

# Deep Learning for Plasma Tomography and Disruption Prediction from Bolometer Data

Diogo R. Ferreira, Pedro J. Carvalho, Horácio Fernandes, and JET Contributors

**Abstract**—The use of deep learning is facilitating a wide range of data processing tasks in many areas. The analysis of fusion data is no exception, since there is a need to process large amounts of data collected from the diagnostic systems attached to a fusion device. Fusion data involves images and time series, and are a natural candidate for the use of convolutional and recurrent neural networks. In this work, we describe how CNNs can be used to reconstruct the plasma radiation profile, and we discuss the potential of using RNNs for disruption prediction based on the same input data. Both approaches have been applied at JET using data from a multi-channel diagnostic system. Similar approaches can be applied to other fusion devices and diagnostics.

**Index Terms**—Nuclear Fusion, Plasma Diagnostics, GPU Computing, Deep Learning

## I. INTRODUCTION

Deep learning [1] has become the state-of-the-art approach to many problems, especially those related to image processing and natural language processing. Convolutional neural networks (CNNs) have been extremely successful in image classification [2, 3], image segmentation [4, 5] and object detection [6, 7], to cite only a few examples. On the other hand, recurrent neural networks (RNNs) have been used for speech recognition [8, 9], language modeling [10, 11] and machine translation [12, 13], among other applications.

In general, it could be said that CNNs are appropriate for problems involving images and computer vision, whereas RNNs are especially useful for text and other sequential data, including time series [14]. However, this distinction is not clear-cut since, for example, it is possible to analyze images with RNNs [15], it is possible to perform sequence learning with CNNs [16], and there are hybrid models combining features from both CNNs and RNNs [17].

For the purpose of this work, we will focus on two broad categories: (1) image processing for plasma tomography, which will be addressed with CNNs, and (2) time series analysis for disruption prediction, which will be addressed with RNNs. In both cases, it becomes apparent that deep learning can play a key role in the compute-intensive tasks associated with the processing of large amounts of data originating from fusion diagnostics.

D. R. Ferreira and H. Fernandes are with the Institute for Plasmas and Nuclear Fusion (IPFN), a member of the EUROfusion Consortium, at IST, University of Lisbon, Portugal.

P. J. Carvalho is currently with JET at Culham Science Centre, Abingdon, OX14 3DB, UK.

For JET Contributors, see the author list of “Overview of the JET preparation for deuterium–tritium operation with the ITER like-wall” by E. Joffrin et al., in *Nuclear Fusion*, vol. 59, no. 11, p. 112021, August 2019.

Here, we focus on JET (Joint European Torus), a D-shaped tokamak with a major radius of 2.96 m and a minor radius of 1.25–2.10 m. JET has a vast assortment of diagnostics, including magnetic coils to measure plasma current and instabilities, interferometers and reflectometers to measure plasma density, Thomson scattering to determine the electron temperature, spectroscopy to measure ion temperature, and X-ray cameras to measure electromagnetic radiation, among others.

In this work, we will be using on a specific diagnostic, the bolometer system [18], which measures the plasma radiation on a poloidal cross-section of the fusion device. The signals collected from the bolometer system can be used to monitor the plasma state across an entire pulse. Several phenomena, such as impurity transport and accumulation at the plasma core, can be detected from the bolometer signals. Since these impurity-related phenomena are one of the most frequent precursors of disruptions at JET, this diagnostic plays an important role in disruption studies as well.

The data coming from this diagnostic is the basis for tomographic reconstructions that provide a 2D image of the plasma radiation profile. The reconstruction process itself is time-consuming. However, with a CNN trained on a large collection of sample tomograms, it becomes possible to produce those results much faster and with high accuracy.

In addition, the bolometer signals can be used to study disruption precursors. With a RNN trained on these signals, it is shown that bolometer data can provide a useful input for disruption prediction, both in terms of probability of disruption and time remaining to an impending disruption.

The paper is structured as follows. Section II provides a brief overview of the bolometer system, where the data is coming from, and of the tomographic method used at JET to reconstruct the plasma radiation profile. Section III describes the CNN that has been developed for plasma tomography, and the results that can be obtained with it. Section IV describes the RNN that has been developed for disruption prediction, and discusses its effectiveness when compared to other methods. Finally, Section V concludes the paper.

## II. THE BOLOMETER SYSTEM AT JET

One way to measure plasma radiation is through the use of bolometers, in particular foil bolometers [19]. These sensors consist of a thin metal foil (about 10  $\mu\text{m}$ ) coupled with a temperature-sensitive resistor. As the metal foil absorbs radiation power, its temperature changes and there is a proportional change in resistance. Overall, such system provides a linear response to the absorbed power, in the range from ultraviolet (UV) to soft X-ray [20].

The use of bolometers at JET dates back to the mid-1980s when the first bolometer system was installed [21]. This was followed by the development of other bolometer systems in the mid-1990s. One of these systems (KB3) was mainly directed at the divertor region at the bottom of the vessel. It has been subject to several improvements [22], and it is still partly in use today. Originally, KB3 included 7 cameras in the divertor region, each with 4 lines of sight.

In the late 1990s and early 2000s, the current bolometer system (KB5) was developed [18, 23, 24]. This diagnostic comprises one horizontal camera and one vertical camera, with 24 bolometers each. The lines of sight for each of these cameras are arranged in such a way that 16 channels cover the whole plasma, and 8 channels are directed towards the divertor region, as illustrated in Figure 1. This means the system provides an overview of the plasma region, and also a more detailed view of the divertor region.

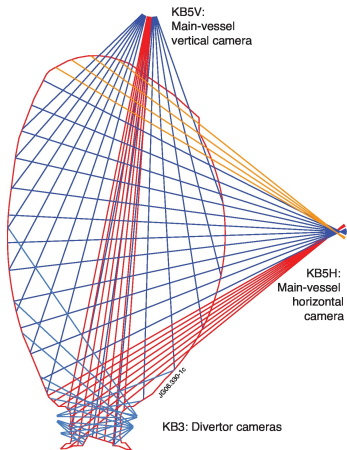


Fig. 1. KB5 cameras and lines of sight. KB5H and KB5V are not in the same octant, so we assume that the plasma is toroidally symmetric. The lines of sight for KB3 are also shown. (EUROfusion figures database JG06.330-1c)

In addition, the vertical camera has 8 extra bolometers that can be used as reserve channels, so in total the system provides  $24 + 24 + 8 = 56$  lines of sight over the plasma.

Technically, each bolometer measures the line-integrated radiation along its line of sight. From the bolometer measurements, it is possible to reconstruct the 2D plasma radiation profile using tomography techniques. The underlying principle is the same as computed tomography (CT) in medical applications [25], but the reconstruction method is different due to the scarce number of lines of sight available.

The method that is used at JET [26] actually predates the current bolometer system, and has been used with previous generations of soft X-ray diagnostics. In essence, it uses an iterative constrained optimization algorithm that minimizes the error with respect to the observed measurements, while requiring the solution to be non-negative. To do so, it solves a generalized eigenvalue problem in order to find a solution as a function of Lagrange multipliers, and then adjusts these Lagrange multipliers iteratively until the non-negativity constraints are satisfied [27]. Figure 2 shows a sample result.

This iterative method takes a significant amount of computation time. The total run-time depends on the actual data, but

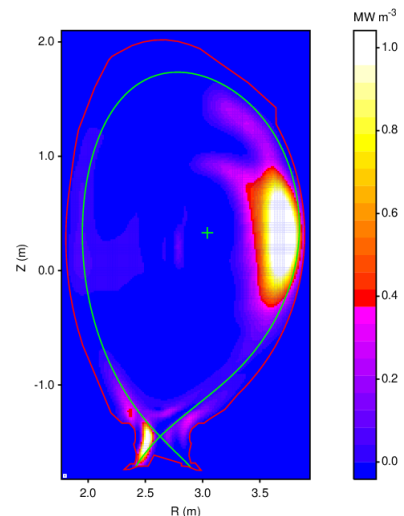


Fig. 2. Example of a tomographic reconstruction for pulse 92213 at  $t=50.0\text{s}$ . Many of such reconstructions have been used as training data in this work.

it can take from several minutes up to 1 hour to produce a single reconstruction. Given that the bolometer system has a sampling rate of 5 kHz, and a JET pulse typically lasts for about 30 seconds, it could take several years to compute all the tomographic reconstructions for a single pulse.

In the next section, we describe how deep learning was used to train a CNN that is able to compute 3000 reconstructions per second on a Graphics Processing Unit (GPU).

### III. DEEP LEARNING FOR PLASMA TOMOGRAPHY

In general, CNNs have a structure comprising multiple convolutional layers. Each convolutional layer applies several filters (in the form of a small kernel or sliding window) to the input. The purpose of a filter is to detect a specific feature, so its output is called a *feature map*. When CNNs are used for image classification, the first convolutional layer operates directly on the input image, while subsequent layers operate on the feature maps produced by previous layers.

In addition to the convolutional layers, CNNs contain subsampling layers. There is usually one subsampling layer after each convolutional layer. The purpose of having subsampling is to make the feature maps smaller and allow their number to progressively increase after each convolution, while keeping the network under a manageable size.

The output from these convolutional and subsampling layers is usually a large number of small feature maps. These feature maps are then flattened and connected to a couple of dense layers. The main idea behind image classification with CNNs is to have a first stage of convolutional layers to extract meaningful features from the input image, and a second stage of dense layers to perform the actual classification based on those features [28].

In summary, the input to a CNN is typically a 2D image and its output is a 1D vector of class probabilities. However, for the purpose of plasma tomography, it would be useful to have a network that takes a 1D vector of bolometer measurements as input, and produces a 2D image of the plasma radiation

profile as output. In the literature, the inverse of a CNN has been referred to as a *deconvolutional network*, and it has found applications in image segmentation [4] and object generation [29]. For example, by specifying the class label and the camera position (1D data), a deconvolutional network is able to generate an object (2D image) of the specified class from the given camera view [30].

### A. Network Architecture

Figure 3 shows the deconvolutional network developed for plasma tomography. It receives the bolometer measurements (a total of 56 lines of sight from both cameras) and produces a reconstruction of the plasma radiation profile with the same resolution as the tomographic reconstructions that are routinely produced at JET from bolometer data ( $196 \times 115$  pixels).

After the network input, there are two dense layers with 7500 nodes, which are reshaped into a 3D tensor of size  $25 \times 15 \times 20$ . This shape can be interpreted as comprising 20 feature maps of size  $25 \times 15$ . By applying a series of transposed convolutions, the feature maps are brought up to a size of  $200 \times 120$ , from which the output image is generated by one last convolution. (The output shape of  $200 \times 120$  is trimmed to  $196 \times 115$  by a simple slicing operation.)

Essentially, a *transposed convolution* is the inverse of a regular convolution in the sense that, if a sliding window would be applied to the output, the result would be the feature map given as input. It can be shown that learning a transposed convolution is equivalent to learning a weight matrix that is the transpose of a regular convolution, hence the name of this operation [31].

In Figure 3, the upsampling of feature maps from  $25 \times 15$  up to  $200 \times 120$  is achieved by having each transposed convolution operate with a stride of two pixels (i.e. one pixel is being skipped between each two consecutive positions of the sliding window). This means that the output is four times larger than the input, except for the very last convolution which uses a stride of one to keep the same size.

### B. Network Training

To train the network shown in Figure 3, we gathered all the tomographic reconstructions that have been computed at JET for the experimental campaigns from 2011 (since the installation of the ITER-like wall [32]) to 2016 (the time of the last completed campaign). We found 1219 pulses with a total of 27894 reconstructions (an average of about 23 reconstructions per pulse), and we split these pulses into 80% for training, 10% for validation, and 10% for testing. The splitting was done by pulse in order to avoid having (possibly similar) reconstructions from the same pulse across the training, validation and test sets. Such splitting yielded a training set of 976 pulses with 22236 reconstructions, a validation set of 122 pulses with 2736 reconstructions, and a test set of 121 pulses with 2922 reconstructions.

The network was trained on an NVIDIA Titan X GPU using accelerated gradient descent (Adam [33]) with a small learning rate ( $10^{-4}$ ) and a batch size of 436 samples. The batch size was chosen to be a perfect divisor of the number of training samples ( $22236/436 = 51$ ) in order to avoid having any partially filled batch in the training set.

The network was trained to minimize the mean absolute error between the output and the sample tomograms that were provided for training. Figure 4 shows the evolution of the training loss and of the validation loss across 1600 epochs. The best result was achieved at around epoch 800, with a minimum validation loss of  $0.0128 \text{ MW m}^{-3}$ . To appreciate the small scale of this error, it can be compared to the dynamic range of the reconstruction in Figure 2 ( $1.0 \text{ MW m}^{-3}$ ).

To assess the quality of the reconstructions produced by the network, it is possible to use image comparison metrics, such as structural similarity (SSIM) [34] or peak signal-to-noise ratio (PSNR) [35]. On the test set, the network achieves an SSIM value of  $0.936 \pm 0.061$  (the maximum is 1.0) and a PSNR of  $35.4 \pm 7.2 \text{ dB}$ , which is roughly similar to the error introduced when compressing an image to JPEG format.

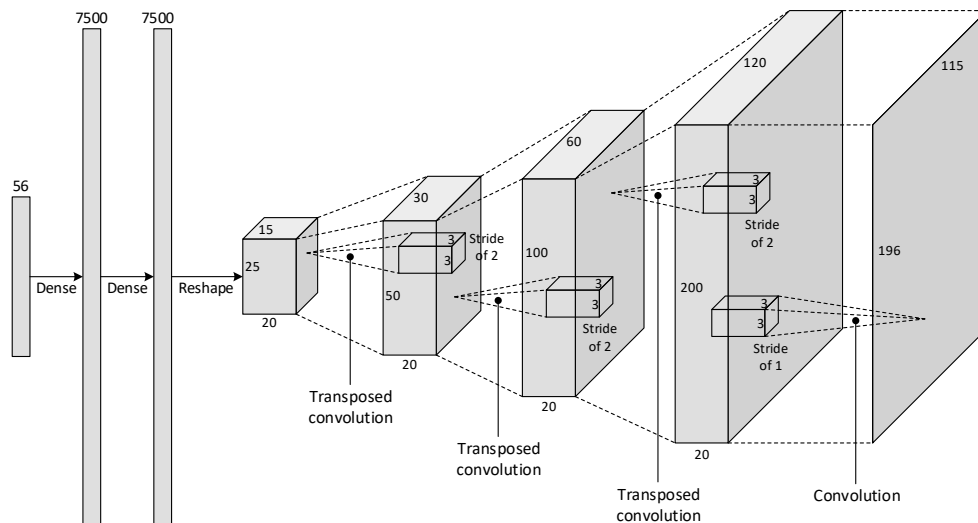


Fig. 3. Deconvolutional neural network for plasma tomography. The input layer receives all the 56 channels from the bolometer system and the output layer provides an image with  $196 \times 115$  resolution.

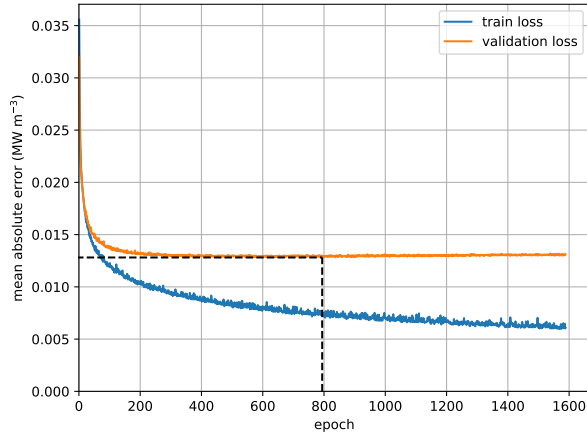


Fig. 4. Loss and validation loss during training of the deconvolutional network. The minimum validation loss, and the epoch at which it occurred, are indicated by dashed lines.

### C. Full-pulse reconstructions

Once trained, the network can be used to generate the reconstruction for any given pulse at any given point in time. In fact, given the bolometer data from an entire pulse, the network can generate a batch of reconstructions for every point in time. This way, it is possible to analyze the evolution of the plasma radiation profile across a full experiment.

As an example, Figure 5 shows the reconstruction of pulse 92213 from  $t=48.1$ s onwards, with a time step of 0.1s (due to space restrictions only). The first and second rows show a focus of radiation developing on the outer wall, which seems to slowly fade away (row 3), only to reappear later at the plasma core with particularly strong intensity (rows 3–4).

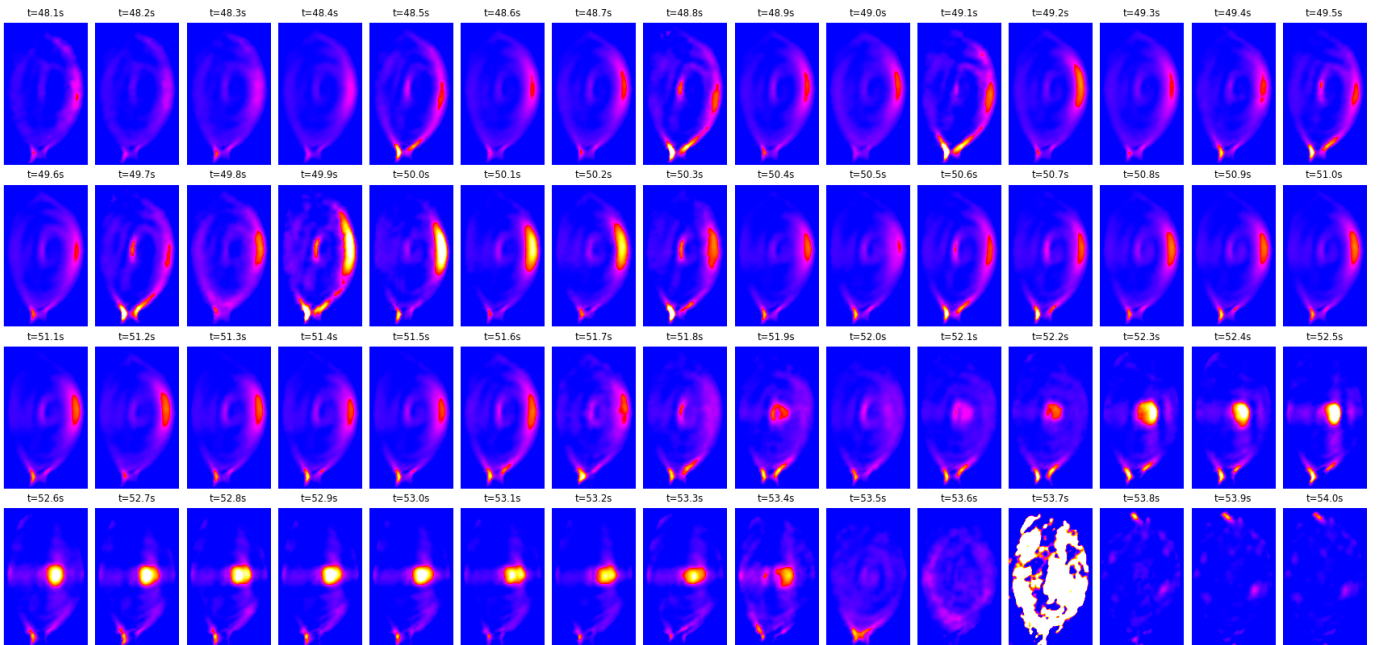


Fig. 5. Reconstructions of pulse 92213 from  $t=48.1$ s to  $t=54.0$ s with a time step of 0.1s. A video of the reconstructions for this pulse can be found in the supplementary material online.

This radiation peaking stays at the core for a relatively long time (at least from  $t=52.3$ s to  $t=53.4$ s), while changing slightly in shape during that interval. Eventually, it also fades away as the heating systems are being turned off. However, just as it seemed that the plasma was about to soft land, there is a disruption at around  $t=53.7$ s.

For comparison, Figure 6 shows the plasma current and heating power for this pulse. The disruption is clearly marked by a sudden current quench.

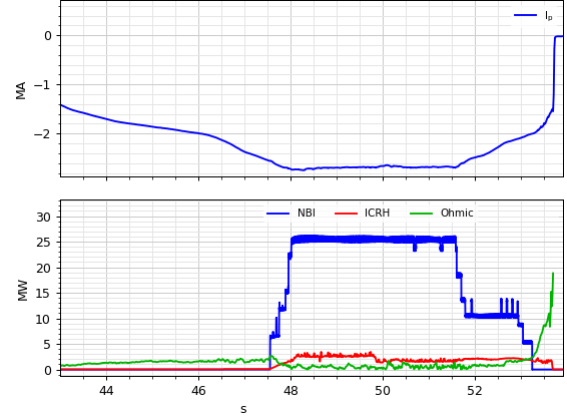


Fig. 6. Plasma current (top) and heating systems (bottom) for pulse 92213, including NBI (neutral beam injection), ICRH (ion cyclotron resonance heating) and ohmic heating.

Disruptions are one of the major problems affecting tokamaks today. While the dynamics of disruptions are not yet fully understood, one of the main reasons is believed to be impurity accumulation at the plasma core, which decreases the core temperature and eventually leads to core collapse [36].

To some extent, this phenomenon is clearly visible in Fig-

ure 5. It is therefore not surprising that the bolometer system has played a key role in several disruption studies at JET [37–40], either by providing a measurement of total radiation, or by providing the tomographic reconstruction at a few specific points in time. With the proposed network, it becomes possible to quickly generate a series of reconstructions that provide a detailed view of how the plasma profile evolves across time.

#### IV. DEEP LEARNING FOR DISRUPTION PREDICTION

Over the years, a number of different approaches to disruption prediction have been developed, including the use of neural networks [41–44], support vector machines [45–47] and decision trees [48, 49], among other techniques. In all of these approaches, the prediction model takes a set of global plasma parameters as input (e.g. the plasma current, the locked mode amplitude, the total input power, the safety factor, the poloidal beta, etc.) and outputs a probability of disruption, or the remaining time to a predicted disruption.

More recently, the use of random forests [50, 51] and even deep learning [52, 53] have shown promising results, but the input signals that are provided to these models still consist of about ten global plasma parameters derived from different diagnostics.

In the previous section, we have seen that a bolometer system with multiple lines of sight is able to capture some of the physical phenomena associated with plasma disruptions. From such bolometer data, it is possible to reconstruct the internal shape of the plasma radiation profile and observe, for example, an impurity concentration at the plasma core, which could be the precursor to an impending disruption.

This suggests that the sensor measurements coming from the bolometer system could, in principle, be used for disruption prediction. Figure 7 shows the bolometer signals for the same pulse that was reconstructed in Figure 5. The disruption is clearly visible at around  $t=53.7s$  and it would be easy to detect it from these signals. However, the point is whether it is possible to predict the disruption before it occurs.

As can be observed in Figure 7, before the disruption there is a period of about one second where the bolometer signals display a distinct pattern from the rest of the pulse. This corresponds to the radiation peaking at the plasma core. The bolometers that are yielding the strongest signals (e.g. KB5V channel 9, KB5V channel 5, and KB5H channel 14) are pointing precisely to that region.

In general, any model that is used for disruption prediction should be able to detect these or other kinds of precursors from data. In previous works, most authors used a set of global plasma parameters to learn and detect disruption precursors. Here, we will be using the bolometer signals, which provide a spatial view of the plasma radiation profile.

In addition to the spatial view, another dimension that will be important to consider is time. As shown in Figure 7, the disruption precursor develops and endures for a relatively long time (on the order of a second). Therefore, it will be important to consider not only the bolometer measurements at a certain point in time, but also the preceding measurements in order to have a sense of how the plasma behavior is developing and how far it is from a possible disruption.

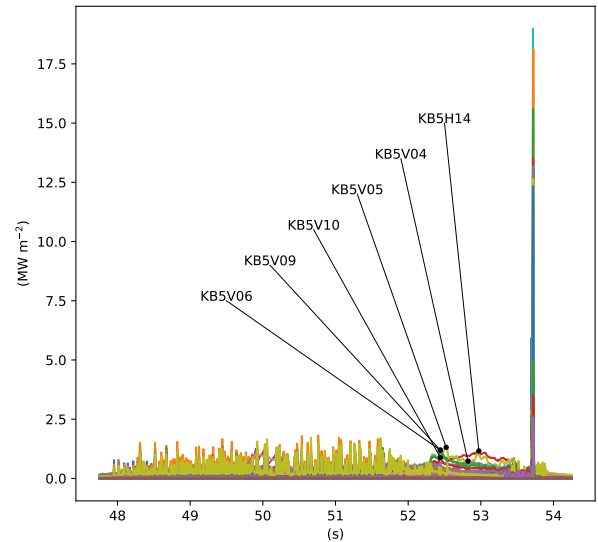


Fig. 7. Bolometer signals for pulse 92213. The channels that become more active before the disruption are indicated in the figure.

With both dimensions, the input to the predictor will be a sequence of vectors, where each vector contains the bolometer measurements at a given point in time. This closely resembles the way in which RNNs are used in natural language processing. In that case, the input is also a sequence of vectors, where the sequence is a text sentence and each vector is a word mapped into a vector space of a certain dimensionality, through a process called *word embedding* [54].

##### A. Network Architecture

A RNN is able to handle sequences of vectors by processing each vector at a time. The key distinctive feature of RNNs is that they maintain an internal state, and this internal state is fed back to the input along with the next vector in the sequence. A RNN can therefore “remember” features from past vectors in the sequence.

In simple RNNs, such “memory” is necessarily short-lived, since the network keeps updating its internal state as it receives new vectors, and gradually “forgets” about vectors that are in the distant past. To provide RNNs with the ability to maintain long-term dependencies, more advanced architectures have been proposed, namely the *long short-term memory* (LSTM) [55, 56]. In a LSTM, information can be carried across multiple time steps, and that information can be updated or forcefully forgotten at each time step, through the use of an *input gate* and a *forget gate*, respectively.

More recently, the combination of CNNs and RNNs has been shown to be very effective in tasks such as video processing [57, 58] and text classification [59, 60]. In essence, the CNN works as a feature extractor before passing the data to a RNN for sequential processing.

In this work, we use a similar approach to preprocess the bolometer data through a series of 1D convolutional layers before handing them over to a LSTM. As in a traditional CNN, we use subsampling after each convolutional layer to

progressively reduce the length of the time sequence, while increasing the number of filters that are applied to it.

The resulting network is shown in Figure 8. The input has a shape of  $200 \times 56$ , with the first being the sequence length, and the latter being the vector size. The sequence length of 200 time steps correspond to a 200 Hz downsampling of the bolometer signals. With a sequence of 200 time steps at 200 Hz, this means that one second of bolometer data is being considered in order to produce a prediction.

As before, the vector size of 56 corresponds to the number of lines of sight available in the bolometer system. Some of these lines correspond to unused reserve channels, which yield always a measurement of zero. Others correspond to bolometers which are known to be faulty, providing erratic measurements. In these cases, the network is expected to filter out those channels, so the number of filters used in the first convolutional layer (32) is actually less than the number of input channels (56), as shown in Figure 8.

In subsequent layers, the number of filters doubles after each convolution. A kernel size of 5 time steps is used in all convolutions, together with a max-pooling operation in order to reduce the sequence length to half. Eventually, the convolutional part of the network yields a sequence of 21 steps, where each step contains a 128-feature vector.

This sequence of feature vectors is handed over to a LSTM with 128 units, which returns the last output after having processed the whole sequence. One final dense layer combines these vector elements into a single output. This output can be either the probability of disruption or the time to disruption, as described next.

## B. Network Training

The network can be trained for two different objectives:

- *Classification* – To predict the probability of disruption (i.e. the probability that the current pulse will end in a

disruption), the last layer of the network will contain a sigmoid activation to produce a value between 0 and 1. The loss function used to train the network will be the binary cross-entropy [61], and the network will be trained using both disruptive and non-disruptive pulses.

- *Regression* – To predict the time to disruption (i.e. the remaining time up to an incoming disruption in the current pulse), the last layer of the network will have a linear activation, so that it can produce positive as well as negative values within any range. The loss function used to train the network is the mean absolute error, and the network is trained using disruptive pulses only.

To train these two variants, we gathered the bolometer signals for all pulses in the experimental campaigns from 2011 to 2016. In addition, we identified the disruptive pulses and gathered their disruption times from the JET disruption database [62]. Pulses with deliberate (i.e. intentional) disruptions were excluded from the dataset, as is done in other works [63]. Eventually, this yielded a total of 9323 pulses, 1444 of which were disruptive.

Again, we divided these pulses into 80% for training, 10% for validation, and 10% for testing. To train the network for probability of disruption, we used a data generator to draw samples at random time points from each pulse and feed those samples to the network, together with a binary label to indicate whether the sample was drawn from a disruptive or non-disruptive pulse. To train the network for time to disruption, we used a data generator to draw samples from disruptive pulses only, together with a label to indicate the time interval between the sample time and the disruption time.

In both cases, a batch size of 500 samples was used, where each sample is a  $200 \times 56$  sequence of vectors. The sample time is defined as the latest time step in that sequence. Each training epoch was defined as a run through 50 batches. For validation purposes, we used samples drawn at regular intervals from the

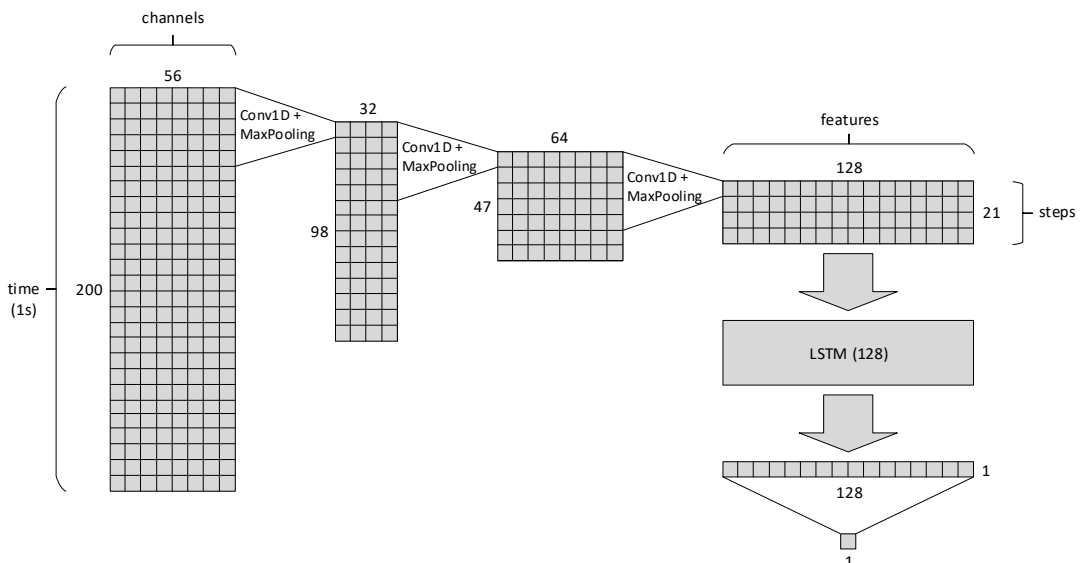


Fig. 8. Recurrent neural network for disruption prediction. The input layer receives a full second of data (200 samples) from the bolometer signals (56 channels) and the output layer provides a single scalar prediction, which can be either the time-to-disruption or the probability of disruption.

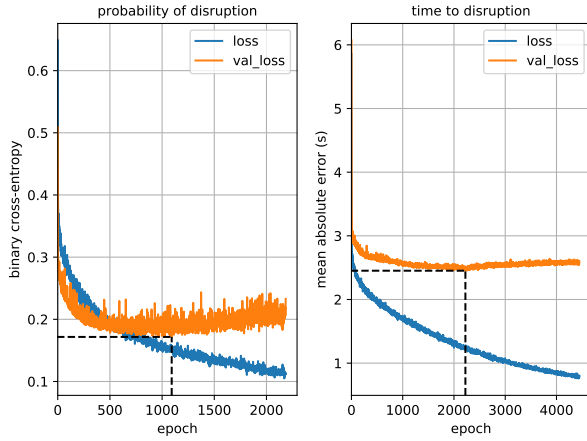


Fig. 9. Loss and validation loss during training of the recurrent network. The minimum validation loss, and the epoch at which it occurred, are indicated by dashed lines.

validation pulses.

Figure 9 shows the training loss and the validation loss for both variants. For probability of disruption, training ran for 2200 epochs, with the best result being achieved at around epoch 1100 with a validation loss (binary cross-entropy) of 0.172. For time to disruption, training ran for 4400 epochs with the best result being achieved at around epoch 2200 with a validation loss (mean absolute error) of 2.452 seconds.

Although these loss values may appear to be somewhat high, it should be noted that they are averaged over many samples. For samples that are far away from the disruption, the loss will be high because it is difficult to predict when or if a disruption will occur. This will be compensated by a small loss for samples that are near the disruption. Thus, the prediction will be more accurate when it matters the most.

In particular, the validation loss for the probability of disruption (left chart in Figure 9) seems to be quite noisy. This has to do with the fact that random samples are being drawn from both disruptive and non-disruptive pulses. In some cases (e.g. at the beginning of a pulse) samples from disruptive and non-disruptive pulses will be similar, despite having opposite labels. This makes the network weights oscillate across training in order to meet such contradictory labels.

### C. Full-pulse predictions

To illustrate the results that can be obtained with the trained network, Figure 10 shows the step-by-step predictions of the probability of disruption and of the time to disruption on an entire pulse. The disruption time is marked with a vertical dashed line at  $t=50.72s$ . For comparison, Figure 11 shows the same predictions for a non-disruptive pulse.

Some important observations can be made from these and other similar results:

- The disruption is characterized by a point in time where the probability of disruption is close to 1.0 and the time to disruption is close to 0.0. Therefore, when the probability of disruption approaches 1.0 and the time to disruption

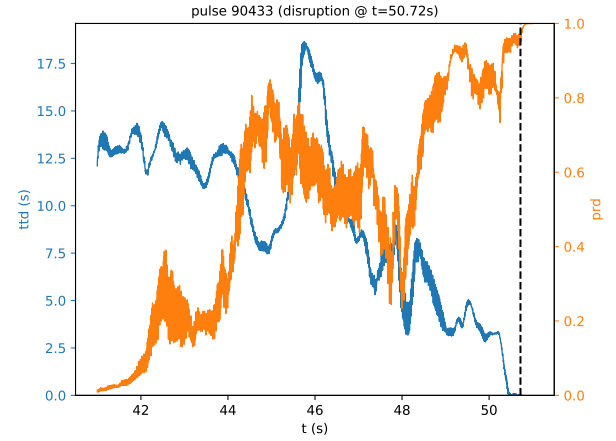


Fig. 10. Time to disruption ( $ttd$ ) and probability of disruption ( $prd$ ) for a disruptive pulse. The time of disruption is indicated by a vertical dashed line.

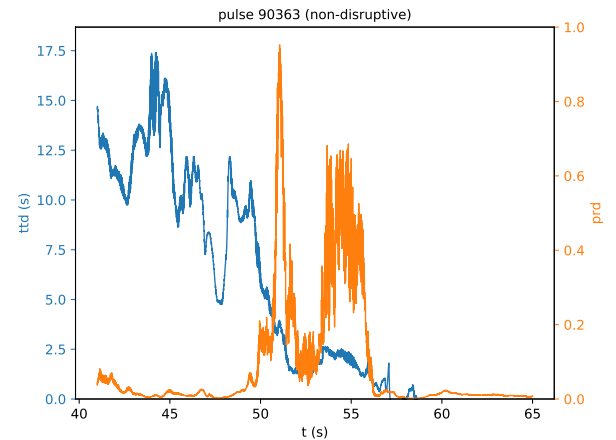


Fig. 11. Time to disruption ( $ttd$ ) and probability of disruption ( $prd$ ) for a non-disruptive pulse. The time-to-disruption reaches zero, even though there is no disruption.

approaches 0.0 at the same time, this is a strong indicator that a disruption is possibly imminent.

- It often happens that the probability of disruption is high at some intermediate points during the pulse, even for non-disruptive pulses. However, such indicator alone is not meaningful if the time to disruption is still high.
- It also happens that the time to disruption reaches zero for non-disruptive pulses. (Since this variant of the network has been trained exclusively on disruptive pulses, it will keep predicting that a disruption will occur even for non-disruptive pulses.) However, such indicator alone is not meaningful if the probability of disruption is low.

From these observations, it follows that the probability of disruption and the time to disruption should both be taken into account when deciding to take some action. For example, when the probability of disruption exceeds a certain threshold and the time to disruption falls below another threshold, an alarm could be raised to trigger the *disruption mitigation valve* (DMV) [64] which, in a matter of a few milliseconds, will

inject a massive amount of gas to cool down the plasma.<sup>1</sup>

However, even when using both predictors, there might be some false alarms. For example in Figure 11, at around  $t=51$ s, the probability of disruption rises and the time to disruption decreases in a way that an alarm could be triggered, even though the pulse is non-disruptive. Also, there might be situations where the network might not be able to predict a disruption on time (missed alarms).

#### D. Prediction accuracy

As with any binary classification problem, there will be true positives (successfully predicted disruptions), true negatives (correctly predicted absence of disruption), false positives (false alarms), and false negatives (missed alarms). Table I shows how the alarm-triggering thresholds on probability of disruption and on time to disruption affect the prediction results on the test set, which includes 932 pulses. For convenience, the results are being reported in terms of percentage. As a reference, the test set contains 157 (16.85%) disruptive pulses, and 775 (83.15%) non-disruptive pulses.

TABLE I  
PREDICTION RESULTS ON THE TEST SET WITH DIFFERENT  
ALARM-TRIGGERING THRESHOLDS

&	$prd \geq 0.95$	$prd \geq 0.90$	$prd \geq 0.85$	$prd \geq 0.80$
$td \leq 0.5$ s	<b>TP: 7.73%</b> <b>TN: 80.58%</b> <b>FP: 2.58%</b> <b>FN: 9.12%</b>	TP: 9.66% TN: 79.83% FP: 3.33% FN: 7.19%	TP: 10.62% TN: 79.51% FP: 3.65% FN: 6.22%	TP: 11.27% TN: 78.54% FP: 4.61% FN: 5.58%
$td \leq 1.0$ s	TP: 8.15% TN: 80.26% FP: 2.90% FN: 8.69%	<b>TP: 9.98%</b> <b>TN: 79.51%</b> <b>FP: 3.65%</b> <b>FN: 6.87%</b>	TP: 11.16% TN: 78.97% FP: 4.18% FN: 5.69%	TP: 12.02% TN: 76.93% FP: 6.22% FN: 4.83%
$td \leq 1.5$ s	TP: 9.12% TN: 79.94% FP: 3.22% FN: 7.73%	TP: 10.84% TN: 78.97% FP: 4.18% FN: 6.01%	<b>TP: 11.70%</b> <b>TN: 77.90%</b> <b>FP: 5.26%</b> <b>FN: 5.15%</b>	TP: 12.34% TN: 76.29% FP: 6.87% FN: 4.51%
$td \leq 2.0$ s	TP: 9.55% TN: 79.40% FP: 3.76% FN: 7.30%	TP: 11.59% TN: 78.22% FP: 4.94% FN: 5.26%	TP: 12.34% TN: 77.15% FP: 6.01% FN: 4.51%	<b>TP: 12.88%</b> <b>TN: 75.54%</b> <b>FP: 7.62%</b> <b>FN: 3.97%</b>

The results in bold (diagonal) illustrate the trade-off between false alarms (FP) and missed alarms (FN). As the threshold on probability of disruption becomes lower, and as the threshold on time to disruption becomes higher, the network is able to catch more disruptions, but at the expense of raising more false alarms. A good balance seems to be  $prd \geq 0.85$  and  $td \leq 1.5$ , where both precision  $TP/(TP+FP) = 69.0\%$  and recall  $TP/(TP+FN) = 69.4\%$  yield a similar value.

These results are below those reported in the literature. For example, the disruption predictor that is currently being used at JET (APODIS) has a recall of 85.38% and a false alarm rate (FP) of 2.46% [66]. However, it should be noted that APODIS is based on a set of global plasma parameters, while our prediction is based on the bolometer signals alone. The conclusion is that the bolometer signals are informative for disruption prediction, and they could possibly be combined

with other inputs in order to improve the accuracy of current predictors.

In recent works [67, 68] the bolometer signals have been used to extract one or two peaking factors of the radiation profile, which were then combined with other global parameters for disruption prediction. An advantage of using deep learning is that, in principle, it should be possible to extract these and/or other relevant features directly from the raw signals.

#### E. Warning time

Every time a disruption is correctly predicted, it is possible to calculate the *warning time* as the difference between the disruption time and the time at which the alarm was triggered. For the alarm-triggering thresholds  $prd \geq 0.85$  and  $td \leq 1.5$  we have already seen that the recall (i.e. the percentage of correctly predicted disruptions) is  $TP/(TP+FN) = 69.4\%$ . The question now is how far ahead, in terms of warning time, those disruptions are predicted.

Figure 12 shows the cumulative fraction of correctly predicted disruptions as a function of warning time. As expected, it is more difficult to predict disruptions far away from the disruption time (i.e. with higher warning times). However, in global terms, this yields an average warning time of about 2 s, which is significantly higher than that of APODIS (0.35 s). This suggests that a recurrent network applied to the time series of bolometer data may contribute to detecting disruption-relevant behavior earlier than other methods.

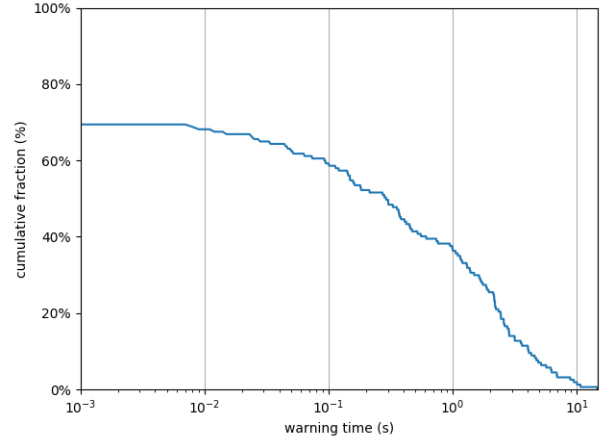


Fig. 12. Cumulative fraction of correctly predicted disruptions vs. warning time, for a recurrent neural network based on bolometer data.

## V. CONCLUSION

In this work, we have applied deep learning to a single, multi-channel diagnostic at JET. Based on the signals from this diagnostic, we used a CNN to reconstruct the plasma radiation profile, and a RNN to perform disruption prediction. The CNN provides results with high accuracy, and is therefore a reliable alternative to more time-consuming methods. On the other hand, the RNN does not reach the performance of other methods presently used at JET, but it provides the prospect of improving current disruption predictors.

<sup>1</sup>An alternative approach that is currently being developed for disruption mitigation at JET and other tokamaks is *shattered pellet injection* [65].



Both of these tasks could benefit from the use of data from multiple diagnostics. For example, it is possible to use the magnetic equilibrium to improve the plasma radiation profile, and this could be provided as an additional input to the CNN. On the other hand, disruption prediction can be made more accurate by taking into account several plasma parameters, and these could be additional inputs to the RNN.

In the future, it should be possible to build large deep learning models using data from many diagnostics in order to gain insights into the plasma behavior. The approaches described herein are just the beginning, and it is possible to envision that these approaches can be extended and applied to other fusion devices with minimal adaptations.

#### ACKNOWLEDGMENTS

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 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. IST activities also received financial support from *Fundação para a Ciência e Tecnologia* through project UID/FIS/50010/2019. The Titan X GPU used in this work was donated by NVIDIA Corporation.

#### REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
- [4] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.
- [5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. LNCS, no. 9351. Springer, 2015, pp. 234–241.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.
- [8] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.
- [9] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
- [10] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. K. (2), "Recurrent neural network based language model," in *11th Annual Conference of the International Speech Communication Association*, 2010, pp. 1045–1048.
- [11] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *13th Annual Conference of the International Speech Communication Association*, 2012, pp. 194–197.
- [12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.
- [13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.
- [14] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015, pp. 89–94.
- [15] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *33rd International Conference on Machine Learning*, 2016, pp. 1747–1756.
- [16] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *34th International Conference on Machine Learning*, 2017, pp. 1243–1252.
- [17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, pp. 802–810, 2015.
- [18] A. Huber, K. McCormick, P. Andrew, P. Beaumont, S. Dalley, J. Fink, J. Fuchs, K. Fullard, W. Fundamenski, L. Ingesson, F. Mast, S. Jachmich, G. Matthews, P. Mertens, V. Philipps, R. Pitts, S. Sanders, and W. Zeidner, "Upgraded bolometer system on JET for improved radiation measurements," *Fusion Engineering and Design*, vol. 82, no. 5, pp. 1327–1334, 2007.
- [19] L. C. Ingesson, B. Alper, B. J. Peterson, and J.-C. Vallet, "Chapter 7: Tomography diagnostics: Bolometry and soft-x-ray detection," *Fusion Science and Technology*, vol. 53, no. 2, pp. 528–576, 2008.
- [20] K. F. Mast, J. C. Vallet, C. Andelfinger, P. Betzler, H. Kraus, and G. Schramm, "A low noise highly integrated bolometer array for absolute measurement of vuv and soft x radiation," *Review of Scientific Instruments*, vol. 62, no. 3, pp. 744–750, 1991.
- [21] K. F. Mast, H. Krause, K. Behringer, A. Bulliard, and G. Magyar, "Bolometric diagnostics in JET," *Review of Scientific Instruments*, vol. 56, no. 5, pp. 969–971, May 1985.

- [22] L. C. Ingesson, C. F. Maggi, and R. Reichle, "Characterization of geometrical detection-system properties for two-dimensional tomography," *Review of Scientific Instruments*, vol. 71, no. 3, pp. 1370–1378, March 2000.
- [23] K. McCormick, A. Huber, C. Ingesson, F. Mast, J. Fink, W. Zeidner, A. Guigond, and S. Sanders, "New bolometry cameras for the JET enhanced performance phase," *Fusion Engineering and Design*, vol. 74, no. 1–4, pp. 679–683, November 2005.
- [24] A. Huber, K. McCormick, P. Andrew, M. de Baar, P. Beaumont, S. Dalley, J. Fink, J. Fuchs, K. Fullard, W. Fundamenski, L. Ingesson, G. Kirnev, P. Lomas, F. Mast, S. Jachmich, G. Matthews, P. Mertens, A. Meigs, V. Philipps, J. Rapp, G. Saibene, S. Sanders, R. Sartori, M. Stamp, and W. Zeidner, "Improved radiation measurements on jet – first results from an upgraded bolometer system," *Journal of Nuclear Materials*, vol. 363–365, pp. 365–370, 2007.
- [25] T. M. Buzug, *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*. Springer, 2008.
- [26] L. Ingesson, B. Alper, H. Chen, A. Edwards, G. Fehmers, J. Fuchs, R. Giannella, R. Gill, L. Lauro-Taroni, and M. Romanelli, "Soft X ray tomography during ELMs and impurity injection in JET," *Nuclear Fusion*, vol. 38, no. 11, pp. 1675–1694, 1998.
- [27] G. C. Fehmers, L. P. J. Kamp, and F. W. Sluijter, "An algorithm for quadratic optimization with one quadratic constraint and bounds on the variables," *Inverse Problems*, vol. 14, no. 4, p. 893, 1998.
- [28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [29] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 1538–1546.
- [30] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, "Learning to generate chairs, tables and cars with convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 692–705, 2017.
- [31] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv:1603.07285*, 2016.
- [32] G. F. Matthews, M. Beurskens, S. Brezinsek, M. Groth, E. Joffrin, A. Loving, M. Kear, M.-L. Mayoral, R. Neu, P. Prior, V. Riccardo, F. Rimini, M. Rubel, G. Sips, E. Villedieu, P. de Vries, and M. L. Watkins, "Jet iter-like wall – overview and experimental programme," *Physica Scripta*, vol. 2011, no. T145, p. 014001, 2011.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [35] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.
- [36] P. C. de Vries, M. F. Johnson, B. Alper, P. Buratti, T. C. Hender, H. R. Koslowski, and V. Riccardo, "Survey of disruption causes at JET," *Nuclear Fusion*, vol. 51, no. 5, p. 053018, 2011.
- [37] G. Arnoux, A. Loarte, V. Riccardo, W. Fundamenski, and A. Huber, "Heat loads on plasma facing components during disruptions on JET," *Nuclear Fusion*, vol. 49, no. 8, p. 085038, 2009.
- [38] V. Riccardo, G. Arnoux, P. Cahyna, T. C. Hender, A. Huber, S. Jachmich, V. Kiptily, R. Koslowski, L. Krilin, M. Lehnen, A. Loarte, E. Nardon, R. Paprok, and D. Tskhakaya, "JET disruption studies in support of ITER," *Plasma Physics and Controlled Fusion*, vol. 52, no. 12, p. 124018, 2010.
- [39] A. Huber, G. Arnoux, M. Beurskens, S. Bozhenkov, S. Brezinsek, T. Eich, C. Fuchs, W. Fundamenski, S. Jachmich, U. Kruezi, M. Lehnen, A. Loarte, G. Matthews, P. Mertens, P. Morgan, V. Philipps, R. Pitts, V. Riccardo, U. Samm, B. Schweer, G. Sergienko, and M. Stamp, "Radiation loads onto plasma-facing components of JET during transient events – experimental results and implications for ITER," *Journal of Nuclear Materials*, vol. 415, no. 1, pp. S821–S827, 2011.
- [40] M. Lehnen, A. Alonso, G. Arnoux, N. Baumgarten, S. A. Bozhenkov, S. Brezinsek, M. Brix, T. Eich, S. N. Gerasimov, A. Huber, S. Jachmich, U. Kruezi, P. D. Morgan, V. V. Plyusnin, C. Reux, V. Riccardo, G. Sergienko, and M. F. Stamp, "Disruption mitigation by massive gas injection in JET," *Nuclear Fusion*, vol. 51, no. 12, p. 123010, 2011.
- [41] G. Pautasso, C. Tichmann, S. Egorov, T. Zehetbauer, O. Gruber, M. Maraschek, K.-F. Mast, V. Mertens, I. Perchermeier, G. Raupp, W. Treutterer, and C. Windsor, "On-line prediction and mitigation of disruptions in ASDEX Upgrade," *Nuclear Fusion*, vol. 42, no. 1, p. 100, 2002.
- [42] B. Cannas, A. Fanni, E. Marongiu, and P. Sonato, "Disruption forecasting at JET using neural networks," *Nuclear Fusion*, vol. 44, no. 1, p. 68, 2004.
- [43] C. Windsor, G. Pautasso, C. Tichmann, R. Buttery, and T. Hender, "A cross-tokamak neural network disruption predictor for the JET and ASDEX Upgrade tokamaks," *Nuclear Fusion*, vol. 45, no. 5, p. 337, 2005.
- [44] R. Yoshino, "Neural-net predictor for beta limit disruptions in JT-60U," *Nuclear Fusion*, vol. 45, no. 11, p. 1232, 2005.
- [45] B. Cannas, R. Delogu, A. Fanni, P. Sonato, and M. Zedda, "Support vector machines for disruption prediction and novelty detection at JET," *Fusion Engineering and Design*, vol. 82, no. 5, pp. 1124–1130, 2007.
- [46] G. Rattá, J. Vega, A. Murari, G. Vagliasindi, M. Johnson, and P. de Vries, "An advanced disruption predictor for JET tested in a simulated real-time environment," *Nuclear Fusion*, vol. 50, no. 2, p. 025005, 2010.
- [47] J. M. López, J. Vega, D. Alves, S. Dormido-Canto, A. Murari, J. M. Ramírez, R. Felton, M. Ruiz, and G. de Arcas, "Implementation of the disruption predictor

- APODIS in JET's real-time network using the MARTE framework," *IEEE Transactions on Nuclear Science*, vol. 61, no. 2, pp. 741–744, 2014.
- [48] A. Murari, G. Vagliasindi, P. Arena, L. Fortuna, O. Barana, and M. Johnson, "Prototype of an adaptive disruption predictor for JET based on fuzzy logic and regression trees," *Nuclear Fusion*, vol. 48, no. 3, p. 035010, 2008.
- [49] A. Murari, J. Vega, G. Rattá, G. Vagliasindi, M. Johnson, and S. Hong, "Unbiased and non-supervised learning methods for disruption prediction at JET," *Nuclear Fusion*, vol. 49, no. 5, p. 055028, 2009.
- [50] C. Rea and R. S. Granetz, "Exploratory machine learning studies for disruption prediction using large databases on DIII-D," *Fusion Science and Technology*, vol. 74, no. 1–2, pp. 89–100, 2018.
- [51] C. Rea, R. S. Granetz, K. Montes, R. A. Tinguely, N. Eidietis, J. M. Hanson, and B. Sammulu, "Disruption prediction investigations using machine learning tools on DIII-D and Alcator C-Mod," *Plasma Physics and Controlled Fusion*, vol. 60, no. 8, p. 084004, 2018.
- [52] J. Kates-Harbeck, A. Svyatkovskiy, and W. Tang, "Predicting disruptive instabilities in controlled fusion plasmas through deep learning," *Nature*, vol. 568, pp. 526–531, 2019.
- [53] P. Sharma, S. Jain, V. Jain, S. Ranjan, R. Manchanda, D. Raju, J. Ghosh, and R. L. Tanna, "A proposed method for disruption classification in tokamak using convolutional neural network," in *Towards Extensible and Adaptable Methods in Computing*. Springer, 2018, pp. 179–193.
- [54] Y. Goldberg, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool, 2017.
- [55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [56] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.
- [57] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 2625–2634.
- [58] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.
- [59] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers)*, 2016, pp. 225–230.
- volutional and recurrent neural network for sentiment analysis of short texts," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, 2016, pp. 2428–2437.
- [61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [62] S. N. Gerasimov, P. Abreu, I. S. Carvalho, R. B. Henriques, G. Artaserse, P. Buratti, M. Baruzzo, I. H. Coffey, E. D. L. Luna, T. C. Hender, R. Felton, P. J. Lomas, P. McCullen, M. Maslov, F. G. Rimini, W. Schippers, G. Szepesi, U. Kruezi, M. Tsalas, S. Moradi, L. Piron, and L. E. Zakharov, "Overview of disruptions with JET-ILW," in *27th IAEA Fusion Energy Conference (FEC 2018)*, 2018.
- [63] J. Vega, S. Dormido-Canto, J. M. López, A. Murari, J. M. Ramírez, R. Moreno, M. Ruiz, D. Alves, and R. Felton, "Results of the JET real-time disruption predictor in the ITER-like wall campaigns," *Fusion Engineering and Design*, vol. 88, no. 6, pp. 1228–1231, 2013.
- [64] K. H. Finken, M. Lehnen, and S. A. Bozhnikov, "A new disruption mitigation valve (DMV) and gas flow in guiding tubes of different diameter," *Nuclear Fusion*, vol. 51, no. 3, p. 033007, 2011.
- [65] L. Baylor, S. Meitner, T. Gebhart, J. Caughman, J. Herfindal, and D. Shiraki, "Shattered pellet injection technology design and characterization for disruption mitigation experiments," in *27th IAEA Fusion Energy Conference (FEC 2018)*, 2018.
- [66] R. Moreno, J. Vega, S. Dormido-Canto, A. Pereira, and A. Murari, "Disruption prediction on JET during the ILW experimental campaigns," *Fusion Science and Technology*, vol. 69, no. 2, pp. 485–494, 2016.
- [67] A. Pau, A. Fanni, B. Cannas, S. Carcangiu, G. Pisano, G. Sias, P. Sparapani, M. Baruzzo, A. Murari, F. Rimini, M. Tsalas, P. C. de Vries, and JET Contributors, "A first analysis of JET plasma profile-based indicators for disruption prediction and avoidance," *IEEE Transactions on Plasma Science*, vol. 46, no. 7, pp. 2691–2698, July 2018.
- [68] A. Pau, A. Fanni, S. Carcangiu, B. Cannas, G. Sias, A. Murari, F. Rimini, and JET Contributors, "A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET," *Nuclear Fusion*, vol. 59, no. 10, p. 106017, August 2019.