EFDA–JET–PR(12)21

M. Odstrcil, A. Murari, J. Mlynar
and JET EFDA contributors

Comparison of Advanced Machine Learning Tools for Disruption Prediction and Disruption Studies

Comparison of Advanced Machine Learning Tools for Disruption Prediction and Disruption Studies

M. Odstrcil[1], A. Murari[2], J. Mlynar[1]
and JET EFDA contributors*

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

[1]JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK
[2]Consorzio RFX Associazione EURATOM-ENEA per la Fusione, 4-35127 Padova, Italy
** See annex of F. Romanelli et al, "Overview of JET Results",*
(23rd IAEA Fusion Energy Conference, Daejon, Republic of Korea (2010)).

Preprint of Paper to be submitted for publication in
IEEE Transactions on Plasma Science

ABSTRACT

Machine learning tools have been used since a long time to study disruptions and to predict their occurrence. On the other hand, the challenges posed by the quality and quantities of the data available remain substantial. In this paper, methods to optimize the training dataset and the potential of advanced machine learning tools, based on kernels, are explored and assessed. Various alternatives, ranging from appropriate selection of the weights to the inclusion of artificial points, have been investigated in order to improve the quality of the training dataset. Support Vector Machines (SVM), Relevance Vector Machines and one class SVM have been compared. The relative performances of the different approaches are first assessed using synthetic data. Then they are applied to a relatively large database of JET disruptions. It is shown that in terms of final results, the optimization of the training databases proved to be very productive. On the other hand, for the problem of disruption prediction, the two classes SVM remains the most performing machine learning tool that were tested in this contribution.

1. INTRODUCTION

Disruptions are sudden and irreversible losses of plasma confinement. The fast release of energy during the disruptions may pose serious problems to plasma facing component surfaces due to thermal quench causing intensive heat flux. Furthermore, eddy and halo currents, induced during the current quench can generate electromagnetic forces on the tokamak chamber. Finally, runaways electrons generated during the current quench can produce localized heat damage or erosion. Disruptions are unavoidable in tokamak plasmas and although the existing machines can withstand many disruptions, in future machines such as ITER, the disruptions can cause severe damage and significantly reduce the lifetime of machine components.

Disruption prediction is a difficult task from the point of view of machine learning. The used tools require a huge number of learning points but despite this, the number of disruptive discharges, which are used to evaluate the success rate, is very low. This fact can make it difficult to choose the best model. Moreover, the training classes are significantly unbalanced, usually less than $10^{-3}\%$ points belong to the disruptive class. Also, most of the nondisruptive points/discharges are very easy to identify and add no new information to the model, therefore the informative value in the data is very sparse. Finally, the data contain a high fraction of outliers. The disruptions themselves are outliers compared to the majority of the points, however, there exist many nondisruptive discharges that can also significantly exceed the \stable" region. Therefore, although the fraction of the nondisruptive outliers can be very small, it can significantly affect the predictions.

In this paper, two different steps, to alleviate the aforementioned problems, are proposed. First of all, some methods to improve the quality of the training data are introduced. These are very general steps that can be implemented independently from the machine learning tools used. The second step relies on an investigation of more advanced learning tools to see whether they can achieve better performance compared to the traditional Support Vector Machines, used as the reference technique.

In Section 2, possible manipulations of the original datasets to reduce the mentioned problems are presented. Section 3 describes the used learning machines. Numerical examples are provided in Section 4. Finally, the results of the proposed improvements are analysed in Section 5 using a database of discharges from tokamak JET.

2. IMPROVEMENTS IN THE SELECTION OF THE TRAINING DATABASE

The problems of the unbalanced dataset, sparse information in the data and huge number of points are interlinked. The unbalance can be removed if the relative weight of the disruptive class is increased. However, decreasing the weight of the nondisruptive class by removing unimportant data proves to be a more efficient way. In the first step, when the probability distribution over the training points is unknown, the least important points are expected to lie near to the median value. Therefore, the majority of points (disruptive and nondisruptive), whose values are in range from -quantile to (1-α)-quantile in all dimensions, can be removed. Using a value of equal to 10^{-3}, the dataset is reduced to less than 0.5% of the original size without a noticeable effect on the results. In this way, the data of the disruptive discharges with no clear precursors are automatically removed and the number of the Support Vectors (SV) (see the following section) is decreased.

Once the learning machines have been trained, the relevance of the various points can be estimated from the predicted probability density in order to perform a more precise selection of the training set for the next iteration. After this second screening, the population of the disruptive samples in the resulting training dataset reached typically more than 5% (a significant increase compared to the original 10^{-3}%).

The second issue is posed by the outliers in the nondisruptive data. To alleviate the problem, the following two approaches have been tested:

- Iterative removal of the worst outliers
- Addition of artificial points

The iterative removal procedure can be applied only in the second step of the training process, when the most outlying points or discharges can be identified and removed from the training set. Usually, only around 500 points were removed from the original training database size of more than $6 \cdot 10^6$ points.

The artificial points can be added to the “clearly disruptive” areas such as very strong locked modes or plasma position oscillations. Generally, the artificial points can be based on a human selected disruptive threshold (training prior) for each dimension. However, the disadvantage is that number of the support vectors for the two-class SVM can significantly increase (see Section 3). This problem is avoided in this contribution by increasing the weight ω_o of the artificial disruptive points (eq.3) by five orders of magnitude.

All these improvements have been applied only to the training set; the testing set was kept unchanged.

3. OVERVIEW OF THE APPLIED MACHINE LEARNING TOOLS

The previous improvements are rather general; therefore they can be applied to a wide range of learning tools. We have tested several kernel based methods: two class Support Vector Machine (SVM), one class SVM, Relevance Vector Machine (RVM). The Gaussian kernel was used in all these cases and moreover two class support vector machine was tested also with the linear kernel.

3.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is a simple but powerful pattern recognition (classification) kernel based method introduced by Vapnik [1]. SVM, basically, searches for a hyperplane that separates two groups of data points. The SVM attempts to minimize the error on training data while the margin between groups in the feature space is maximized. The separating boundary of the kernel based methods can be very flexible with a non-linear mapping of the hyperplane to the feature space. Moreover, the maximum margin principle leads to good generalization.

The maximization of the distance from the boundary can be reformulated as the minimization of a term $\|w\|^2$. If we denote the set of N target values as $\{t_i\} \in 1...N$ and the corresponding input vectors as $\{x_i\}$, the technique results in the following quadratic optimization problems

$$\min_{w} \frac{1}{2} ||w||^2 \tag{1}$$

with constrains

$$t_i(w^T x_i + b) \geq 1 \quad i = 1,..., N \tag{2}$$

A standard method to allow data points exceeding the boundary with some penalty is called *soft margins*. New *slack variables* must be introduced to measure this overlapping, as well as a constant C that controls the trade-off between the data overlapping the boundary and the size of the margins. Moreover, the trade-off can be selected artificially for each point if an individual weighting $\omega_i > 0$ is applied

$$\min_{w} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \omega_i \xi_i \tag{3}$$

with constrains

$$t_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1,..., N, \quad \xi_i \geq 0 \tag{4}$$

These equations can be directly optimized. However, they can be transformed to their dual representation using the so-called kernel trick [1]

$$L(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m K(x_n, x_m)$$

$$0 \leq a_n \leq \omega_n C$$

where L is a minimized Lagrangian, an are searched dual parameters, t_n are the target values normalized to the set {–1;1}, input vectors x_n and $K(x_n; x_m)$ is the kernel.

The Lagrangian L is minimized with respect to the transformed constrains

$$\sum_{n=1}^{N} a_n t_n = 0$$

The prediction output of the SVM is proportional to the distance from the boundary

$$\sum_{n=1}^{N} a_n t_n K(\mathrm{x}_n, \mathrm{x})$$

Only the vectors with nonzero parameters a_n are used for the prediction and these vectors are called support vectors. Support vectors points always lie at the boundary or outside the boundary. The rest of the points are ignored after the learning phase, therefore SVM leads to a sparse model.

In this paper, the implementation of C-SVM from the LibSVM library [2] is used.

3.2. ONE CLASS SVM

Although the SVM algorithm is essentially a two class classier, Scholkopf proposed [3] a modification for one class only. This method is usually used for identification of outliers or novelty detection. The basic idea of this algorithm is to find a hyperplane that separates the majority of the data points from the outliers such that the fraction of training points getting beyond the boundary is equal to v. This can be done if the origin after kernel transformation is treated as the only member of the second class and if the v-SVM [4] technique is used. The v-SVM method is very similar to the C-SVM from the previous section, except the fact that the parameter C is replaced by a parameter $v \in [0, 1]$. This parameter determines the number of the support vectors lying on the wrong side of the hyperplane.

The one class SVM selects a separating hyperplane so that the majority of the points is lying inside the boundary and only a minority of the points is lying outside.

The following quadratic optimization is used to separate the data set from the origin

$$\min_{\omega i \xi, \rho} \frac{1}{2} ||\mathrm{w}|| + \frac{1}{\upsilon\iota} \sum_i \xi_i - \rho$$

with constraints

$$\mathrm{w}x_i \geq \rho - \xi_{i,} \qquad \xi_i \geq 0$$

This quadratic problem is then transformed into the dual problem in the same way as the SVM. The final decision rule (distance from the boundary) is

$$f(\mathrm{x}) = \sum \alpha_i K(\mathrm{x}_i, \mathrm{x}) - \rho$$

When the parameter v goes to zero, the boundary should behave as a hard margin, therefore no outliers in the training set are allowed.

The main advantage of this algorithm is the possibility to use only one class for training or a very limited size of the second class for cross-validation to estimate the free parameters of the model: v and kernel parameters. However, in the case of plasma disruptions the results depends only weakly on v. Therefore, only the kernel parameter must be estimated. Furthermore, this algorithm can be used for Novelty detection in order to recognize shots very distant from the training data in the operational space.

In this paper, the implementation of One Class SVM from the LibSVM library [2] is used.

3.3 RELEVANCE VECTOR MACHINE (RVM)

In spite of the state-of-the-art results in many tasks where the SVM was used, this tool suffers from several disadvantages:

- The first disadvantage is that the predictions are not probabilistic. Although some attempts to add an estimation of the prediction probability exist [5, 6, 7], the results are not completely well founded theoretically.
- SVM uses an unnecessarily high number of support vectors and although the resulting model is sparse, the number of the support vectors can be a significant fraction of the training set. Moreover, the number of SV usually grows linearly with the number of training points. To alleviate this, some postprocessing methods exist that can decrease the number of SV [8].
- No straightforward and universal way exists on how to determine C and kernel parameters for a nonlinear kernel.
- The kernel function $K(\mathrm{x_i}, \mathrm{x})$ must satisfy Mercer's condition [1].

Relevance Vector Machine algorithm was introduced by Tipping [9] mainly to overcome the disadvantage of SVM, in particular the no native probability output and the issue of additional model parameters C and . Another advantage is that the RVM model is usually much sparser than the SVM model and thus the predictions are faster. The derivation of the RVM can be found in [9], an introduction to the idea is given in the following.

The final equation for the RVM prediction is identical to the SVM

$$y(\mathrm{x}, \mathrm{w}) = \sum_{i=1}^{M} \omega_m K(\mathrm{x}_i, \mathrm{x}) + b \tag{5}$$

where **w** is vector of weights and b is the intercept.

However, the Bernoulli distribution should be adopted to obtain the probabilistic prediction $p(\mathrm{t}|\mathrm{x})$ because only values 0 and 1 are possible. Therefore, the logistic sigmoid function $\sigma(y) = 1/(1 + \mathrm{e}^{-\mathrm{y}})$ must be applied to transform the regression eq. 5, which is linear in coefficients, to a fully nonlinear probabilistic function. According to the definition of the Bernoulli distribution, the conditional probability for N points under an assumption of known weights **w** can be written as

$$p(\mathrm{t}|\mathrm{w}) = \prod_{n=1}^{N} \sigma\{y(x_n, \mathrm{w})\}^{t_n} [1 - \sigma\{y(x_n, \mathrm{w})\}]^{1 - t_n}$$

for targets $t \in \{0, 1\}$. Maximum likelihood estimation of the weights leads to overfitting. The maximum likelihood estimation of the weights leads to overfitting. In Ref. [9], it was suggested to introduce prior constraints on the weights **w** penalizing the complexity of the model. Prior distributions in combination with the Bayes rule were introduced to avoid overfitting:

$$p(\mathrm{w}|\alpha) = \prod_{n=1}^{N} \sqrt{\frac{\alpha_i}{2\pi}} \exp\left(-\frac{\alpha_i \omega_i^2}{2}\right)$$

where α denotes hyperparameters that control the distribution of the associated weights **w**.

These hyperparameters, separate for each weight, present the most important property of the RVM algorithm.

Given the prior, new predictions are made using the so-called predictive distribution:

$$p(\mathrm{t}^*|\mathrm{t}) = \int p(\mathrm{t}^*|\mathrm{w}, \alpha) p(\mathrm{w}, \alpha|\mathrm{t})\, d\mathrm{w} d\alpha \tag{6}$$

However, as it is the case for many Bayes problems, this integral cannot be solved analytically. The posterior $p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t})$ is not known. Instead, it can be decomposed as

$$p(\mathrm{w}\,\alpha|\mathrm{t}) = p(\mathrm{w}|\mathrm{t}, \alpha) p(\alpha|\mathrm{t})$$

The posterior over the weights $p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t})$ is according to the Bayes rule equal to

$$p(\mathrm{w}|\mathrm{t}, \alpha) = \frac{p(\mathrm{t}|\mathrm{w}) p(\mathrm{w}|\alpha)}{\int p(\mathrm{t}|\mathrm{w}) p(\mathrm{w}|\alpha)\, \mathrm{dw}} \tag{7}$$

The integral in the denominator of the previous equation has no analytical solution for the Bernoulli distribution, therefore the Laplace approximation should be applied.

The second term in eq. (6) $p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t})$ is approximated by the most probable point estimated values. The distribution $p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t})$ is therefore approximated by (MP). The hyperparameters are obtained from the type II maximum likelihood method as shown in [10].

During the optimization, most of the α_i values get large and the corresponding weights **w** are almost infinitely peaked around zero. As a result, and, because the corresponding kernels are pruned, the models become very sparse. This is called ARD prior – Automatic Relevance Determination. The remaining vectors are called relevance vectors and, contrary to the SVM, these vectors present the most representative points of both groups. The optimization continues until the change in the $\boldsymbol{\alpha}$ vector is below a certain threshold or until a maximum number of iteration is reached.

Despite the described advantages of the RVM, this algorithm also suffers from several disadvantages. Firstly, the RVM optimization can reach a local maximum on the contrary to the

SVM optimization where the global optimum is always guaranteed because the optimized function is convex. Therefore, the RVM can result in a significantly worse model, however it is possible to detect these corrupted results and remove them. The computational complexity and high memory demands presents another disadvantage. The memory limitation restricts the training set to less than 104 points for large kernel sizes, whereas the LibSVM algorithm can be effectively computed with more than 105 points thanks to the implemented heuristic algorithms and caching [2].

In this work, the implementation of RVM in MatLab was used, i.e.SparseBayes 2.0 [11], with a few improvements in speed and memory usage based on the sparse matrices.

4. NUMERICAL EXAMPLES

All the three algorithms introduced in the previous section have been tested on a standard dataset [5]. The classification data contain 200 points, sampled from a 3-component Gaussian mixture in two dimensional space. The parameters of the Gaussian mixture model are:

$$p_1 = 0.5;\ p_2 = 0.25;\ p_3 = 0.2{:}5$$

$$\mu_1 = [0.0, -0.1]\ \mu_2 = [1.0,\ 1.0]\ \ \mu_3 = [1.0,\ -1.0]$$

$$\Sigma_1 = \begin{pmatrix} 0.625 & -0.217 \\ -0.217 & 0.875 \end{pmatrix}\ \ \Sigma_2 = \begin{pmatrix} 0.224 & -0.137 \\ -0.137 & 0.976 \end{pmatrix}\ \ \Sigma_3 = \begin{pmatrix} 0.238 & -0.157 \\ -0.157 & 0.413 \end{pmatrix}$$

where p_i denotes prior probabilities, μ_i is center and Σ_i is covariance matrix of the i-th Gaussian. The first Gaussian corresponds to the first class and the second and third corresponds to the second class of points.

The results are shown in Figs.1, 2, 3. Points in circles correspond to the support/relevance vectors and the colour denotes the class of the point.

The probability predictions for this simple dataset show very similar results despite the very different nature of the techniques.

The error rates of the two-class SVM and the RVM algorithm are equal (22–23%). In spite of the missing information, the error rate of the one class SVM is only slightly higher (27%). In this example the highest number of support vectors is used by the two-class SVM (Fig.1). However, note that the one-class SVM (Fig 2) uses almost all the training points as SV, therefore the number of support vectors can be very high in the case of unbalanced training classes. Finally, the RVM algorithm uses only 3 relevance vectors.

5. DISRUPTIONS FROM JET

The above introduced algorithms and their improvements have been tested to estimate the performance in the plasma disruption prediction. The used data originate from the tokamak JET and the disruption events were manually verified [12]. Campaigns C19-C22 from February 2007 to August 2008 were used for training and similar campaigns C24,C25,C27 from October 2008 to October 2009 were used for testing. During these campaigns JET still had a carbon wall.

Campaign C26 was omitted becuase the quality of the measurements was lower and there was a systematic use of the EFCC which alter the results. The training set contains 1246 nondisruptive and 178 disruptive discharges and the set for testing contains 1963 nondisruptive discharges and 143 disruptive discharges. All dimensions were linearly interpolated to 10 ms time resolution with a node in the time of the disruption for disruptive pulses. Sampling time 10 ms was selected as a compromise between sufficient time resolution and the time resolution saved in the JET database. Moreover, the relatively low time resolution was selected to avoid unrealistic improvement in score. The improvement appears when a variation sensitive preprocessing is used and it is caused by a change in sampling frequency of plasma current and locked modes signals approximately 300 ms before the disruption.

These inputs have been selected using Forward Feature Selection (FFS) method from a large set of variables with three preprocessings: mean, time derivative and time variation (abs(diff)) of the signals. The disruptions from campaigns C15 to C27 were used for the feature selection –440 unintended disruptive shots. Further 4110 nondisruptive discharges from similar campaigns C19–C22 and C24–C27 were added to the FFS dataset. Each combination of inputs was solved 10 times with different training/validation set in order to estimate uncertainties and select the best inputs. The optimal set of data inputs is listed in Tab.1. The overall score was evaluated in the following way: 1–(%FA+%MA+%EAdisruptivity) – False Alarms (FA) are nondisruptive discharges where at least one point is recognized as disruptive, Missed Alarms (MA) are disruptive discharges that were not identified as such more than 30 ms before the disruption and Early Alarms (EA) are disruptive discharges that would trigger an alarm more than 1s before the disruption. The disruptive class contains only 150ms of signals before the disruption.

5.1 PERFORMANCE ASSESSMENT

Notice that the missed alarms rate and false alarms rate in the disruption prediction are not proportional to the number of misclassified training points. MA and FA are function of the number of misclassified discharges that are determined as sequences of decisions whereas the introduced learning machine tools treat each decision separately and minimize the simple zero-one (misclassification) error. In this contribution, an optimal threshold on the learning machines output has been determined in order to maximize the success score in terms of correctly identified discharges. The optimal threshold search has been implemented even during the cross-validation, not only in the last step before the score evaluation.

6. RESULTS

Each algorithm was trained 20 times with a different training and validation discharges that were randomly chosen from the total training set in order to estimate the variance of the resulting scores using cross-validation. However, this could not remove the possible bias caused by selection of the testing set based on the JET campaigns C24,C25,C27. It would need more similar campaigns

to suppress this bias in the score. The learning machines were trained on two datasets: the first training set (full database) contained all discharges and the second (small database) contained only 10% of discharges that have been randomly but unambiguously selected from the full database. The analysis of a small database has been introduced to assess the impact of a limited number of examples on the various techniques. This aspect is particularly relevant for the training of machine learning tools at the beginning of new campaigns or at the start of the operation of new devices.

The final results are shown in Figs 4, 5. It can be seen that the iterative removing of outliers significantly decreases the variance of the scores and also increases the mean and median of the scores. However, the maximal achieved scores can be similar to the case of the entire database. This is because the random combinations of discharges used as training set can by chance include a low number of outlying discharges. Further, the artificial data avoids overfitting in the case of two class SVM. It is more significant for the full training database (Fig.4) because mere 10% of the training discharges was used for training in the Fig.5 and this avoided the over-fitting in the training score. Another relevant observation is that the artificial data increases the score for the RBF kernel based methods while the score for the linear kernel SVM is less than 0.5. This is caused by the increased nonlinear separability of the resulting training set. A similar problem can be seen also in Fig.5 for RVM that behaves more like a linear kernel with a low number of discharges.

It should be noticed that the scores for the training set are low compared to the testing set. It is caused by the fact that the score is particularly determined by the disruption type in the selected campaigns while the learning machine selection has only a secondary effect. However, the number of missed disruptions was not statistically significant to deduce any further analysis.

Finally, the results of the methods significantly differ in the number of the support or relevance vectors. The two-class SVM uses around 1000 support vectors, one class SVM uses around 10000 support vectors and RVM uses less than 50 relevance vectors. The linear SVM needs around 3000 vectors, however the prediction can be done using N+ 1 constants, where N is number of the input dimensions.

The time evolution of the detected disruptions is presented in Fig.6. The evolutions for both datasets are very similar because the basic shape is determined by the precursors in the data. The most important result is that the SVM has clearly better performance than the JPS in starting from 100ms before the disruptions. The score evolutions for the other learning machines are very similar. It should be also noticed that even the worst success rates of a properly trained machine learning tool usually exceed 80%. Therefore, the majority of the disruptions can be recognized routinely. Up to 90% of the disruptions can be properly recognized with a sufficiently large database; however, approximately 5% of the disruptions in the test database were never predicted early enough using the input signals listed in Tab.1. Figure 7 shows the performance of the two classes SVM particularized for the various campaigns. The performance of the classier is quite consistent and the performance remains very constant in time. Indeed the campaigns in which the predictor scores are particularly low were affected by some known problems either in the signals of the operation of the device (ie.

during campaign C23 the Error Field Correction Coils were extensively used). Campaigns C19–C22 were used for training and campaigns C24,C25,C27 were used for testing.

CONCLUSIONS

In this paper a series of methods to improve the success rates of disruption predictions have been implemented and tested. They belong to two main classes of approaches: a) the improvement of the training datasets b) the increased sophistication of the learning techniques of the learning systems. The various improvements have been tested using a quite large database of JET disruptions.

With regard to the first approach, improving the datasets, by reducing the outliers and in general by increasing the percentage of relevant information, has significant and consistent benefits on the final performance. These improvements are of course less pronounced in the case of small databases because these datasets are inherently less sparse and therefore all the measures taken, which basically are aimed at reducing the sparsity of the data, are less effective.

With regard to the machine learning tools, the typically used two class SVM gives consistently the best results. Since this is also the learning technique which is easier to implement and routinely used in many other applications, there is no evident reason not to prefer it.

An important remark is the fact that even if the objective of the present analysis is the comparison of the various techniques more than the achievement of high performance in terms of success rate, the machine learning tools can easily achieve a success rate well in excess of 80% and can typically reach 90%. The next issues to attack in future works are therefore first the developments of techniques to learn from a limited number of examples, a situation in which the present methods show a certain weakness. Second, the same approaches should be applied to the more difficult task of not only predicting the incoming disruption but also the type, so that more specific remedial actions can be undertaken.

ACKNOWLEDGMENTS

This work was supported by EURATOM and 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]. V.N. Vapnik. The nature of statistical learning theory. Springer Verlag, 2000.

[2]. C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), **2**(3):27, 2011.

[3]. B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. Neural computation, **13**(7):1443{1471, 2001.

[4]. B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. Neural computation, **12**(5):1207–1245, 2000.

[5]. C.M. Bishop and SpringerLink (Online service). Pattern recognition and machine learning, volume **4**. Springer New York, 2006.

[6]. J. Platt. Probabilistic outputs for support vector machines. Bartlett P. Schoelkopf B. Schurmans D. Smola, AJ, editor, Advances in Large Margin Classiers, page 61–74.

[7]. H.T. Lin, C.J. Lin, and R.C. Weng. A note on Platt's probabilistic outputs for support vector machines. Machine Learning, **68**(3):267–276, 2007.

[8]. C.J.C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. Advances in neural information processing systems, **9**:375–381, 1997.

[9]. M.E. Tipping. Sparse Bayesian learning and the relevance vector machine. The Journal of Machine Learning Research, **1**:211–244, 2001.

[10]. M.E. Tipping. Bayesian inference: An introduction to principles and practice in machine learning. Advanced lectures on machine Learning, page 41–62, 2004.

[11]. M.E. Tipping. An Efficient Matlab Implementation of the Sparse Bayesian Modelling Algorithm (Version 2.0). 2009.

[12]. P.C. de Vries, MF Johnson, and I. Segui. Statistical analysis of disruptions in JET. Nuclear Fusion, **49**:055011, 2009.

Diagnostics
Mock lock amplitude
Plasma internal inductance
Beta normalized
Greenwald density ratio
Inverse safety factor
Total radiated power ratio
Plasma current derivative
Plasma verticle centroid position variation

Table 1: The complete list of the JET database outputs used in this work.

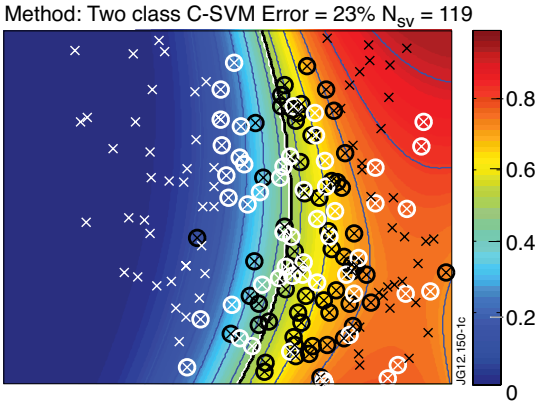

Figure 1: An example of the two class SVM algorithm with the Gaussian kernel and a probability estimation [7]. The colour code corresponds to the estimated probability.

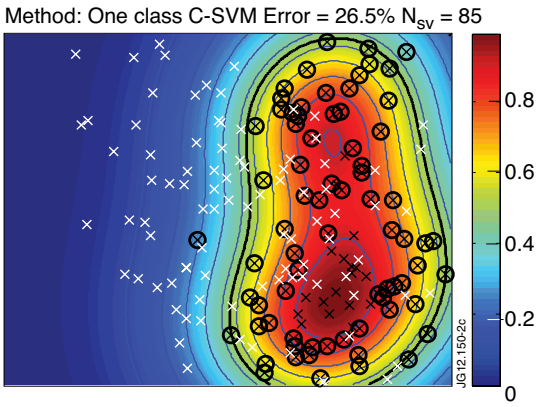

Figure 2: An example of the one class SVM algorithm with the Gaussian kernel. The colour code corresponds to distance from the boundary renormalized from 0 to 1 and with the threshold shifted to 0.5.

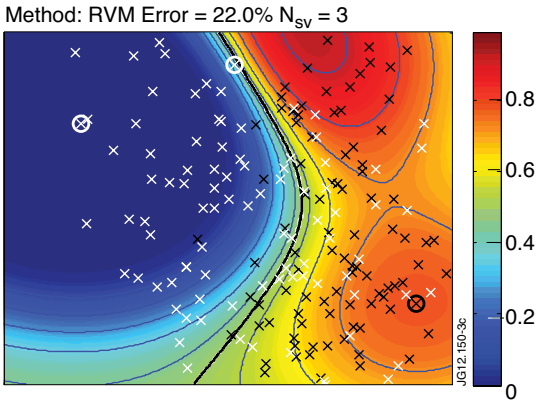

Figure 3: An example of the RVM algorithm with the Gaussian kernel. The colour code corresponds to the estimated probability.

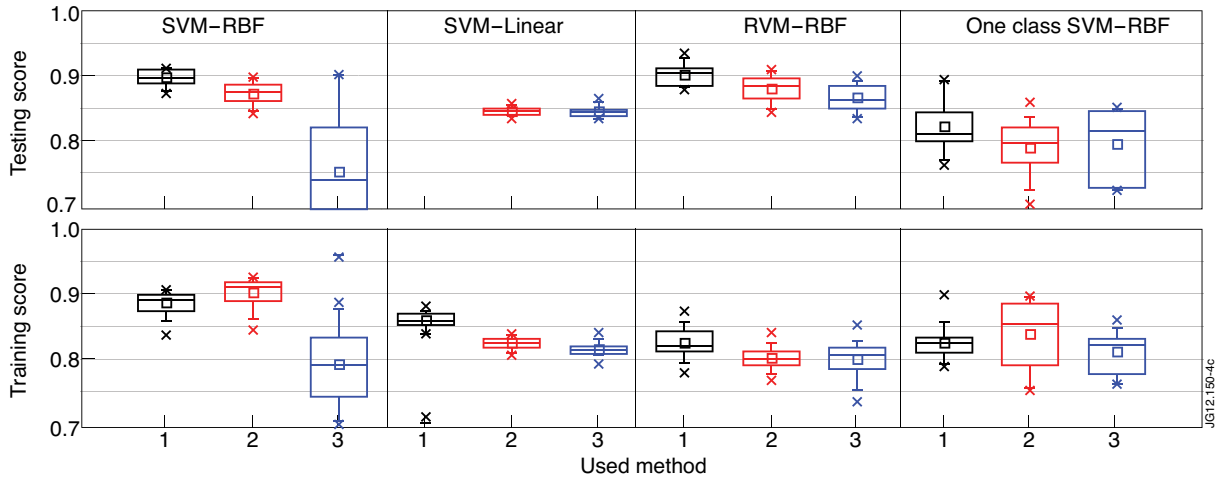

Figure 4: Overall results for all tested algorithms using the complete training set. The column cells contain the scores of following algorithms: two class Support Vector Machines (SVM) with RBF kernel; two class SVM with linear kernel; Relevance Vector Machines (RVM), one class SVM. The first row contains the results for the test set, the second shows the scores for the training set. Finally, in each cell the first box plot contains the following improvements of the training set: 1 – artificial data and iterative removal of the worst outliers (left, black), 2 – removal of the worst outliers (middle, red), 3 – none (right, blue).

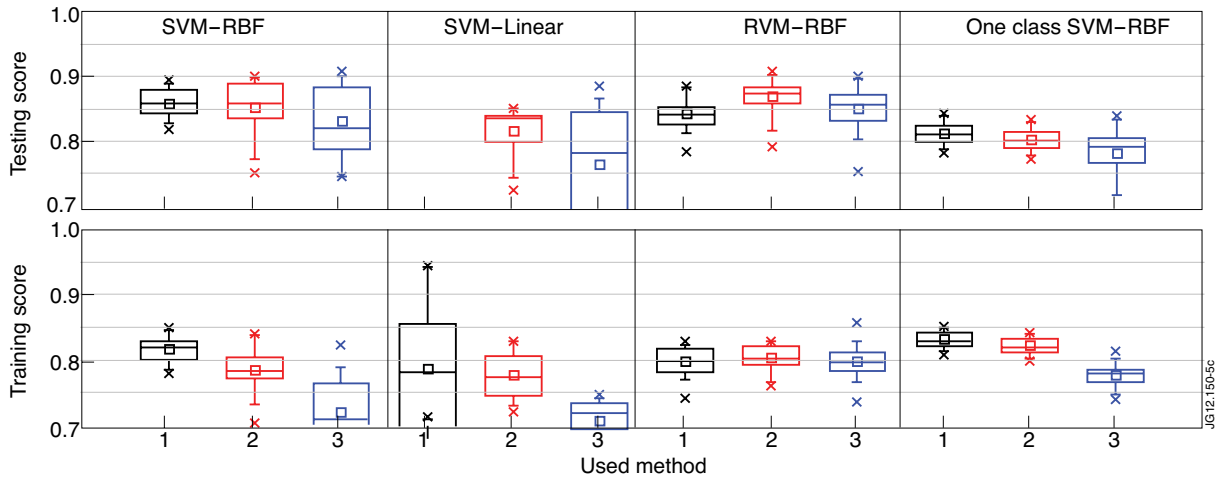

Figure 5: Overall results for all tested algorithms using the small training set. However, the showed training score was evaluated using the full training database. The arrangement of plots is the same as in the Fig.4.

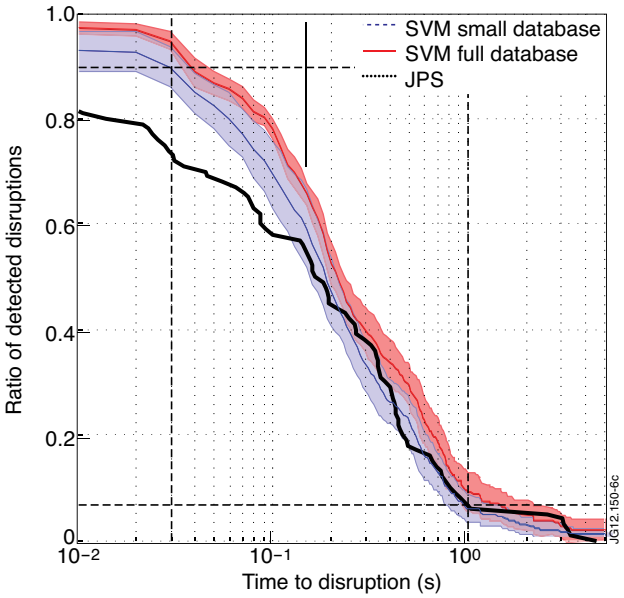

Figure 6: Time evolution of disruption prediction ratio for two-class SVM with iterative removal of outliers and artificial points. The red curve SVM was trained with the full database while the blue curve SVM with a small database (10% of discharges). Time points corresponding to 30ms and 1s before the disruption are identified by dashed lines.

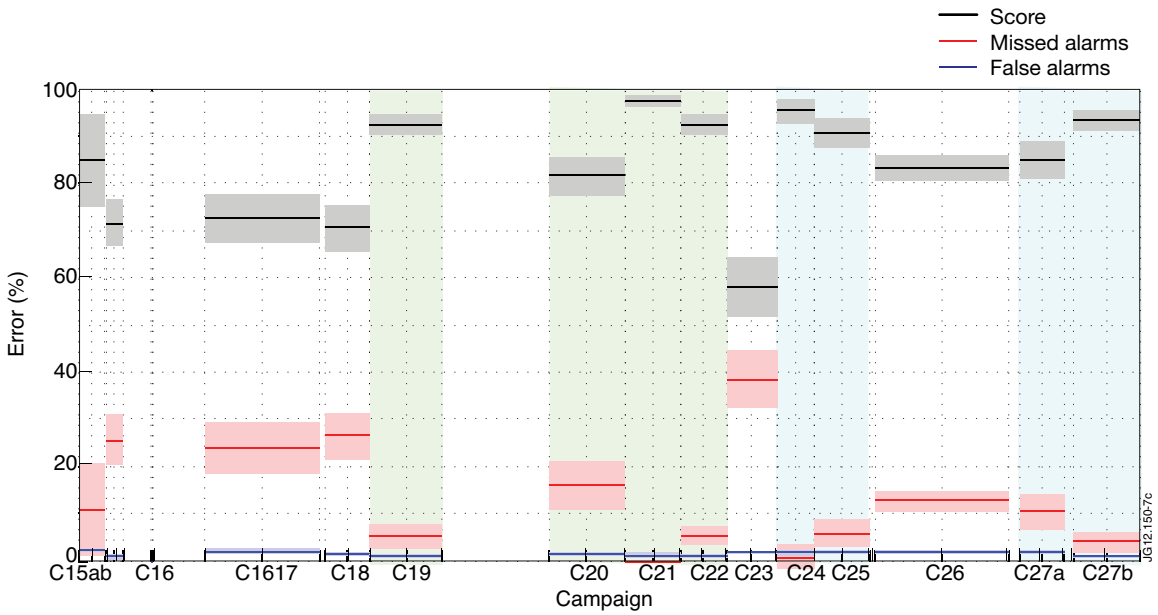

Figure 7: The total score for the optimized two class SVM in the JET campaigns C15 to C27. This figure illustrates the variance in score for different campaigns. Campaigns C19-C22 were used for training and campaigns C24,C25,C27 were used for testing.