
3	Stored diamagnetic energy time derivative	*W*
4	Plasma current	*A*
5	Plasma density	m^{-3}
6	Total input power	*W*
7	Radiated power	*W*
8	**Plasma vertical centroid position**	***M***
9	**Poloidal beta**	
10	**Toroidal magnetic field**	***T***

Table 2: Set of plasma signals and their units.

Signal	Feature	Goodness of fit	γ	C
LI	mean	94,06 %	0,5	1
LI	std(FFT)	73,24 %	0,1	1
Mode Lock amplitude	mean	40,60 %	0,001	1
Mode Lock amplitude	std(FFT)	24,28 %	0,5	10
dW/dt	mean	17,31 %	0,5	10
dW/dt	std(FFT)	35,92 %	0,001	1000

Table 3: Goodness of fit for each SFM given the best combination of γ *and* C *parameters.*

Signal	Feature	Goodness of fit
LI	Mean	93,02 %
LI	std(FFT)	65,57 %
Mode Lock amplitude	Mean	50,04 %
Mode Lock amplitude	std(FFT)	40,69 %
dW/dt	Mean	21,06 %
dW/dt	std(FFT)	56,58 %

Table 4: Goodness of fit for the best individuals using SR.

Signal	All signals		SVR-SFM "synthetic" signal features		SR-SFM "synthetic" signal features		Without the signal under simulation	
	False alarms	Success Rates	False alarms	Success Rates	False alarms	Success Rates	False alarms	Success Rates
LI	**6,019% (31/515)**	**91,72% (133/145)**	**5,44% (28/515)**	**91,03% (132/145)**	**6,40% (33/515)**	**91,03% (132/145)**	**0% (0/515)**	**6,20% (9/145)**
ML	**6,019% (31/515)**	**91,72% (133/145)**	**3,88% (20/515)**	**58,62% (85/145)**	**7,38% (38/515)**	**66.20% (96/145)**	**0% (0/515)**	**0% (0/145)**
dW/dt	**6,019% (31/515)**	**91,72% (133/145)**	**4,66% (24/515)**	**91,03% (132/145)**	**6,21% (32/515)**	**89,65% (130/145)**	**1,16% (6/515)**	**79,31% (155/145)**
Extra testing database								
dW/dt	**Not available**	**Not available**	**5,83% (7/120)**	**93,33% (14/15)**	**5,83% (7/120)**	**93,33% (14/15)**	**0% (0/120)**	**0% (0/15)**

Table 5. False alarms and success rates for each simulation. These values are not available for the extra testing dataset since the dW/dt was in failure in that period.

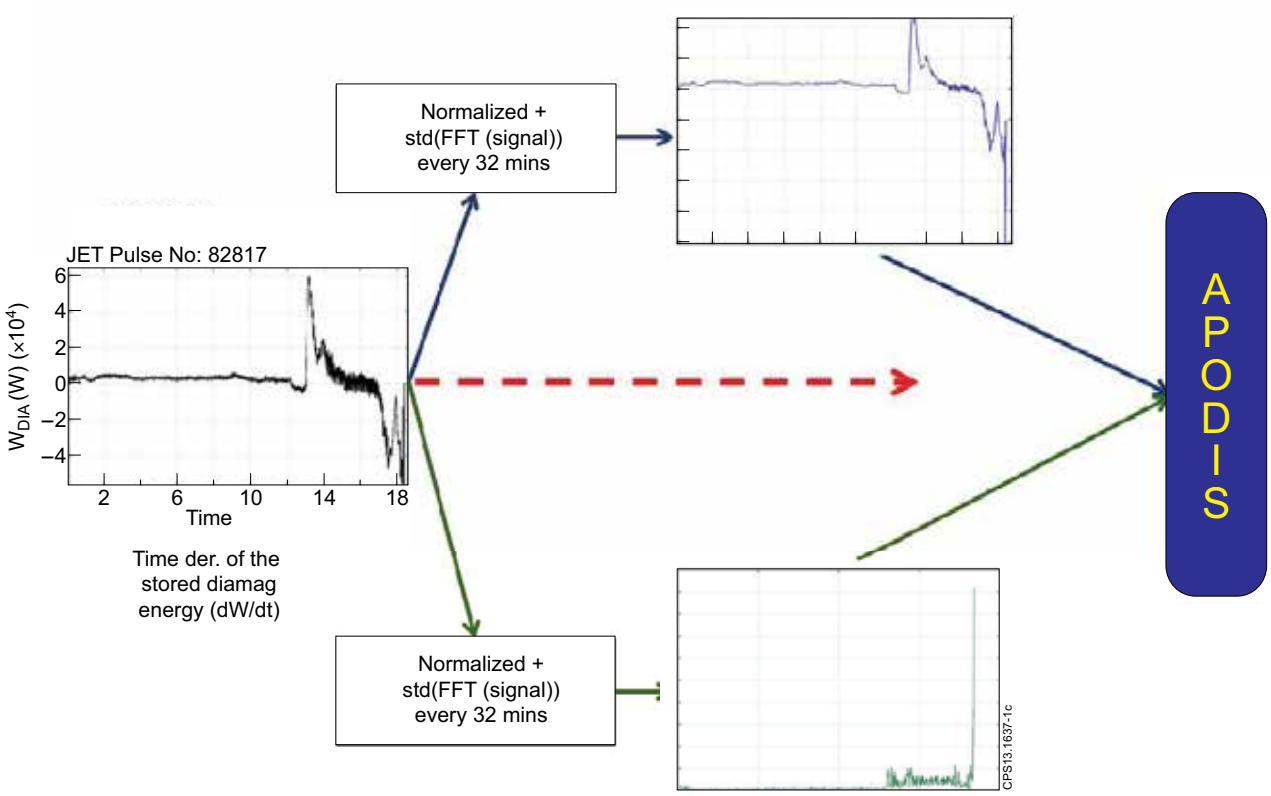

Figure 1: Schematic representation of APODIS inputs for the dW/dt signal.

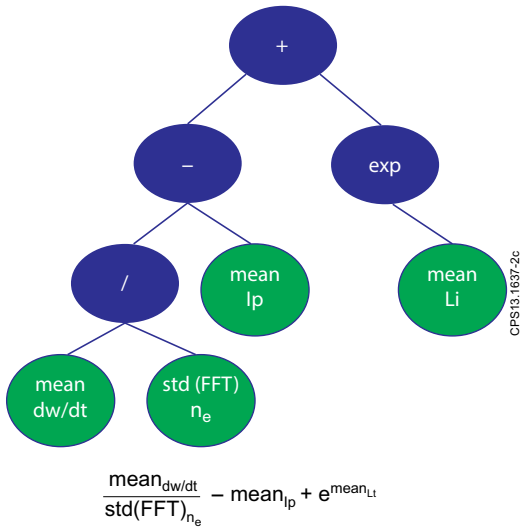

Figure 2:Example of the syntax tree structure. The combination of operator nodes (in blue) and variable nodes (green) creates the function: $mean_{dw/dt} / std(FFT)_{ne} - mean_{Ip} + e^{mean_{Li}}$.

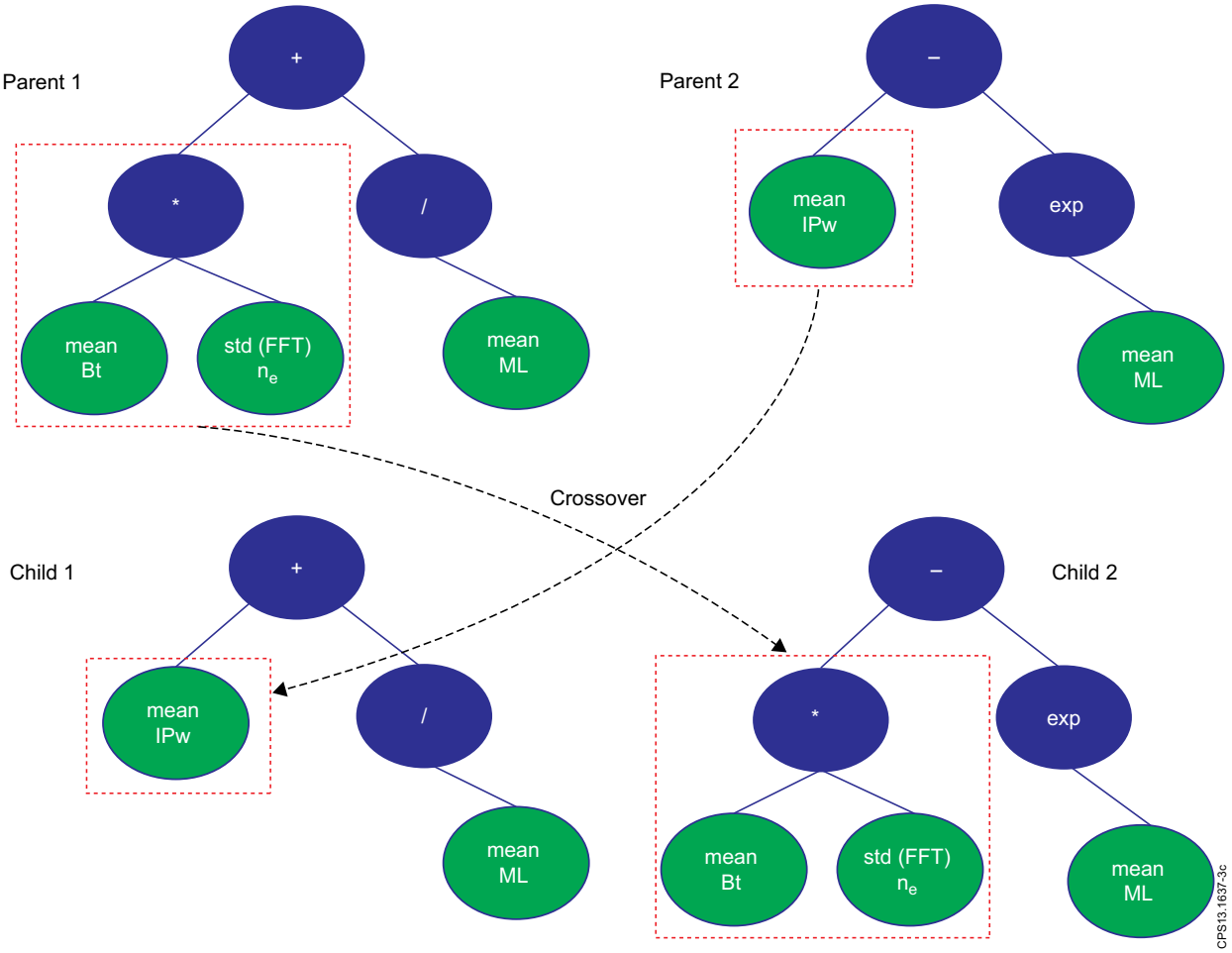

Figure 3: Example of the crossover genetic operation. Sections of trees' structure of the selected parents are swapped to create children.

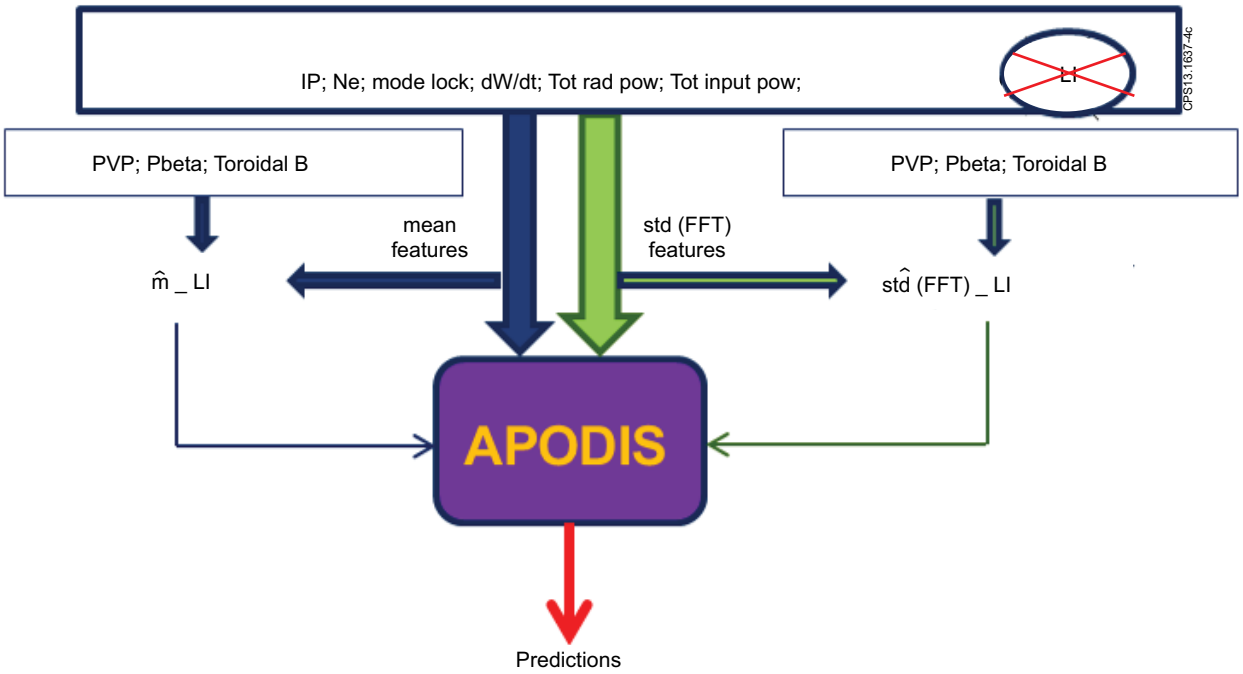

Figure 4: Schematical representation of the emulated real-time scenario under the assumption that LI signal is missing. APODIS uses as inputs all the signals features except the ones coming from the LI. The SFMs estimate LI "synthetic" signal features that are used in replacement of the original ones.

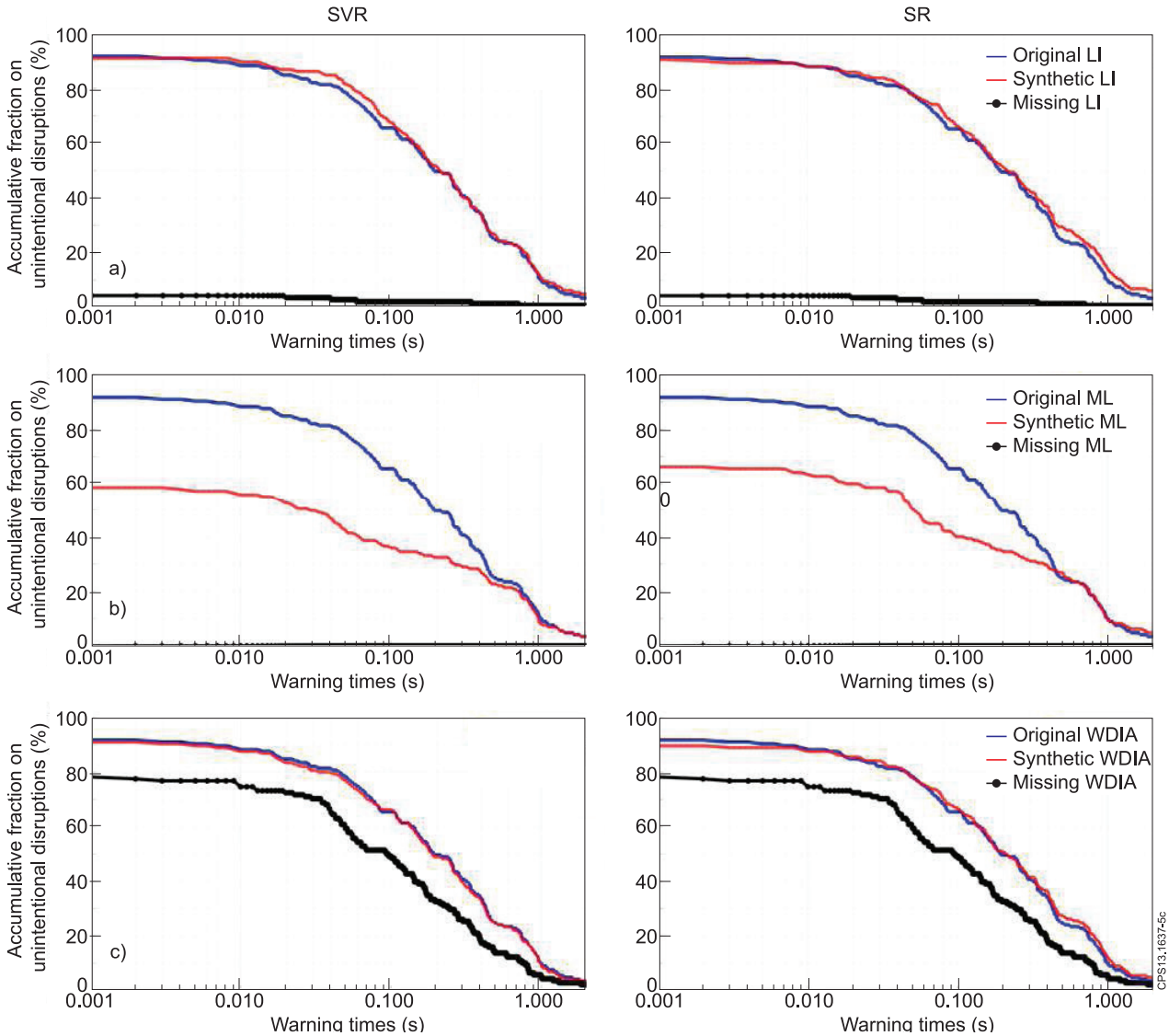

Figure 5: Comparison of APODIS performances using the original signals (blue), using the synthetic features (red) and without the LI (dotted black line). Plots 5a, 5b and 5c address the case of LI, Mode Lock amplitude and dW/dt signals respectively. Figures at left portrait the results of the "synthetic" signals features obtained with SVR-based SFMs and Figures at right the ones attained with SR-based SFMs.

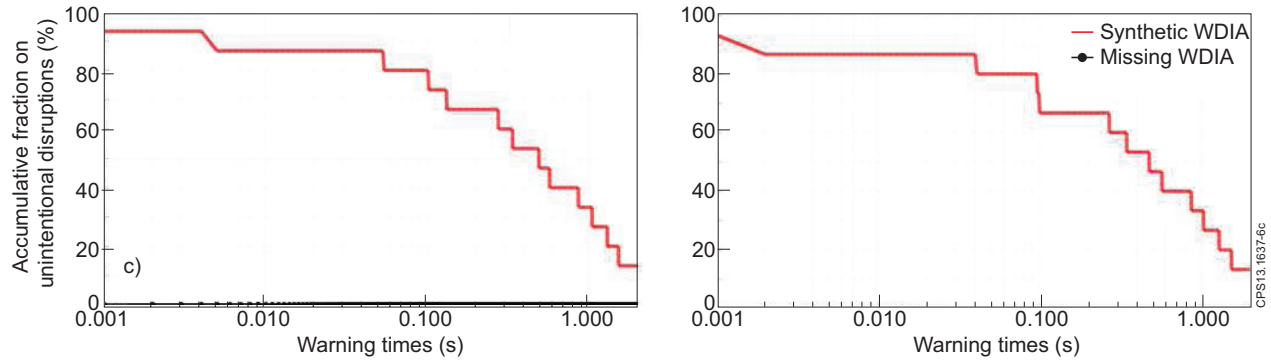

Figure 6: Comparison of performances of APODIS tested with the first 132 discharges of campaign C31 using the synthetic features (at left using SVR-based SFMs and at right SR-based SFMs). At bottom the black circled line represents that in this period in absence of the dW/dt APODIS did not triggered any alarm.