CNN disruption predictor at JET: early versus late data fusion approach

E. Aymerich[1], G. Sias[1], F. Pisano[1], B. Cannas[1], A. Fanni[1] and the JET Contributors*

[1]Dept. of Electrical and Electronic Engineering-University of Cagliari, Cagliari, Italy.

*See the author list of Overview of JET results for optimising ITER operation by J. Mailloux et al. to be published in Nuclear Fusion Special issue: Overview and Summary Papers from the 28th Fusion Energy Conference (Nice, France, 10-15 May 2021)
Corresponding author: enrico.aymerich@unica.it

Abstract
This work focuses on the development of a data driven model, based on Convolutional Neural Networks (CNNs), for the real-time detection of disruptive events at JET. The predictor exploits the ability of CNNs in learning relevant spatiotemporal information straight from 1D plasma profiles, avoiding hand-engineered feature extraction procedures. In this paper, the radiation profiles from both the bolometer horizontal and vertical cameras have been considered among the predictor inputs, with the aim of discriminating between the core radiation due to impurity accumulations and the outboard radiation phenomena. Moreover, an innovative predictor architecture is proposed, where two separate CNNs are trained to focus on events with different timescales, that is, the destabilization of radiation, electron density and temperature profiles, and the mode-locking and current profile variations. The outputs of the two CNNs are combined with a logic OR function to provide the disruption alarm trigger. The advantages of this data fusion approach impact on the predictor performance, with a very limited number of false alarms (only 1 in the considered test set), and on the model output interpretability as the two different branches are triggered by different types of events.

1. Introduction

Plasma disruptions pose considerable risks to the operation of high-power nuclear fusion devices. In fact, very stringent requirements apply to next-generation tokamaks in terms of the allowable number of unmitigated disruptions. In addition, the scientific community is currently actively working on the task of disruption avoidance, aiming to determine the mechanism that causes sudden discharge termination. The goal is to enable a recovery action or safe termination of the experiment without Massive Gas Injection (MGI). Data-driven methods have proven to be very powerful in fault prediction, and many approaches, such as Fully Connected Neural Networks (FC), Support Vector Machines (SVM), Self-Organizing Maps (SOM) and Generative Topographic Mapping (GTM), Classification and Regression Trees (CART) and Random Forests (RF), have been employed in disruption prediction models at JET [1]–[8], ASDEX Upgrade [9]–[11], EAST [12], J-TEXT [13], DIIID [14] and Alcator C-Mod [12]. Moreover, research is moving toward the objective of linking the physical phenomena involved in disruptive processes to the inputs of the data-driven prediction models, especially in case of predictive models designed to enable disruption avoidance [6], [15]–[18]. Recently, deep Convolutional Neural Networks (CNNs) have been adopted in several applications and became the state of the art for image processing and computer vision. In fact, they are able to automatically extract features from images, overcoming the need for hand-engineered feature extraction [19]–[21]. In [8] the authors proposed a disruption predictor based on a Convolutional Neural Network (CNN) for JET tokamak. The CNN processes an image obtained from a set of 1D profile data and 0D signals and returns two likelihoods: the disruptive one (in red in Figure 1a) and the non-disruptive one (in green in the same Figure 1a). The predictor was trained and tested on data spanning several years, showing overall good performances. However, a study of the performances evolution over the different campaigns revealed the predictor ageing, with an accuracy degradation, mainly in the false alarm number ([8], Table 3). Indeed, during the 2020 JET high power experiments, researchers observed the appearance of localized radiation in the Low Field Side (LFS) [22]. Since the previously developed predictor was only analysing the information from the bolometer horizontal camera, the CNN could not correctly locate the radiation source in these cases, hence triggering a false alarm. Figure 1 reports a false alarm triggered by the predictor in [8] on pulse #96893. As can be noted, at around 10.5 s, despite a non-disruptive behaviour shown by the HRTS profiles and the 0D signals, high radiation seen from the central lines of sight of the bolometer horizontal camera (Figure 1d) triggers an alarm, highlighted with a black dashed line in Figure 1a-f. This behaviour, observed in 4 other pulses, motivated the development of a new version of the predictor, also including the information from the vertical bolometer camera among the set of images.

This paper is structured as follows: Section 2 reports the Database used for this study; Section 3 explains the predictor architecture; Section 4 reports the training of the predictor model; Section 5 details the results. Finally, in Section 6 the conclusions of the study are discussed.

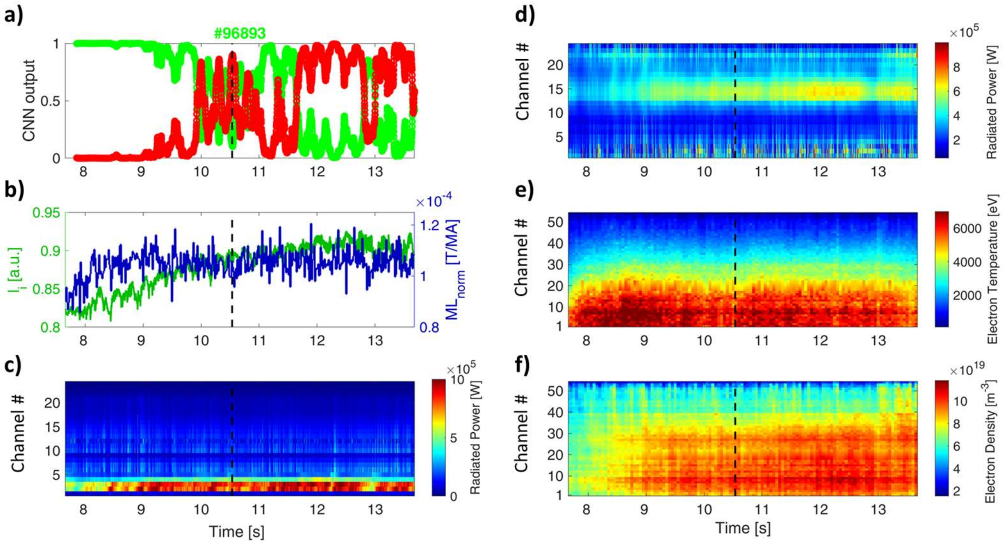

Figure 1: False alarm triggered by the predictor in [8] on the regularly terminated discharge #96893. (a) CNN likelihood curves, where the green line is the regularly termination likelihood and the red line is the disruption likelihood; (b) internal inductance in green and mode lock normalized by the plasma current in blue; (c) radiated power from the bolometer vertical camera; (d) radiated power from the bolometer horizontal camera; (e) electron temperature from the HRTS; (f) electron density from the HRTS. The dashed black line indicates the CNN alarm time.

2. Database

In this work, an upgraded version of the disruption predictor based on the CNN architecture presented in [8] is proposed. For sake of comparison, the database used to train and test the predictor is the same as in [8]. Precisely, the model has been trained using the same 85 disrupted and 70 regularly terminated discharges used in [7], [8], selected from the 2011-2013 JET campaigns. Then, the model was tested on 108 disruptive and 149 regularly terminated subsequent pulses from the 2011-2013 (42 disrupted /45 non-disrupted), 2016 (29/41), and 2019-2020 (37/63) JET campaigns. The pulses are analysed in the flat-top phase. For each selected discharge, the flat-top starting time is the first time instant where the plasma is in X-point configuration. The flat-top ending time t_{end}, is the last available time instant in the flat-top for the non-disrupted discharges, while for the disrupted ones, t_{end} is the minimum time between the valve activation time for mitigated discharges, and the disruption time (t_D), for the unmitigated ones. The predictor architecture has been designed to optimize the performance according to the new set of inputs. The 1D plasma profile data, from the High-resolution Thomson Scattering and the bolometer horizontal and vertical cameras were sorted to preserve the spatial ordering of the channels, resampled to maintain a common time basis, and pre-processed to remove outliers and unreliable channels. Afterwards, the input images to be fed to the CNN were extracted using a 200 ms sliding window. Then by using the maximum and the minimum values from each diagnostic in the training set, the four different images are normalized and vertically stacked, as shown in Figure 2.

3.Predictor architecture

In general, the architecture of a CNN contains several layers of blocks, connected in cascade, which filter an input image for feature extraction purposes [19]. Each filtering layer is interconnected with the following one by a nonlinear block (usually a rectified linear unit). Finally, a multi-layer perceptron combines the extracted features to produce the output of the CNN. A dropout layer may be included to improve the generalization of the model. Figure 3 shows the architecture of the predictor proposed in this work. It consists of two branches, each one being a separate CNN. The top branch, which processes the images of the 1D profiles, has two convolutional units (CU$_1$, CU$_2$) followed by a max pooling layer (P$_{max}$) and an average pooling layer (P$_{avg}$) respectively. The CU$_1$ and P$_{max}$ blocks, filter out vertically (spatial dimension) the input image by reducing its size from 154×101 to 18×101. The other blocks (CU$_2$, P$_{avg}$) filter out horizontally (time dimension) the resulting image by reducing the image size to 18×20. Each convolutional unit consists of three layers: a convolutional layer (C$_k$), a batch normalization layer (N$_k$) and a rectified linear unit (ReLU) activation layer (A$_k$). The first convolutional layer has a single filter (1-channel kernel) of size 5×1, while the second one has one of size 1×11. The output of the 2nd convolutional unit is then a 18×20 image. The lower branch processes the stacked signals of the internal inductance l_i and the normalized Locked Mode ML_{norm} signals. It consists of a separate Convolutional Unit (CU$_3$) with 4 filters (4-channel kernel) of size 1×5 with dilation 1×5 and stride 1×1, which process the

0D dimensional data along the horizontal direction. The block is followed by a max pooling layer with size and stride 1×5, which also down samples the features along the horizontal direction. The extracted features have a size of $2 \times 16 \times 4$. On both branches, the features are flattened and fed into a Fully Connected (FC) block, which combines them before a SoftMax layer (S). Before the two fully connected layers, a dropout layer with dropout probability of 20% reduces the overfitting on the training set and improves the model generalization.

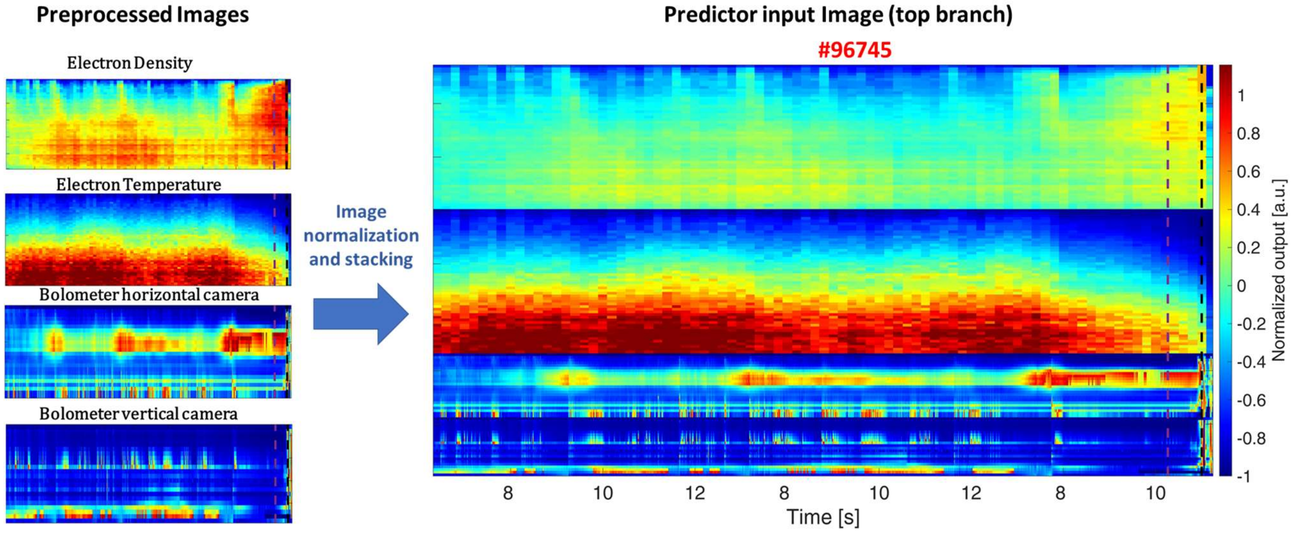

Figure 2: Sketch of the pre-processing steps applied to pulse #96745 to generate the input images for the predictor left) Pre-processed data are converted into images; Right) Input data for the CNN model, obtained by normalizing the data with the training set ranges and by vertically stacking the different diagnostics.

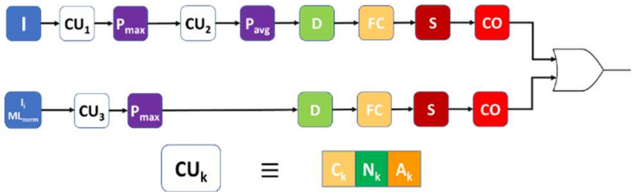

Figure 3: CNN architecture, where: I is the image input; CU$_k$ is the kth convolutional unit, composed by the cascade of a convolutional layer (C$_k$), a batch-normalization layer (N$_k$) and a nonlinear activation layer with ReLU functions (A$_k$); P$_{max}$ and P$_{avg}$ are the max-pooling and average-pooling layers, respectively; D is a dropout layer; FC is a fully-connected layer; S and CO are the SoftMax and classification output layers, respectively. Finally, an OR logic block activates the predictor whether one of the two branches output is 1.

The SoftMax layer produces the likelihood of the input segment to belong to a disrupted discharge. As an example, Figure 4a shows the SoftMax outputs for the JET disrupted pulse #96998, where the blue line refers to the disruptive likelihood from the top branch and the magenta line that one from the bottom branch. Finally, for each branch, a final classification layer (CO) simply thresholds the disruptive likelihood to perform the image classification. For each branch, a threshold on the likelihood is optimized by minimising the errors of the entire predictor on the training set, as detailed in Section 4. Figure 4b shows the branch binary outputs, which are obtained by setting to 1 the membership values greater than or equal to their own optimized threshold, and by setting to 0 the remaining ones. A disruptive behaviour is detected by a branch when its binary output equals 1 (blue curve for top branch and magenta curve for the bottom branch in Figure 4b). The logic OR function produces the final disruption trigger.

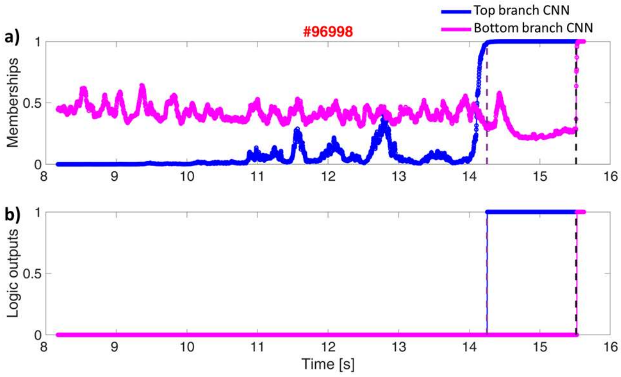

Figure 4: Disrupted pulse #96998 a) Disruptive membership functions for each predictor branch, where the blue line is the top branch one, and the magenta line is the bottom branch membership; b) Logic output for each branch (blue for the top branch, magenta for the bottom one) for the same pulse. The dashed purple line indicates the $t_{pre\text{-}disr}$, the dashed black line indicates the mode-locking time.

4 Predictor training

Since the CNN is employed in a supervised learning framework, it is necessary to explicitly label the time windows (or time slices) in the training dataset. In this work, the non-disrupted time slices are selected from the non-disrupted discharges, while the disrupted ones are obtained by identifying specific time windows within the disrupted discharges. Since the two CNNs were trained independently from each other, two different criteria have been adopted for defining the disruptive phase. The reason for adopting a different definition is the training of two specialized CNN branches, where each of them is focusing on events with different timings. In particular, the destabilization of the profiles at JET is usually due to the process of impurity accumulation or to the edge cooling [23], revealable by the plasma radiated power and density profiles, and it is exhibited at longer timescales than the insurgence of the locked mode. Hence, the two branches aim to increase the performance of the entire model exploiting the different information carried out by the profiles and the 0D signals. For the 1D profile images, the onset of disrupted phase is defined by the automatically identified pre-disruptive times $t_{pre-disr}$ as in [7], [8], whereas, for the 0D signal images, the onset of disrupted phase is defined by the mode locking time (t_{ML}). To this purpose, a threshold has been optimized, resulting in $2 \cdot 10^{-4}$ mT/MA, on the Locked Mode signal normalized by the plasma current. The time interval [t_{ML}, $min(t_{ML}$+0.3s, $t_{end})$] has been labelled as disruptive phase.

Due to the unbalance between the number of non-disrupted and disrupted samples, caused by the different duration of the two pre-disrupted phases, different subsampling strategies for the 200 ms sliding window have been adopted for the training. For the CNN top branch, one image every 24 ms has been sampled from the disrupted discharges in the timespan after $t_{pre-disr}$, whereas one image every 150 ms has been extracted from the non-disrupted discharges. This choice is motivated by the low resolution of the HRTS, which has a 50 ms sampling period. Instead, for the 0D signals every segment of pre-disrupted phase (i.e., one every 2ms) is considered for the training, whereas one segment every 200 ms is sampled from the regularly terminated discharges. In the test instead, the sliding window has a stride of 2ms, so that every sample of all the test discharges (regularly terminated and disrupted) has been classified. The alarm thresholds of the CO layers have been chosen by optimizing the full predictor performances on the training data. In the disruption prediction literature, the following metrics are generally considered when comparing different predictors:

- Successful predictions (SP): pulses that are correctly predicted (hence, an alarm is triggered in disruptive pulses and no alarm is raised in non-disruptive discharges).
- Missed alarms (MAs): pulses which disrupt where the predictor does not trigger an alarm.
- False alarms (FAs): non-disruptive discharges where the model triggers an alarm.

Considering these metrics, the single branch thresholds have been selected by minimizing the sum of the full predictor MAs and FAs, and then the distance between the alarm times and the $t_{pre-disr}$ on the training discharges. In fact, firstly a scan of the different thresholds identifies the combinations where the sum of the FAs and MAs is minimized. In this subset, the thresholds which minimize the mean distance between alarm times and $t_{pre-disr}$ are selected. The optimized thresholds result in 0.99 for the top branch and 0.925 for the bottom one.

5. Results and discussion

The results of the predictor are reported and compared with [8] in Table 1. The new model performs better both in the training and in the test sets. In particular, the predictor allows to greatly reduce the number of false alarms in the test set (from 14 to 1).

Table 1: Predictor performance

Dataset	SP	MA	FA
Train	98.71%	0%	2.85%
Train [8]	98.00%	0%	4.28%
Test	98.83%	1.87%	0.67%
Test [8]	93%	3.7%	9.4%

Another important metric for evaluating disruption predictors designed for avoidance and/or mitigation purposes is the warning time distribution, used to statistically evaluate the available time from the predictor alarm before the t_{end}. An early warning time could allow the adoption of automatic procedures to try to recover the disruptive plasma state or to safely terminate the experiment, while with a short warning time the disruption is generally mitigated. However, to allow the adoption of disruption avoidance strategies, the model should also provide information on the type of instability which is destabilizing the discharge. Figure 5 reports the warning times of the top branch (blue line), bottom branch (green line), and full predictor (black line) in the test dataset. If both branches are triggered in the same discharge, only the first alarm is plotted. Note that the top branch CNN, which processes the 1D profile data, can provide larger warning times than the bottom one, which instead detects the mode-locking phase. The separation of the two different mechanisms makes the predictor alarm more interpretable, in view of the development of avoidance schemes. Finally, the vertical red dashed line highlights that disruptions should be identified at least 10 ms in advance to adopt mitigation actions at JET. Detections with a warning time shorter than 10 ms are late or tardy alarms. The predictor can detect different disruptive patterns, as visible in Figure 6 which refers to the disrupted test pulse #96998 (outside the training range).

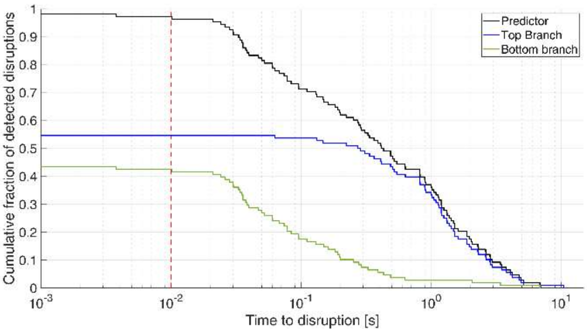

Figure 5: CNN model warning time distributions in the test set for the top branch (blue line), the bottom one (green line) and full predictor (black line). Only the first alarm is reported. The vertical red dashed line allows to identify tardive detections.

The top branch of the predictor in Figure 6a (blue line) triggers an alarm at around 14.10 s, coherently with the change of the plasma profiles shown in Figures 6c-f and in correspondence of the $t_{pre-disr}$, identified by a dashed line. In fact, the electron temperature flattens (Figure 6e) and the electron density peaks (Figure 6f). This phenomenon is synchronous with strong radiation from the central channels of the horizontal and vertical bolometer (Figures 6c-d). On the other hand, the bottom branch of the predictor in Figure 6a (magenta line) triggers an alarm at around 15.7s close to the end of the discharge, in correspondence with the rise of li and ML_{norm} signals. Hence, the top-branch is trained to detect destabilizations in the 1D profiles distributions, while the bottom branch on detecting the onset of a locked-mode and a late disruption pattern. Figure 7 shows the regularly terminated pulse #96893, which was detected as disruptive in [8]. In this case, the predictor does not trigger an alarm, because the high radiation pattern at chords #13-16 of the horizontal bolometer is not coincident with a high radiation from the central lines of sight of the vertical bolometer camera.

6. Conclusions

In this work, a disruption predictor based on CNNs has been developed using data from 2011-2013 campaigns at JET. The test of the model included more recent JET discharges and high power experiments up to the 2020 experimental campaigns. First, the vertical bolometer camera is added to the set of 1D plasma profile features considered in [8]. Then, two different CNN classifiers, whose thresholds are optimized to achieve the best full predictor performance, are trained to detect different destabilizing events. The automatic detection of the pre-disruptive phase of disruptions is used to train the top branch CNN, while an automatically identified locked mode time is employed for training the bottom branch of the model. The model can correctly identify the local perturbations of the 1D plasma profiles, leading to about, 98.87% of SPs, 0.67% of FAs and 1.87% of MA, considering a test set with 108 disruptive and 149 non-disruptive discharges. Moreover, the proposed approach allows to associate the predictor alarm with the destabilizing mechanism of the discharge. The automatic classification of the different profile instabilities, for instance distinguishing between temperature hollowing and edge cooling [23], would be another step forward towards the implementation of machine learning aided avoidance schemes.

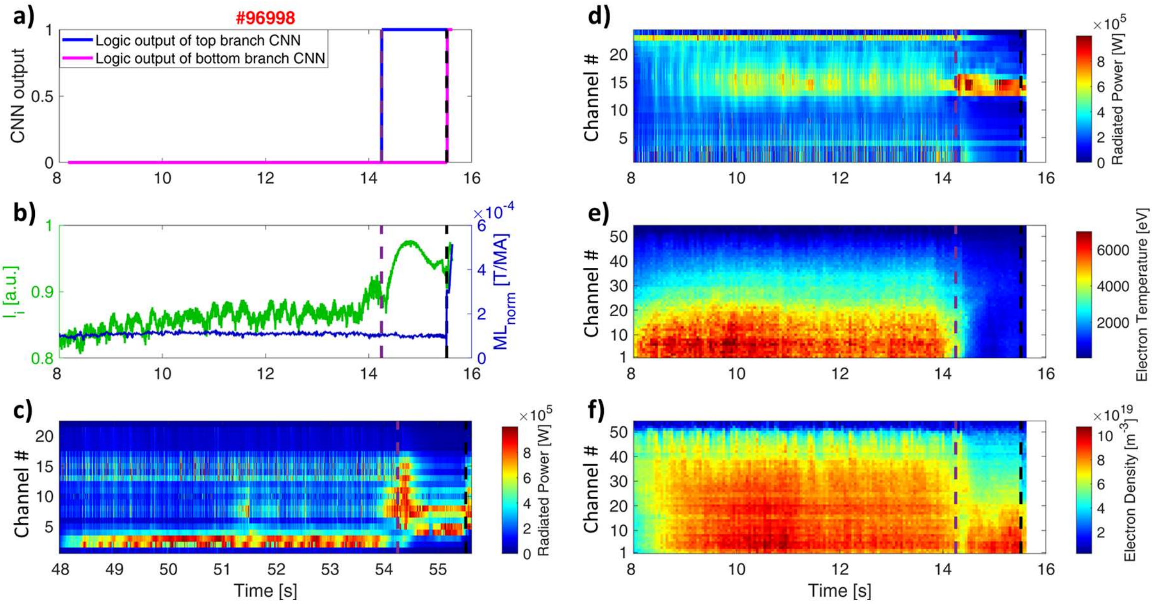

Figure 6: JET disrupted discharge #96998. (a) CNN logic output curves, where the blue line is the top branch logic output, and the magenta line is the bottom branch logic output.; (b) internal inductance, in green, and mode lock normalized by the plasma current, in

blue; c) radiated power from the bolometer vertical camera; (d) radiated power from the bolometer horizontal camera; (e) electron temperature from the HRTS; (f) electron density from the HRTS. The dashed purple line indicates the $T_{pre-disr}$

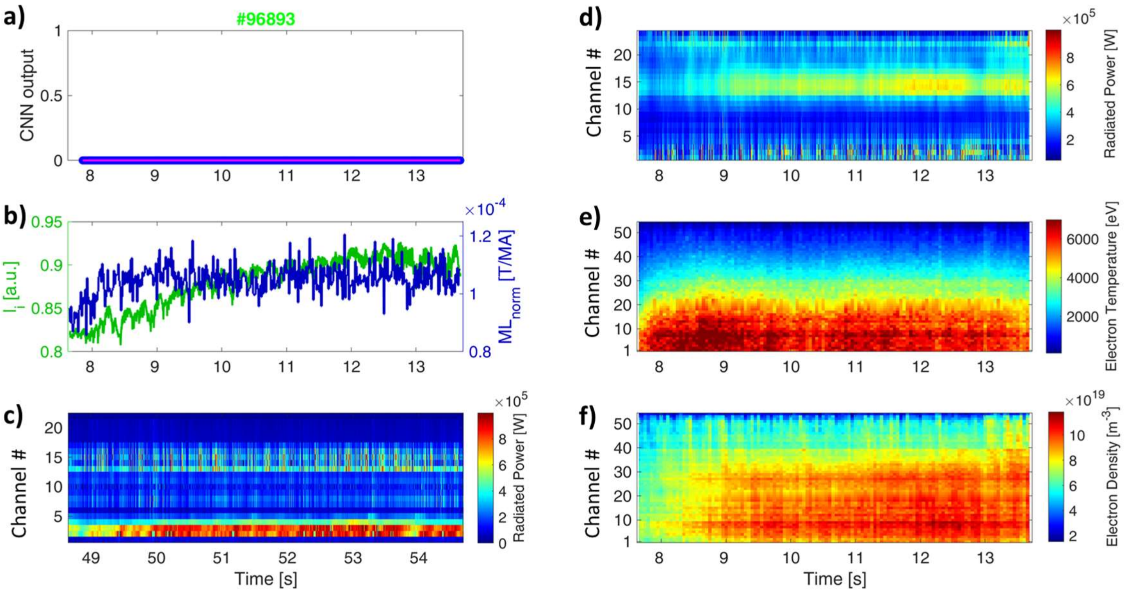

Figure 7: JET regularly terminated discharge #96893. (a) CNN logic output curves, where the blue line is the top branch logic output, and the magenta line is the bottom branch logic output.; (b) internal inductance, in green, and mode lock normalized by the plasma current, in blue; c) radiated power from the bolometer vertical camera; (d) radiated power from the bolometer horizontal camera; (e) electron temperature from the HRTS; (f) electron density from the HRTS.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] B. Cannas, A. Fanni, E. Marongiu, and P. Sonato, *Nucl. Fusion*, vol. 44, no. 1, pp. 68–76, Dec. 2003, doi: 10.1088/0029-5515/44/1/008.

[2] B. Cannas, A. Fanni, P. Sonato, and M. K. Z. and, *Nucl. Fusion*, vol. 47, no. 11, pp. 1559–1569, Oct. 2007, doi: 10.1088/0029-5515/47/11/018.

[3] B. Cannas, R. S. Delogu, A. Fanni, P. Sonato, and M. K. Zedda, *Fusion Engineering and Design*, vol. 82, no. 5, pp. 1124–1130, Oct. 2007, doi: 10.1016/j.fusengdes.2007.07.004.

[4] S. Dormido-Canto *et al.*, *Nucl. Fusion*, vol. 53, no. 11, p. 113001, Sep. 2013, doi: 10.1088/0029-5515/53/11/113001.

[5] A. Murari *et al.*, *Nucl. Fusion*, vol. 58, no. 5, p. 056002, Mar. 2018, doi: 10.1088/1741-4326/aaaf9c.

[6] A. Pau *et al.*, *Nucl. Fusion*, vol. 59, no. 10, p. 106017, Aug. 2019, doi: 10.1088/1741-4326/ab2ea9.

[7] E. Aymerich *et al.*, *Nucl. Fusion*, vol. 61, no. 3, p. 036013, Feb. 2021, doi: 10.1088/1741-4326/abcb28.

[8] E. Aymerich *et al.*, *Nucl. Fusion*, 2022, doi: 10.1088/1741-4326/ac525e.

[9] B. Cannas, A. Fanni, G. Pautasso, G. Sias, and P. Sonato, *Nucl. Fusion*, vol. 50, no. 7, p. 075004, Jun. 2010, doi: 10.1088/0029-5515/50/7/075004.

[10] B. Cannas, A. Fanni, G. Pautasso, and G. Sias, *Fusion Engineering and Design*, vol. 86, no. 6, pp. 1039–1044, Oct. 2011, doi: 10.1016/j.fusengdes.2011.01.069.

[11] R. Aledda, B. Cannas, A. Fanni, A. Pau, and G. Sias, *Fusion Engineering and Design*, vol. 96–97, pp. 698–702, Oct. 2015, doi: 10.1016/j.fusengdes.2015.03.045.

[12] K. J. Montes *et al.*, *Nucl. Fusion*, vol. 59, no. 9, p. 096015, Jul. 2019, doi: 10.1088/1741-4326/ab1df4.

[13] W. Zheng *et al.*, *Nucl. Fusion*, vol. 58, no. 5, p. 056016, Mar. 2018, doi: 10.1088/1741-4326/aaad17.

[14] C. Rea *et al.*, *Plasma Phys. Control. Fusion*, vol. 60, no. 8, p. 084004, Jun. 2018, doi: 10.1088/1361-6587/aac7fe.

[15] A. Pau *et al.*, *IEEE Trans. Plasma Sci.*, vol. 46, no. 7, pp. 2691–2698, Jul. 2018, doi: 10.1109/TPS.2018.2841394.

[16] J. Kates-Harbeck, A. Svyatkovskiy, and W. Tang, *Nature*, vol. 568, no. 7753, Art. no. 7753, Apr. 2019, doi: 10.1038/s41586-019-1116-4.

[17] C. Rea et al., *Fusion Science and Technology*, vol. 76, no. 8, pp. 912–924, Nov. 2020, doi: 10.1080/15361055.2020.1798589.

[18] P. C. de Vries *et al.*, *Physics of Plasmas*, vol. 21, no. 5, p. 056101, May 2014, doi: 10.1063/1.4872017.

[19] Y. Bengio, *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009, doi: 10.1561/2200000006.

[20] R. M. Churchill et al. , *Physics of Plasmas*, vol. 27, no. 6, p. 062510, Jun. 2020, doi: 10.1063/1.5144458.

[21] J. X. Zhu, et al., *Nucl. Fusion*, vol. 61, no. 2, p. 026007, Dec. 2020, doi: 10.1088/1741-4326/abc664.

[22] J. Garcia *et al.*, presented at the 28th IAEA Fusion Energy Conference (FEC 2020), 2021. Accessed: Oct. 11, 2021. [Online]. Available: https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_3320851

[23] G. Pucella *et al.*, *Nucl. Fusion*, vol. 61, no. 4, p. 046020, Mar. 2021, doi: 10.1088/1741-4326/abe3c7.