UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN

Industrial Engineering

Cycle XXXV

TITLE OF THE Ph.D. THESIS

Machine Learning and Deep Learning applications for the protection of nuclear fusion devices

Scientific Disciplinary Sector(s)

Electrotechnics (ING-IND/31)

Ph.D. Student: Enrico Aymerich

Supervisor Prof. Alessandra Fanni

Co-Supervisor Dr. Fabio Pisano

Final exam. Academic Year 2021/2022

Thesis defence: April 2023 Session

Dichiarazione sul riutilizzo

Introduzione

La fusione nucleare controllata è un'opzione per produrre energia pulita, sicura, scalabile e non intermittente. In particolare, la fusione a confinamento magnetico è l'approccio più studiato per la produzione di energia e i Tokamak e gli Stellarator sono di gran lunga i concetti più promettenti per i futuri reattori a fusione. Entrambi i dispositivi confinano il plasma con forti campi magnetici e producono linee di campo magnetico elicoidali. I Tokamak sono macchine a funzionamento pulsato in cui il campo magnetico poloidale, necessario per il confinamento, è ottenuto inducendo una corrente toroidale nel plasma, mentre negli Stellarator le bobine hanno una geometria complessa per generare direttamente il campo elicoidale. Questa differenza determina il funzionamento dei due dispositivi e i vantaggi e gli svantaggi di ciascun progetto.

Il funzionamento di entrambi i dispositivi richiede una profonda comprensione della fisica del plasma e un'attenta pianificazione dei parametri di controllo. In effetti, ci sono due obiettivi principali che devono essere soddisfatti durante l'esecuzione di esperimenti nei dispositivi attuali: il raggiungimento di elevate prestazioni del plasma e la salvaguardia dell'integrità della macchina. Infatti, se da un lato i ricercatori mirano a ottenere le migliori condizioni di plasma in termini di temperatura, densità e lunghezza di scarica, dall'altro l'interazione con il plasma può danneggiare i componenti di prima parete, costringendo a interrompere le operazioni, riducendo il tempo sperimentale disponibile nei dispositivi attuali e portando a una costosa riparazione del dispositivo.

Uno dei principali svantaggi dei Tokamak è che la corrente toroidale rende i dispositivi soggetti a disruzioni. La disruzione è la perdita improvvisa della corrente di plasma e rilascia enormi forze elettromeccaniche e termiche sulle pareti del dispositivo. Poiché le disruzioni possono causare gravi danni ai componenti del plasma, molti sforzi sono diretti all'identificazione dei precursori, delle cause e delle conseguenze delle disruzioni dei Tokamak; l'obiettivo finale è lo sviluppo di schemi e strategie automatiche per mitigare o evitare le disruzioni. Esistono studi in cui le disruzioni vengono classificate identificando le sequenze di eventi che portano alla disruzione, e altri dedicati allo sviluppo di routine e algoritmi in grado di rilevare eventi specifici correlati alle disruzioni. In letteratura sono stati implementati sia approcci basati sulla fisica sia approcci guidati dai dati per prevedere e classificare le disruzioni. I metodi basati sulla fisica hanno il vantaggio di essere direttamente interpretabili e più scalabili tra diversi dispositivi, ma non sono ancora disponibili modelli fisici generali e autoconsistenti che possano essere eseguiti in tempo reale. Gli approcci basati sui dati possono invece sfruttare la grande quantità di dati disponibili dagli esperimenti e sono considerati un approccio alternativo alla previsione delle disruzioni. A questo scopo sono stati studiati sia metodi statistici che di intelligenza artificiale (AI), come il Machine Learning e il Deep Learning.

D'altra parte, lo Stellarator può funzionare in regime stazionario e gli effetti delle perdite di confinamento sulla struttura della macchina sono trascurabili. Tuttavia, a causa dei lunghi tempi di funzionamento ottenibili con questa configurazione, molti sforzi sono dedicati alla prevenzione dei surriscaldamenti nella prima parete di questi dispositivi. A questo scopo, un'intensa attività di ricerca mira a rilevare gli eventi termici nello stellarator Weldenstein 7-X (W7-X), dove telecamere a infrarossi monitorano lo stato della prima parete durante gli esperimenti e dove è in fase di sviluppo un sistema di sicurezza completamente automatico per interrompere il funzionamento se viene rilevato un surriscaldamento. Soprattutto negli stellarator, a causa della loro geometria tridimensionale, una complessa interazione tra la topologia magnetica nel confine dell'isola, la modellazione locale delle componenti di fronte al plasma e il rapporto tra il trasporto in campo parallelo e quello in campo incrociato determina la distribuzione del flusso di calore. Per analizzare e controllare la distribuzione del flusso di calore sulla prima parete, è necessario sviluppare algoritmi in grado di stimare in modo affidabile il flusso di calore dalla temperatura in tempo reale.

Questa tesi discute l'uso di metodi di intelligenza artificiale per la protezione dei dispositivi di fusione nucleare con riferimento al Tokamak Joint European Torus (JET) situato a Culham, nel Regno Unito, e allo Stellarator Wendenstein 7-X (W7-X), a Greifswald, in Germania. Entrambi i dispositivi fanno parte del programma EUROfusion per lo sviluppo della ricerca sulla fusione nucleare. JET è attualmente il più grande Tokamak operativo al mondo e l'unico che può funzionare con il combustibile Deuterio-Trizio, mentre W7-X è lo Stellarator più grande e avanzato al mondo, con l'obiettivo di studiare la possibilità di un'alternativa di tipo Stellarator in vista di una commercializzazione dei reattori a fusione.

Per quanto riguarda JET, in questo lavoro di tesi, il database esistente, gestito dall'Università di Cagliari, è stato aggiornato con le scariche provenienti dalle campagne sperimentali JET dal 2016 al 2020, con particolare attenzione alle campagne C36 (2016) [1] e C38 (2019-2020) [2]. Sia le scariche disrotte che quelle regolarmente terminate sono state selezionate dalle campagne sperimentali effettuate al JET, dopo l'installazione della parete simile a quella di ITER (ILW o ITER Like Wall). In totale, il database di questo lavoro contiene 198 scariche disrotte e 219 scariche regolarmente terminate con una corrente di plasma al flat-top superiore a 1,5 MA e una lunghezza di flat-top superiore a 200 ms. L'analisi degli impulsi si riferisce alla fase di flat-top e il tempo di inizio del flat-top è stato assunto come il primo istante in cui il plasma è in configurazione X-point. Negli impulsi del database delle campagne sperimentali C28-C30 (2011-2013), gli autori di [A. Pau, et al., Nucl. Fusion 51 (2019) 106017] hanno identificato manualmente il cosiddetto tempo pre-disruttivo di una scarica disrotta, che fornisce un tempo di riferimento per separare la corrente di plasma flat-top di ogni scarica disrotta in due parti: una parte non perturbata e una parte pre-disruttiva. Questa seconda parte è definita come la fase in cui si verifica la catena di eventi che porta alla disruzione. L'introduzione di

tempi pre-disruttivi coerenti ($t_{pre-disr}$) è doppiamente vantaggiosa. In primo luogo, questi tempi permettono di identificare la fase pre-disruttiva, che viene utilizzata per descrivere lo spazio di input disrotto di qualsiasi modello predittivo di IA. In secondo luogo, essendo il tempo pre-disruttivo fortemente legato all'insorgenza di fenomeni destabilizzanti, la risposta del predittore dovrebbe essere collegata alla fenomenologia o ai precursori che caratterizzano i vari tipi di disruzioni. Nella maggior parte della letteratura, questa fase pre-disruttiva è stata identificata statisticamente o euristicamente e assunta uguale per tutti le disruzioni presenti nel database, introducendo informazioni contraddittorie nel modello di previsione. La chiave per un modello di previsione di successo è quindi la capacità, per ogni scarica disrotta nell'insieme di addestramento, di discriminare tra le fasi non disrotte e quelle pre-disrotte seguendo criteri standard e coerenti, legati ai meccanismi fisici osservati. Tuttavia, questa classificazione richiede un'analisi manuale molto dispendiosa in termini di tempo; di conseguenza, adottarla per classificare decine di migliaia di impulsi sarebbe altamente impraticabile. Pertanto, durante il dottorato, è stato sviluppato un algoritmo per l'identificazione automatica dei tempi pre-disruttivi, basato su un approccio statistico [1]. Gli istogrammi sono stati utilizzati per stimare la distribuzione di probabilità, basandosi sul fatto che l'istogramma di una misura fornisce la base per una stima empirica della distribuzione di probabilità. Per quantificare la somiglianza/difformità di due istogrammi si possono utilizzare diversi approcci.

Nell'approccio proposto, i due istogrammi sono considerati come vettori multidimensionali e la somiglianza/dissimilarità di due istogrammi (o distribuzioni di probabilità) è valutata come distanza tra vettori. Per valutare la misura di distanza geometrica sono disponibili diverse metriche, come le semplici funzioni norma L1 o L2, o quelle appartenenti alle famiglie dell'intersezione o del prodotto interno. In questo caso è stata utilizzata la metrica del coseno, appartenente a quest'ultima famiglia. La dissimilarità viene valutata per diversi parametri del plasma e poi si assume una somma ponderata ottimale come dissimilarità complessiva. È stato introdotto un criterio ottimale per scegliere automaticamente, per ogni scarica, il tempo di pre-disruzione su questa misura di dissimilarità totale. L'algoritmo si basa sull'analisi statistica dei parametri del plasma selezionati dalle campagne sperimentali JET eseguite dal 2011 al 2013 e fornisce, per ogni scarica perturbata, il tempo di pre-distruzione ($t_{pre-disr,AUT}$). Questi tempi sono stati confrontati con quelli ($t_{pre-disr,MAN}$) identificati manualmente in [A. Pau, et al., Nucl. Fusion 51 (2019) 106017], in termini di prestazioni dei modelli di previsione basati sul machine learning Generative Topographic Mapping (GTM) [C.Bishop, et al., Neural Comput. (1998) 10(1), 215-234]. Inoltre, l'algoritmo proposto è stato applicato al database JET aggiornato e le prestazioni del modello GTM aggiornato confermano l'idoneità dell'algoritmo [1,6]. Un altro aspetto legato ai modelli AI data-driven riguarda l'uso di caratteristiche informative legate ai fenomeni fisici osservati sperimentalmente. In particolare, informazioni preziose possono essere ottenute dai

profili di temperatura, densità e radiazione del plasma a causa della loro stretta connessione con la stabilità del plasma e la destabilizzazione dei modi MHD che possono causare perturbazioni. A titolo di esempio, nella maggior parte dei casi, una disruzione è la conseguenza dello sviluppo di tearing modes nel plasma, che porta alla crescita delle isole magnetiche. Di solito, ben prima dell'insorgere dei tearing modes, si osserva un aumento dell'emissione di radiazione nel nucleo, che porta a un profilo di temperatura bucato, mentre un aumento dell'emissione di radiazione ai bordi del plasma porta a un raffreddamento ai bordi. In caso di collasso della temperatura, si osserva un allargamento del profilo della densità di corrente dall'interno, mentre un restringimento dello stesso profilo dall'esterno corrisponde al raffreddamento del bordo. In entrambi i casi si può verificare uno scenario MHD instabile, dovuto a un continuo aumento del gradiente di densità di corrente in prossimità della superficie di risonanza del modo. D'altra parte, l'informazione spazio-temporale contenuta nei profili del plasma è cruciale per descrivere fenomeni localizzati destabilizzanti, come l'emissione di radiazione nel nucleo piuttosto che al bordo, che non può essere sufficientemente descritta da parametri zero-dimensionali (0-D) variabili nel tempo, come la frazione della potenza irradiata. A tal fine, sono stati sintetizzati parametri 0-D con peaking factors per codificare l'informazione spaziale contenuta nei profili unidimensionali (1-D), attraverso il rapporto tra i valori medi delle misure su diverse regioni della sezione trasversale del plasma. La dimensione temporale viene trascurata quando si considera la dimensione dei dati poiché il modello data-driven riceverà un singolo valore del segnale alla volta.

I parametri dei peaking factors costruiti a partire dai profili di temperatura, densità e radiazione del plasma, e quindi direttamente collegati alla fisica del plasma, hanno dimostrato di aumentare le prestazioni del modello GTM di apprendimento automatico che prevede le disruzioni con un preavviso sufficientemente ampio per consentire la messa in atto di strategie di recupero dell'esperimento. Tuttavia, le definizioni di tali peaking factors si basano su euristiche che assumono arbitrariamente le definizioni del "core" e del "divertore" e possono perdere preziose informazioni spaziali contenute nei profili del plasma. Inoltre, le definizioni dei peaking factors devono essere modificate in base ai diversi sistemi diagnostici disponibili nei vari dispositivi. Recentemente, le reti neurali convoluzionali (CNN), appartenenti al paradigma del Deep Learning, si sono dimostrate in grado di superare le tecniche di apprendimento automatico più consolidate, soprattutto nel campo dell'elaborazione delle immagini e della computer vision, per la loro capacità di apprendere caratteristiche rilevanti da immagini a diverse scale, evitando l'estrazione manuale delle caratteristiche.

In questa tesi, le CNN vengono proposte sia per estrarre le caratteristiche spazio-temporali dai profili di temperatura, densità e radiazione del plasma, superando i limiti precedentemente descritti dei peaking factors 0-D, sia per sviluppare un modello di previsione delle disruzioni basato su una rete neurale profonda piuttosto semplice che utilizza queste caratteristiche insieme ad altri

segnali diagnostici comunemente utilizzati in letteratura. Il modello di previsione della rete neurale profonda è stato addestrato utilizzando i dati delle campagne sperimentali effettuate al JET dal 2011 al 2013. Poiché la CNN è un algoritmo supervisionato, durante l'addestramento è necessario assegnare esplicitamente un'etichetta alle finestre temporali (o fette temporali) del dataset. Tutti i segmenti appartenenti alle scariche regolarmente terminate sono stati etichettati come "stabili". Per ogni scarica disrotta, l'etichettatura di "instabile" è stata effettuata identificando automaticamente la fase pre-disruttiva mediante l'algoritmo proposto in [1]. Le sue prestazioni in predizione sono state valutate utilizzando scariche disrotte e regolarmente terminate di un decennio di campagne sperimentali JET, dal 2011 al 2020, dimostrando la robustezza dell'algoritmo. Inoltre, le prestazioni del predittore di disruzioni proposto sono state confrontate, sugli stessi set di test, con quelle del predittore GTM [1] e di una rete neurale Fully Connected (FC), dimostrando il vantaggio della CNN nell'elaborazione dei dati dell'intero profilo. Tuttavia, uno studio dell'evoluzione delle prestazioni nelle diverse campagne ha rivelato l'invecchiamento del predittore, con un degrado dell'accuratezza, soprattutto nel tasso di falsi allarmi. Infatti, durante gli esperimenti ad alta potenza di JET 2020, i ricercatori hanno osservato la comparsa di radiazioni localizzate nel Low Field Side (LFS). Poiché il predittore precedentemente sviluppato analizzava solo le informazioni provenienti dalla telecamera orizzontale del bolometro, in questi casi la CNN non era in grado di localizzare correttamente la sorgente di radiazioni, innescando così un falso allarme. Questo fatto ha motivato lo sviluppo di un nuovo predittore basato sulle CNN, in cui la telecamera verticale del bolometro viene aggiunta all'insieme di caratteristiche 1-D del profilo del plasma [4]. Quindi, due diversi classificatori CNN, i cui parametri liberi sono ottimizzati per ottenere le migliori prestazioni complessive del predittore, sono addestrati per rilevare diversi eventi destabilizzanti. Il rilevamento automatico di specifici meccanismi destabilizzanti viene utilizzato per addestrare le architetture CNN [1,2,4,5].

Il monitoraggio e la limitazione in tempo reale del flusso di calore sulle piastrelle del divertore è un obiettivo fondamentale per il funzionamento a lungo termine della fusione ad alte prestazioni. Per questo motivo, sono necessari diversi strumenti diagnostici per monitorare lo stato del dispositivo durante gli esperimenti. A questo proposito, una delle questioni fondamentali per i dispositivi di fusione magnetica è garantire l'integrità dei componenti di prima parete (*Plasma Facing Components* o PFCs) durante il funzionamento ad alte prestazioni. A W7-X, le telecamere a infrarossi (IR) monitorano i componenti di prima parete misurando la loro temperatura superficiale. In genere, il flusso di calore è localizzato su specifiche regioni ad alto carico del divertore, chiamate *strike lines*. Poiché valori elevati di flusso di calore localizzato possono danneggiare i PFCs, molti sforzi sono dedicati alla stima e al controllo del flusso di calore sulle tegole del divertore. Per quanto riguarda la stima, a partire dalla temperatura misurata sulla superficie del target, la distribuzione della temperatura interna può essere calcolata risolvendo

l'equazione di diffusione del calore. Diversi codici di ricostruzione del flusso di calore sono stati sviluppati seguendo questo approccio, come THEODOR, attualmente impiegato a W7 X per l'analisi dei dati offline [Y. Gao et al., Nucl. Fusion 51 (2019) 106017]. Tuttavia, ai fini del controllo del carico termico, sono necessari schemi di calcolo veloci per la stima del flusso di calore in tempo reale. Nell'ambito della borsa di ricerca finanziata dall'associazione DAAD, parte del lavoro di dottorato è stato dedicato al refactoring e all'ottimizzazione del codice THEODOR, con l'obiettivo di velocizzare i tempi di calcolo e renderlo compatibile con un utilizzo in tempo reale. Questo lavoro ha portato a velocizzare THEODOR di un fattore 10 su un computer di test. Tuttavia, poiché il computer disponibile in tempo reale poteva dedicare solo le unità di elaborazione grafica (GPU) a questa operazione, il codice THEODOR originale non era adatto a essere eseguito in tempo reale. Per questo motivo, è stato proposto un modello di Physics Informed Neural Network (PINN) per accelerare il calcolo del flusso di calore verso l'implementazione in tempo reale [3]. Una PINN è essenzialmente una rete neurale (NN) tradizionale, in cui una parte della funzione di errore, o *Loss Function*, vincola la rete a rispettare una legge fisica, sotto forma di equazione differenziale ordinaria o parziale. Questo fatto rende l'architettura della PINN piuttosto flessibile, poiché è possibile sfruttare molte architetture di reti neurali tradizionali, anche se la rete Feed-Forward è spesso utilizzata per la sua semplicità. Le PINN presentano diversi vantaggi rispetto agli altri risolutori numerici di PDE: possono essere utilizzate per regredire operatori di PDE non lineari; sono prive di mesh e possono gestire domini irregolari; sono in grado di sfruttare le capacità di calcolo parallelo delle GPU. In questa tesi, viene proposta un'architettura Feed-Forward NN per ricostruire la distribuzione di temperatura all'interno della tegola e il flusso di calore sulla superficie di un tipico profilo del divertore dello Stellarator W7X.

Introduction

Controlled nuclear fusion power is an option to produce clean, safe, scalable, and non-intermittent energy. In particular, magnetic confinement fusion is the most investigated approach for energy production, and Tokamaks and Stellarators are by far the most promising concepts for the future fusion reactors. Both devices confine the plasma with strong magnetic fields and produce helicoidal magnetic field lines. Tokamaks are pulsed operation machines where the poloidal magnetic field, necessary for the confinement, is obtained by inducing a toroidal current in the plasma, while in Stellarators the coils have a complex geometry to directly generate the helicoidal field. This difference determines how the two devices are operated, and the advantages and disadvantages of each design.

The operation of both devices requires a deep understanding of plasma physics and the careful planning of the control parameters. In fact, there are two main objectives which should be matched while running experiments in the current devices: the achievement of high plasma performance and the preservation safeguard of the integrity of the machine. In fact, while researchers aim to achieve the best plasma conditions in terms of temperature, density and discharge length, the interaction with the plasma can damage the components, forcing the operation to stop, reducing the available experimental time in the actual devices, and leading to a costly repair of the device.

One of the main disadvantages of Tokamaks is that the toroidal current makes the devices subject to disruptions. The disruption is the abrupt loss of the plasma current and releases huge electromechanical and thermal forces on the walls of the device. Since disruptions could cause severe damage to the plasma facing components, a lot of effort is directed to the identification of the precursors, the causes, and the consequences of Tokamak disruptions; the final goal is the development of automatic schemes and strategies to mitigate or avoid disruptions. There are studies where disruptions are classified by identifying the sequences of events that lead to the disruption, and others dedicated to the development of routines and algorithms capable of detecting specific events correlated to disruptions. In literature, both physics-based and data-driven approaches are implemented for predicting and classifying disruptions. The physics-based methods have the advantage of being directly interpretable and more scalable among different devices, but self-consistent and general physical models which can be run in real-time are not yet available. Instead, data-driven approaches can exploit the large amount of data available from the experiments and are considered as an alternative approach to disruption prediction. Both statistical, and Artificial Intelligence (AI) methods, such as Machine Learning and Deep Learning, have been investigated to this purpose.

On the other hand, the Stellarator can be operated in steady state and the effects of confinement losses on the vessel structure are negligible. However, due to the long operation times achievable with this configuration, a lot of efforts are devoted to the prevention of overloads in the first wall of these devices. For this purpose, an intense research activity aims to detect thermal events at Weldenstein 7-X (W7X) stellarator, where infrared cameras monitor the state of the first wall during the experiments and a fully automatic interlock system is under development to stop the operation if an overload is detected. Especially in Stellarators, due to their 3D geometry, a complex interplay of magnetic topology in the island boundary, local shaping of the Plasma Facing Components (PFC), ratio between parallel- and cross-field transport determines the heat flux distribution patterns. To analyse and control the heat flux distribution on the first wall, there is the need to develop algorithms able to reliably estimate the heat flux from the temperature in real-time.

This thesis discusses the use of Artificial Intelligence methods for the protection of the nuclear fusion devices with reference to the Joint European Torus (JET) tokamak situated in Culham, UK and the Weldenstein 7-X (W7X) stellarator, in Greifswald, Germany. Both devices are part of the EUROfusion program for the development of nuclear fusion research. JET is currently the largest operating tokamak in the world and the only one which can run with the Deuterium-Tritium fuel, while W7-X is the largest and most advanced stellarator in the world, with the aim to investigate the possibility of a stellarator reactor path to fusion commercial reactors.

Concerning JET, the existing database maintained by the University of Cagliari, has been updated with the discharges coming from the JET experimental campaigns from 2016 to 2020, focusing on the C36 (2016) [1] and the C38 campaigns (2019-2020) [2]. Both disrupted and regular terminated discharges have been selected from experimental campaigns performed at JET, after the installation of the ITER-Like Wall. In total, the database for this work contains a total of 198 disrupted and 219 regularly terminated discharges having a flat-top plasma current higher than 1.5 MA, and a flat-top length greater than 200 ms. The analysis of the pulses refers to the flat-top phase, and the flat-top starting time has been assumed as the first time instant where the plasma is in X-point configuration. In the pulses of the database from the C28-C30 experimental campaigns (2011-2013), the authors of [A. Pau, et al., Nucl. Fusion 51 (2019) 106017] manually identified the so-called pre-disruptive time of a disruption, which provides a reference time to separate the plasma current flat-top of each disrupted discharge into two parts: a non-disrupted part and a pre-disrupted part. This second part is defined as the phase where the chain of events leading to the disruption occur.

The introduction of consistent pre-disruptive times ($t_{pre-disr}$) is doubly beneficial. Firstly, these times allow to identify the pre-disruptive phase, which is used to describe the disrupted input space of whatever AI predictive models. Secondly, being the pre-disrupted time strongly linked to the onset of destabilizing

phenomena, the predictor response should be connected to phenomenology or precursors that characterize the various types of disruptions. In most of the literature, this pre-disruptive phase was statistically or heuristically identified and assumed equal for all the disruptions in the database, introducing contradictory information in the prediction model. The key to a successful prediction model is therefore the capability, for each disrupted discharge in the training set, to discriminate among the non-disrupted and the pre-disruptive phases following standard and coherent criteria, linked to the observed physical mechanisms. However, this classification requires a very time-consuming manual analysis; hence, adopting it to classify tens of thousands of shots would be highly impractical. Therefore, during the Ph.D., an algorithm for the automatic identification of the pre-disruptive times has been developed, based on a statistical approach [1]. The histograms have been used here to estimate the probability density function (PDF,) relying on the fact that a histogram of a measurement provides the basis for an empirical estimate of the PDF. Several approaches can be used to quantify how similar/dissimilar two histograms are.

In the proposed approach, the two histograms are considered as multidimensional vectors, and the similarity/dissimilarity of two histograms (or PDFs) is evaluated as the distance between vectors. Several metrics are available to evaluate the geometric distance measure, such as the straightforward L1-norm or L2- norm functions, or those belonging to the intersection or inner product families. The cosine metric, belonging to the latter family, has been used here. The dissimilarity is evaluated for several plasma parameters and then an optimal weighted sum is assumed as the overall dissimilarity. An optimal criterion has been introduced to automatically choose, for each discharge, the pre-disruptive time over this total dissimilarity measure. The algorithm is based on the statistical analysis of the plasma parameters selected from the JET experimental campaigns performed from 2011–2013, and provides, for each disrupted discharge, the pre-disruptive time ($T_{pre-disr,AUT}$). These times have been compared with the ones ($T_{pre-disr,MAN}$) manually identified in [A. Pau, et al., Nucl. Fusion 51 (2019) 106017], in terms of the performance of prediction models based on machine learning Generative Topographic Mapping (GTM) [C.Bishop, et al., *Neural Comput.* (1998) **10**(1), 215–234]. Moreover, the proposed algorithm has been applied to the updated JET database, and the performance of the updated GTM model confirms the suitability of the algorithm [1,6].

Another aspect related to the data-driven AI models concerns the use of informative features linked to the physical phenomena observed experimentally. In particular, invaluable information can be obtained from temperature, density and radiation plasma profiles due to their close connection with plasma stability and destabilization of MHD modes that may cause disruptions. Just as an example, in most cases, a disruption is the consequence of the development of tearing modes inside the plasma, which leads to the growing of the magnetic islands. Usually, well

before the onset of the tearing modes, an increase of the radiation emission in the core, which leads to a hollow temperature profile, can be observed, whereas an increase in the radiation emission at the edge of the plasma leads to a cooling at the edge. In case of temperature hollowing, there is a broadening of the current density profile from inside, whereas a shrinking of the same profile from outside corresponds to the edge cooling. In both cases an unstable MHD scenario may arise due to a continuous increase of the current density gradient near the mode resonant surface. On the other hand, the spatiotemporal information contained in the plasma profiles is crucial in describing destabilizing localized phenomena, such as the radiation emission in the core rather than at the edge, which cannot be enough described by zero-dimensional (0-D) time varying parameters, as the radiated fraction of the total input power. The time dimension is neglected when considering the data dimension since the data-driven model will receive a single instance of the signal at once.

To this end, 0-D peaking factor time variant signals have been synthesized to encode the spatial information contained in the one-dimensional (1-D) profiles, through the ratio between the mean values of measurements over different regions of the plasma cross section. The peaking factor signals constructed starting from temperature, density and plasma radiation profiles, and therefore well anchored to the plasma physics, demonstrated to increase the performance of the machine learning GTM models predicting disruptions with enough warning time to more efficiently enable avoidance strategies. However, the definitions of such peaking factors are based on heuristics that arbitrarily assume the 'core' chords and the 'divertor' chords and can lose precious spatial information contained in the plasma profiles. Moreover, the peaking factor definitions must be changed depending on the different diagnostic systems available in the different devices. Recently, Convolutional Neural Networks (CNNs), belonging to the Deep Learning paradigm, have proved capable of overcoming the most established machine learning techniques, especially in the field of image processing and computer vision, for their ability to learn relevant features from images at different scales, avoiding hand-engineered feature extraction. In this dissertation, CNNs are proposed both to extract the spatiotemporal features from the plasma profiles of temperature, density and plasma radiation, overcoming the previously described limits of the 0-D peaking factors, and to develop a quite simple deep neural network disruption prediction model that uses these features together with other diagnostic signals commonly used in the literature. The deep-CNN predictor has been trained using data from experimental campaigns performed at JET from 2011 to 2013. As the CNN is a supervised algorithm, during the training, a label must be explicitly assigned to the time windows (or time slices) in the dataset. All the segments belonging to the regularly terminated discharges have been labelled as 'stable'. For each disruptive discharge, the labelling of the 'unstable' has been carried out by automatically identifying the pre-disruptive phase by means the algorithm proposed in [1]. Its prediction performance has been evaluated using disrupted and regularly

terminated discharges from a decade of JET experimental campaigns, from 2011 to 2020, showing the robustness of the algorithm. Moreover, the performance of the proposed disruption predictor has been compared, on the same test sets, with the performance of the GTM predictor [1] and of a Fully Connected (FC) neural network, demonstrating the advantage of the CNN in the processing of the entire profile data. However, a study of the performances evolution over the different campaigns revealed the predictor ageing, with an accuracy degradation, mainly in the false alarm rate. Indeed, during the 2020 JET high power experiments, researchers observed the appearance of localized radiation in the Low Field Side (LFS). Since the previously developed predictor was only analyzing the information from the bolometer horizontal camera, the CNN could not correctly locate the radiation source in these cases, hence triggering a false alarm. This fact motivated the development of a new predictor based on CNNs, where the vertical bolometer camera is added to the set of 1-D plasma profile features [4]. Then, two different CNN classifiers, whose free parameters are optimized to achieve the best overall predictor performance, are trained to detect different destabilizing events. The automatic detection of specific disruptive mechanisms is used to train the CNN architectures [1,2,4,5].

Monitoring and limiting the heat flux on the divertor tiles in real-time is a key objective for long high-performance fusion operation. For this reason, several diagnostics are needed to monitor the state of the device during the experiments. In this regard, one of the fundamental issues for magnetic fusion devices is to ensure the integrity of the PFCs during high-performance operation. At W7-X, infrared (IR) cameras monitor the plasma facing components (PFCs) by measuring their surface temperature. Typically, the heat flux is localized on specific high load regions of the divertor called strike lines. Since localized high heat flux values can damage the PFCs, a lot of effort is devoted to the estimation and control of the heat flux on the divertor tiles. Regarding the estimation, starting from the measured temperature at the target surface, the internal temperature distribution can be computed by solving the transient heat conduction equation. Several heat flux reconstruction codes have been developed following this approach, such as THEODOR which is currently employed at W7-X for offline data analysis [Y. Gao et al., Nucl. Fusion 51 (2019) 106017]. However, for heat load control purposes, fast computation schemes for the real-time heat flux estimation are required. In the framework of the Research Grant funded by the DAAD association, part of the Ph.D. work was dedicated to the refactoring and optimization of the THEODOR code, with the aim of speeding up the computation time and to make it compatible with a real-time use. This work led to the speed up of THEODOR by a factor of 10 on a test computer. Nevertheless, since the real-time computer available in real-time could only dedicate the Graphical Processing Units (GPU) for this operation, the original THEODOR code was not suitable to be run in real-time. For this reason, a Physics Informed Neural Network (PINN) model is proposed to speed up the heat-flux computation towards the real-time implementation [3]. A PINN is essentially a traditional Neural Network (NN),

where a part of the loss function constrains the network to respect a physic law, in the form of an Ordinary or Partial Differential Equation. This fact makes the architecture of the PINN quite flexible, since many traditional NN architectures can be exploited, even though the Feed-Forward NN is often used due to its simplicity. PINNs have several advantages with respect to the other numerical PDE solvers: they can be used to regress nonlinear PDE operators; they are mesh-free and can handle irregular domains; they are able to exploit the parallel computing capabilities of GPUs. In this thesis, a Feed-Forward NN architecture is proposed to reconstruct the temperature distribution in the bulk and the heat-flux on the surface of a typical divertor profile of W7X stellarator.

List of Publications related to this work

[1] E. Aymerich, A. Fanni, G. Sias, S. Carcangiu, B. Cannas, A. Murari, A. Pau, A statistical approach for the automatic identification of the start of the chain of events leading to the disruptions at JET, Nucl. Fusion. 61 (2021) 036013. https://doi.org/10.1088/1741-4326/abcb28.

[2] E. Aymerich, G. Sias, F. Pisano, B. Cannas, S. Carcangiu, C. Sozzi, C. Stuart, P. Carvalho, A. Fanni, Disruption prediction at JET through Deep Convolutional Neural Networks using spatiotemporal information from plasma profiles, Nucl. Fusion. (2022). https://doi.org/10.1088/1741-4326/ac525e.

[3] E. Aymerich, F. Pisano, B. Cannas, G. Sias, A. Fanni, Y. Gao, D. Böckenhoff, M. Jakubowski, 'Physics Informed Neural Networks towards the real-time calculation of heat fluxes at W7-X', Nucl. Mater. Energy, vol. 34, p. 101401, Mar. 2023, doi: 10.1016/j.nme.2023.101401.

[4] E. Aymerich, G. Sias, F. Pisano, B. Cannas, A. Fanni, and the-JET-Contributors, 'CNN disruption predictor at JET: Early versus late data fusion approach', Fusion Eng. Des., vol. 193, p. 113668, Aug. 2023, doi: 10.1016/j.fusengdes.2023.113668.

[5] E. Aymerich, B. Cannas, F. Pisano, G. Sias, C. Sozzi, C. Stuart, P. Carvalho, A. Fanni, 'Performance Comparison of Machine Learning Disruption Predictors at JET', Appl. Sci., vol. 13, no. 3, Art. no. 3, Jan. 2023, doi: 10.3390/app13032006.

[6] E. Aymerich, S. Carcangiu, Cannas, Barbara, A. Fanni, Sias, Giuliana, Murari, Andrea, Pau, Alessandro, Continuous update of machine learning disruption prediction and prevention models at JET, in: (Virtual) Technical Meeting on Plasma Disruptions and their Mitigation, 2020.

[7] E. Aymerich, G. Sias, B. Cannas, A. Fanni, Evolution of Data-driven Disruption Prediction: from Machine Learning to Deep Learning, in: Second Technical Meeting on Plasma Disruptions and their Mitigation, 2022, ITER Headquarters, Jul 19 – 22, 2022

Organization of the thesis

The thesis is organized as follows.

Part 1

Chapter 1 provides a background on nuclear fusion energy, and on the differences between the Tokamak and Stellarator designs. Moreover, it links the characteristics of these devices to the problems addressed in this thesis.

In Chapter 2, the machine learning and deep learning methodologies used in this thesis are explained in detail. The large quantity of available data from fusion experiments motivates the use of data-driven approaches.

Part 2: Disruption prediction at JET

Chapter 3 details the disruption prediction problem and the state of the art. Over many years, several techniques, either physics based or data-driven, have been developed for disruption prediction in tokamaks. This chapter provides an overview of the approaches in the literature highlighting the advantages and disadvantages of each one.

In Chapter 4, the JET database used for the disruption prediction is discussed, describing the available diagnostics and signals and the number of discharges belonging to the disrupted and non-disrupted classes.

Chapter 5 introduces the disruption prediction (DP) approach using the machine learning GTM method, presents the algorithm for automatically determining the pre-disruptive phase, and reports the performance of the GTM DP models trained using the automatic identification of the pre-disruptive phase.

Chapter 6 proposes a feed forward neural network model for DP and discusses its performance on the full database.

Chapter 7 proposes CNN models for disruption prediction and discusses their performance. Early and late fusion of features are compared, and different diagnostics combinations are tested.

Chapter 8 reports the comparison of the different AI models on the same training and test databases, and the same features, based on a wide set of performance metrics. This Chapter draws some conclusions on the metrics used to compare the different disruption predictors and on their main advantages and disadvantages.

Part 3: Heat-Flux computation at W7-X

In Chapter 9, the problem and the adopted methods applied for monitoring the thermal loads at W7-X are presented. The automatic system for real-time overload detection is described.

Chapter 10 discusses the computation of heat-fluxes at W7-X with the THEODOR code, together with its optimization and the bottlenecks for the real-time implementation.

Chapter 11 discusses the development of a PINN approach for the solution of the PDE heat equation, to port the heat-flux computation on a GPU in view of a real-time application.

Finally, in the Conclusions the work is summarized, and some future research directions are given.

Contents

List of Figures

List of Tables

Part 1: Nuclear fusion and Methods

Chapter 1

Fusion and magnetic confinement

1.1 What is Fusion

Fusion is a form of nuclear energy that powers the Sun and the stars and has the potential to provide an *almost* unlimited source of energy for the Earth. Fusion represents an attractive source of energy for a list of reasons:

- **Environmental sustainability:** The fuels used in a typical fusion power plant, water and lithium, are clean and environmentally sustainable not producing atmospheric pollution as the greenhouse gases. In the middle of the process a radioactive isotope of Hydrogen, Tritium, is produced, but its half-life is 12 years, compared to the 10^9 years of Plutonium
- **Fuel supply**: The necessary fuels are also particularly abundant in the Earth, such that their supply will not represent a problem in the future
- **Limited risk of proliferation**: Fusion doesn't employ fissile materials like uranium and plutonium. (Radioactive tritium is neither a fissile nor a fissionable material). There are no enriched materials in a fusion reactor like ITER that could be exploited to make nuclear weapons.
- **No risk of accidents**: A meltdown nuclear accident is not possible in a tokamak fusion device. Reaching and maintaining the necessary conditions for fusion is difficult, and if the plasma is destabilized, it cools suddenly stopping the reaction. Moreover, the quantity of fuel present in the vessel at any one time is enough for a few seconds only and there is no risk of a chain reaction.

A nuclear fusion reaction produces energy for the same reason nuclear fission does. The mass defect is related to the energy released when the nucleus is formed according to the well-known Einstein law:

$$E = \Delta m \cdot c^2 \tag{1-1}$$

Therefore, energy is released when two lighter atoms join to form a heavier one, as in the case of fusion, or if a very heavy atom splits to form lighter fragments in a fission process. Figure 1.1 shows a graph where the nuclear binding energy (which keeps the components of the atom nucleus together) is plotted with respect to the mass of the element. We can also notice that, from the Iron (Fe), the fusion process does not produce any more energy: this is exactly the reason why stars stop burning when only the iron core remains [http://abyss.uoregon.edu/~js/ast122/lectures/lec18.html]. On Earth, conditions for fusion are extremely hard to achieve. Low atomic number elements, as hydrogen and its isotopes, must be heated to very high temperatures (around 100 million degrees Celsius) for reaching the right conditions for fusion. When these conditions are met, gas mixture evolves into another state of the matter named plasma, where the

negatively charged electrons are separated from the positively charged atomic nuclei (ions). The extremely hot temperatures are necessary so that the nuclei receive enough energy to overcome the electrostatic repulsive forces, allowing fusion between the nuclei (due to attractive nuclear forces) and the resulting release of energy.

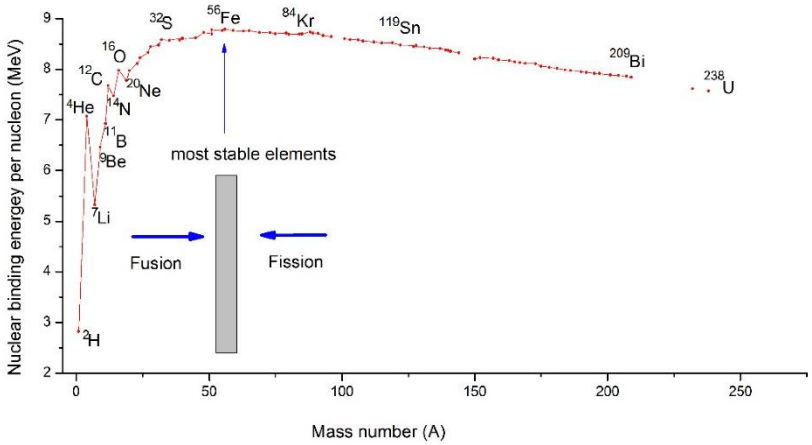

Figure 1.1: Nucleon binding energy in function of their mass number (A) [1]

Fusion energy has the potential to provide substantial amounts of baseload electricity, and this would have an enormous impact on the economy and society overall. Nowadays, there are already thermonuclear magnetic confinement fusion experiment which are operated all over the world, but in order to make fusion a commercial energy source, several critical issues must be addressed from the technological and engineering point of view. In fact, this is the purpose of next generation of fusion reactors such as ITER and DEMO, which are among the most challenging scientific experiments of the upcoming future. The goal is the development of proper technologies to demonstrate the technical and the economic feasibility of a fusion power plant, which provides energy to electric grid. In Figure 1.2 a schematic representation of the DEMO power plant is reported. Deuterium (D) and Tritium (T) fuel burn at an extremely hot temperature in the central reaction chamber. The energy is released as charged particles, neutrons, and radiation, and it is absorbed in a Lithium blanket surrounding the reaction chamber, where the neutrons react to convert the lithium into tritium fuel. Energy is then generated

using the heat of the reaction in a conventional steam-generating plant. The waste product from the nuclear reaction is helium.

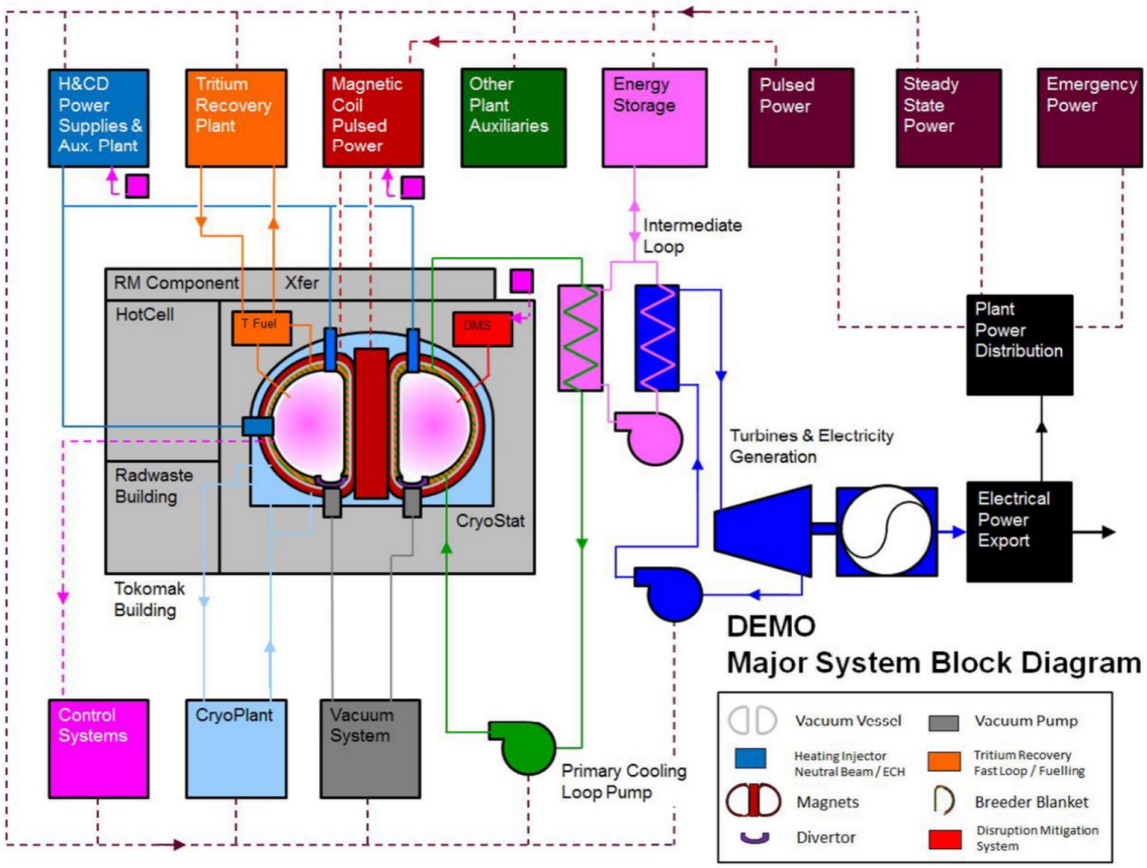

Figure 1.2: Schematic of a DEMO power plant [2]

1.2 Basis of Fusion reaction

The strong interest in fusion reactions is motivated by political, economic and environmental considerations regarding the use of fossil fuels for the energy production. Moreover, the fusion energy has an enormous potential in terms of produced energy with respect to other fuels and sources of energy. In fact, the energy produced with 0.14 tons of Deuterium by fusion reactions is equivalent to the one produced by burning 106 tons of fossil oil or 0.8 tons of Uranium by nuclear fission. At the moment, the most relevant fusion reactions are the D-D and the D-T reactions. The D-D reaction produces energy by the nuclear interaction between two deuterium nuclei according to the two equally likely reactions:

$$D + D \rightarrow {}^{3}He + n + 3.27\ MeV \tag{1-2}$$

$$D + D \rightarrow T + p + 4.03\ MeV \tag{1-3}$$

Instead, in the D-T reaction a Deuterium and Tritium nuclei interact, as shown in Figure 1.3. This reaction is the one considered for the next generation devices such as ITER, because it is the one with the highest likelihood of occurrence. The D-T reaction can be written as:

$$D + T \rightarrow \alpha + n + 17.6\,MeV \tag{1-4}$$

This reaction produces 17.6 MeV of energy, which is released in the form of kinetic energy associated in part with the neutron (14.1 MeV) and in part with the alpha particles (3.5 MeV). The goal in fusion is the confinement of the alpha particles within the plasma, so that their energy can be transferred by collisions to plasma ions and electrons. In this case, the reaction would release 3.5 MeV per nucleon, oppositely to the 1.01 MeV per nucleon of the D-D reaction.

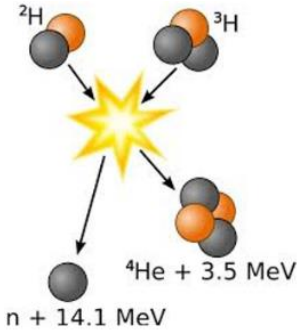

Figure 1.3: Deuterium-Tritium reaction.

Unfortunately, the high energetic neutrons may activate the vessel materials, and the reaction requires the breeding of the Tritium. In fact, Tritium is radioactive, it undergoes beta decay with a half-life (approximately 12.5 years) and is not naturally present on Earth. Nevertheless, the high likelihood of occurrence with respect to the others, makes this reaction the main option under study for future fusion reactions. In Figure 1.4 it is possible to confirm that the D-T reaction has a much higher probability of occurring, especially at energy below 100 keV. Note that the current record in terms of sustained operation of a nuclear fusion device has been set in KSTAR, where researchers have been able to sustain an ion temperature of 10 keV (corresponding to 120 million degrees Celsius) for tens of seconds [3]. For that range of temperatures, the probability for the D-T reaction to take place is much higher than for the other reactions.

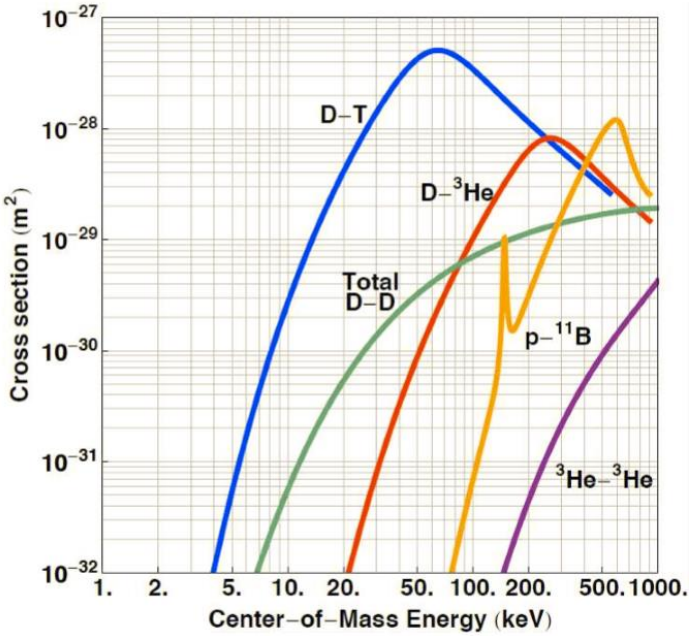

Figure 1.4: Cross sections versus center-of-mass energy for key fusion reactions. [from http://iec.neep.wisc.edu/]

On the other hand, for the D-T reaction, Tritium must be bred in the fusion power plant, by capturing neutrons in Lithium. When a neutron interacts with Lithium, the primary reactions through which Tritium can be produced are the following:

$$Li_6 + n(slow) \rightarrow \alpha + T + 4.8\,MeV \quad (1\text{-}5)$$

$$Li_7 + n(fast) \rightarrow T + \alpha + n - 2.5\,MeV \quad (1\text{-}6)$$

Both reactions give rise to the production of Tritium, although one releases energy whereas the second one consumes it. However, the reaction with Li_7 is particularly important as it releases another neutron, which would enable the self-sufficient Tritium production in a fusion reactor. This is a critical issue, since the fusion devices are designed to satisfy a specific Tritium budget which depends on the amount of space left for the Lithium blanket, where Tritium is bred. The availability of Li_7 is larger, but the reaction related to Li_6 has a higher likelihood of occurrence. Therefore, the first one is the reaction which dominates in the breeding of Tritium.

1.3 Magnetic confinement fusion

Magnetic confinement and inertial confinement are the two different approaches studied for nuclear fusion. The first approach confines the plasma with strong magnetic fields, whereas, in the second one, small pellets containing fusion fuel are compressed to extremely high densities through strong lasers or particle beams. Inertial confinement fusion recently demonstrated, for the first time, the possibility of having a net energy gain from a fusion reaction, when on 5th December 2022 “NIF researchers fired 2 MJ of energy at a fuel target and recorded a fusion energy release of just over 3 MJ” [4]. Even though the net energy gain does not take into account the efficiency of the energy transfer from the power supply to the laser

system and then to the plasma, this result is quite remarkable and marked a milestone for nuclear fusion research.

Regarding magnetic confinement, the widely investigated concepts are tokamaks (together with spherical tokamaks), stellarators, reversed field pinches, spheromaks and levitated dipoles [5]. All the machines are basically 2-D axisymmetric toroidal configurations, except the stellarator, which is an inherently 3-D configuration. Among all the configurations, tokamaks have achieved the best overall performance, followed by stellarators. In fact, these two configurations are the ones mainly investigated by the EUROfusion research program [https://www.euro-fusion.org/eurofusion/roadmap/]

In magnetic confinement nuclear fusion, the plasma can be treated as a system which is globally neutral but locally charged and interacts with the magnetic field. A particle with charge q moving in a magnetic field with velocity v will be subject to the Lorentz force $\boldsymbol{F} = q(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B})$, where $\boldsymbol{B}$ is the magnetic field, and $\boldsymbol{E}$ is the electric field. This force produces a circular particle motion in the plane perpendicular to the magnetic field line, making the particle spiral along the magnetic field line. In presence of a simple toroidal field, the magnetic field curvature and gradient cause the particles to drift. Since the drift force depends on the particle charge, the ions and electrons velocities $\boldsymbol{v_{d.i}}$ and $\boldsymbol{v_{d.e}}$ respectively will have opposite signs, determining an electric field in the vertical direction, as shown in Figure 1.5. Finally, the electric field produced by the charge separation causes an outward $\boldsymbol{E} \times \boldsymbol{B}$ drift of the plasma particles. Hence, the toroidal magnetic field (B_{tor}) cannot confine the plasma by itself, and an additional poloidal magnetic field component (B_{pol}) is necessary. The configuration produces helical magnetic field lines around the torus and the typical magnetic surfaces, which are surfaces with constant magnetic flux.

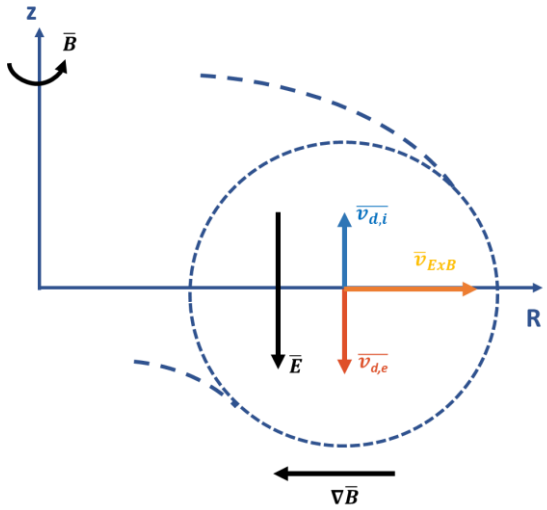

Figure 1.5: Particles drift in a toroidal configuration

1.3.1 The tokamak

In a tokamak, the toroidal field coils produce B_{tor}, whereas B_{pol} is due to the inductive toroidal current flowing in the plasma. To induce the plasma current, a central solenoid acts as the transformer primary where the plasma is the transformer secondary, so that the plasma current is ramped up by a transformer effect. Figure 1.6 reports a schematic of the configuration. The typical plasma current values in JET are of few Mega-Amperes (MA). This configuration has the following drawbacks:

- The tokamak is a pulsed operation machine: the central solenoid can induce a current until the core saturates, which means that the core is fully magnetized, and the magnetic flux cannot increase. For this reason, fusion researchers investigate the use of non-inductive current drive by external systems, such as Electron Cyclotron Resonance Heating (ECRH), Ion Cyclotron Resonance Heating (ICRH), Lower Hybrid Waves (LHW) and Neutral Beam Injection (NBI). The non-inductive current drive is fundamental for running long experiments, in view of ITER.
- The toroidal current is strongly coupled with the equilibrium of the system. The loss of the plasma current can determine a disruption and the release of huge electromechanical forces on the walls of the device. The real-time prediction of disruptions allows the operator to mitigate the effects of a disruption or to enable early termination strategies if enough time is available before the final destabilization.

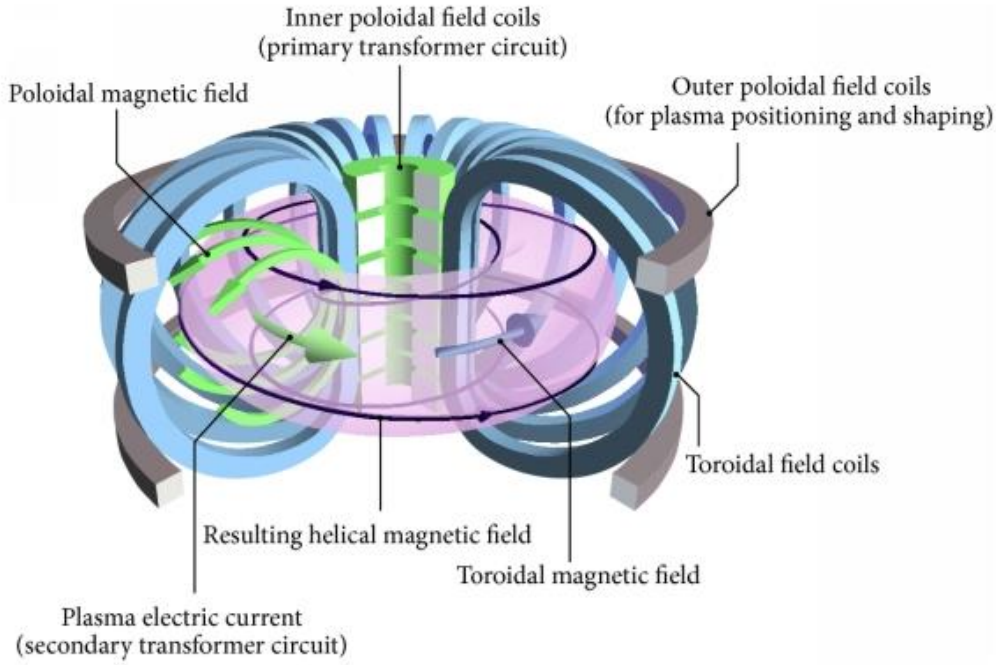

Figure 1.6: Schematic of a tokamak [6]

1.3.2 Disruptions in tokamaks

A disruption is an abrupt loss of the plasma confinement in a tokamak plasma. The sudden loss of plasma confinement, followed by the quench of the plasma current, could result in the release of large amounts of energy and large thermal and electromagnetic loads, possibly causing severe damage to the plasma facing components and stressing the device with high mechanical forces [7]. Therefore, huge efforts are devoted to the task of identifying the precursors, the causes, and the consequences of tokamak disruptions. In particular, the following phases of a disruption have been identified by the community [8], [9]:

1. Pre-precursor phase: there is a change in the operative conditions that leads toward an unstable configuration. This change is often clear, as in the case of an increase of the plasma density or the auxiliary power shutdown when the reactor operates near at the Greenwald density limit. Actually, in a pioneering study of the JET tokamak disruptions, it has been observed that [10], [11]: "As expected, all disruptions at JET were eventually pushed close to an operational limit resulting in the onset of physics instabilities." Unfortunately, due to the complex interplay of phenomena that govern the disruptions, this phase is not always clearly identifiable.

2. Precursor phase: in this phase, the magnetic confinement starts to deteriorate and MHD instability grows, for instance with the development of perturbations of the magnetic field as in the second plot of Figure 1.7.

3. Fast phase/thermal quench: the central temperature collapses.

4. Quench phase/current quench: the plasma current decays to zero.

The disruption is a very complex phenomenon. Often the chain of events that leads to a disruption has numerous root causes and follows a complicate path [10]. Different events and paths can lead to the same disruption type. In the literature, several types or classes of disruption have been identified on the base of the chain of events that leads to the disruption, depending on the operative regime [10], [12].

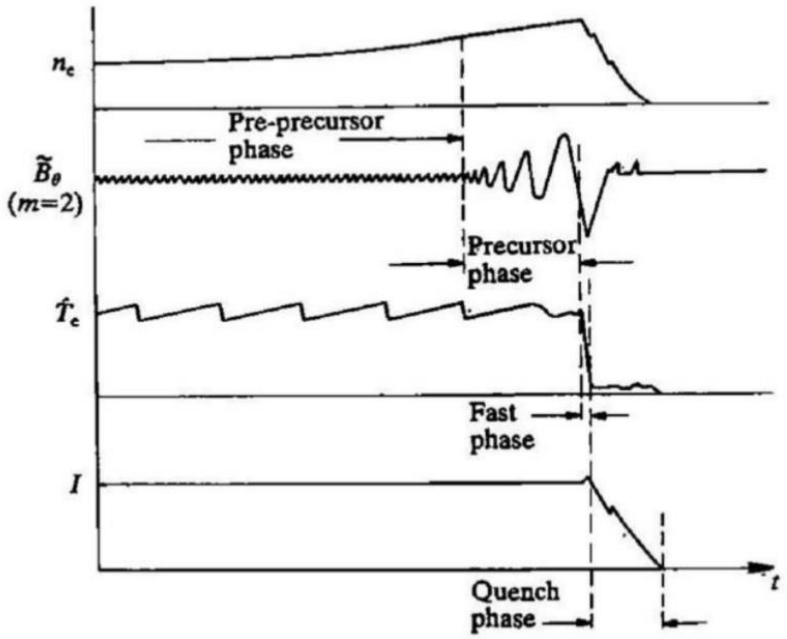

Figure 1.7: Temporal sequence of a disruption [8]

The aim of disruption prediction is the automatic identification of the disruptions, in order to avoid or mitigate the occurrence of disruptions in tokamak plasmas.

For an avoidance action, the early identification of a specific destabilization mechanism is a necessary step to adopt a strategy for recovering the discharge, for instance in the case of the mode stabilization with ECRH [13], or for the early termination of the discharge [14]. Instead, the mitigation aims to lessen the damaged produced by a disruption. In this case, the automatic identification of an incoming disruption triggers the Disruption Mitigation System (DMS), which in the timespan of the tens of milliseconds injects impurities into the plasma, with the aim of radiating most of the plasma energy before the disruption. Mitigating or, even better, avoiding plasma disruptions has become mandatory in view of future nuclear fusion devices. In ITER, for instance, during the operations at full performances, the failure rate in the detection of the current quench (CQ) and the vertical displacement event (VDE), should be less than 1% [15]. The estimated total cost of ITER is around 20 billion euros [16]: this means, of course, that a huge cost is associated with any damage to the machine components.

1.3.3 The stellarator

In a stellarator (Figure 1.8), external coils generate B_{tor} and B_{pol}, which makes the magnetic configuration non-axisymmetric [17]. Instead of having two different sets of toroidal and poloidal field coils, modern stellarators have a complex set of coils. W7-X was designed to be a flexible device, allowing several magnetic configurations. It is equipped with 50 modular non-planar coils and 20 auxiliary planar coils. Because of the periodic and symmetric geometry of the stellarator only

five modular and two auxiliary coil currents are free for variations, i.e. there are seven different currents to produce various magnetic fields and configurations [18]. Moreover, a set of trim coils is used to correct error fields and equalize the power loads among 10 divertors [19].

Apart from the main superconducting nonplanar and planar coils for main magnetic field configuration, a group of 10 control coils, one for each submodule, are situated inside the plasma vessel, behind the baffle and target plates. Their main functions are to control and modify the size and the position of the islands intersecting the divertor at the boundary, correct symmetry-breaking error fields and sweep the strike-line to avoid excessive temperatures. [20] The main advantages of the stellarator are due to the absence of the inductive toroidal current. The configuration can be operated in steady state, but more unconfined particle orbits in stellarators can lead to higher neoclassical transport of energetic and thermal particles than in tokamaks [17].

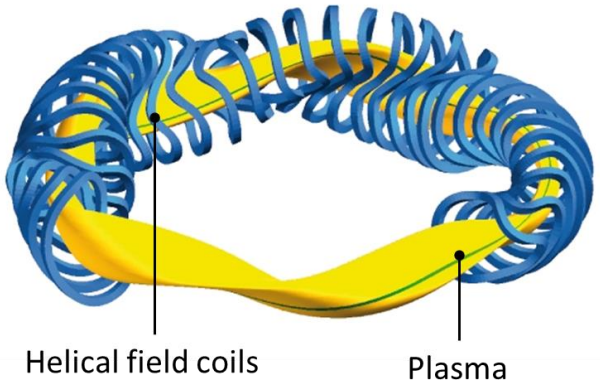

Figure 1.8: Schematic of a stellarator

The interaction of the plasma with the first wall during long experiments may lead to overload of the divertor, and an erroneous magnetic configuration may expose some parts of the first wall to higher heat flux than the one they can withstand.

The geometry of W7-X is 3D helical shape with a five-fold modular symmetry, as in Figure 1.9. Each module has two divertor units (an upper and a lower one) for a total of 10 units. The divertor targets intersect the magnetic islands at the edge to maximize power transfer and particle exhaust removal. Figure 1.10 shows an IR image overlaid to the CAD with the different divertor parts.

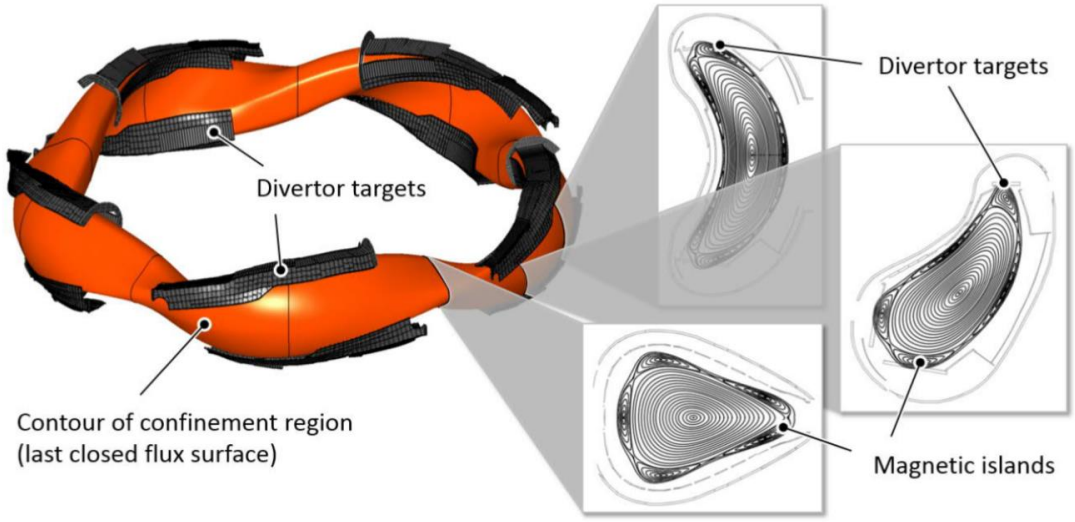

Figure 1.9: W7-X five-fold modules with its 10 divertor units, 5 upper and 5 lower divertor units. The divertor targets intersect the magnetic islands at the edge for power and particle exhaust [21].

The dominant fraction of the energy leaving the confined plasma region is guided towards the divertor targets. Each divertor unit consists of a horizontal and a vertical target and it is designed to sustain a maximum heat flux of up to $10\ MW\ m^{-2}$.

1.3.4 Thermal events and overloads of the first wall in steady state operation

In W7-X the magnetic configuration, plasma parameters and chosen scenario determine the structure of the power deposition pattern. During operation phase OP1.2, ten IR thermographic systems with wide-angle optics were installed to monitor the surface temperature on the fine-grain graphite plates of the ten inertially cooled test divertor units (TDUs) [22]. The activity involved the investigation of the heat fluxes [23], local effects of leading edges [24], error fields [25] and particle drifts [26]. In particular, an intense research activity is devoted to the detection of thermal events carried out at W7-X, where infrared cameras monitor the state of the first wall during the experiments and a fully automatic interlock system is being developed to stop the operation if an overload is detected. Thermal events are due to the presence of temperature peaks in the machine. In terms of image processing, they are defined as a cluster of pixels which have a temperature significantly higher than the nearby ones. However, the events are different from each other, and they can be classified depending on their size, shape, location or on the cause of their appearance.

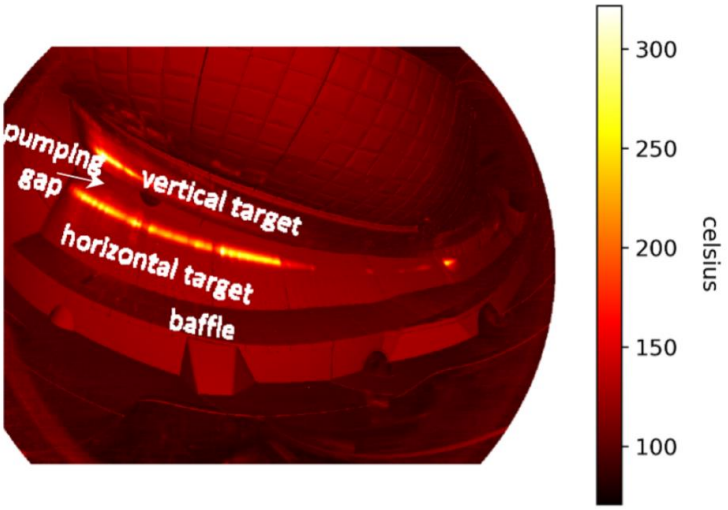

Figure 1.10: Example of IR image of the divertor, where the different parts of the wall are identified.

The classification of these different events is a complicated and cumbersome work, also due to the presence of different causes for the same event. At W7-X, a system to detect thermal events in real-time and timely interrupt operation in the case of a critical event is under development [21]. The fast reaction times required for damage prevention need the creation of fully automatic image analysis algorithms.

Identifying and controlling the regions of highest thermal load (strike lines) on the targets is a key challenge in view of the future fusion power plants, which will have a long operation time. The observation confirmed that a complex interplay of magnetic topology in the island boundary, local shaping of the PFCs, ratio between parallel- and cross-field transport determines the strike lines properties. Actuators such as electron cyclotron current drive (ECCD) and control coils were tested in the first divertor campaign for an active control of power distributions on the divertor.

Chapter 2
Methods

2.1 Introduction

Fusion experiments are complex, and a large set of diagnostics is available to assess the plasma state, understand the physics and protecting the machine. The huge amount of data produced by the diagnostics is difficult to handle, but it provides an opportunity to develop data-driven models for a variety of tasks. Literature proposes a plethora of data-driven approaches, many of them belonging to the wide area of Artificial Intelligence (AI) techniques in general, and to Machine Learning (ML) and Deep Learning (DL) methods, in particular. In this thesis, ML and DL models are developed and applied to two different tasks: disruption prediction and heat-flux computation.

In the disruption prediction case, the goal is to develop models able to detect the onset of an instability, so that an action can be taken in response to the plasma destabilization, thus avoiding the disruption or mitigating its effects. In this case, despite the progress in the modelling of the disruption physics, a general physical model, able to predict the disruption occurrence, does not exist. Hence, the objective is to be able to construct models which can recognise the disruption behaviour from the experimental data. This work will focus on manifold mapping with Generative Topographic Mapping (GTM) [27], and deep learning with Artificial Neural Networks (ANNs) [28] and Convolutional Neural Networks (CNNs) [29]. Moreover, since these algorithms require the construction of labelled databases to train the models, the development of codes able to automatically build such databases is pivotal.

On the other hand, in the case of the heat-flux computation, codes able to reconstruct the surface heat flux on the divertor tiles exist, but these codes are not sufficiently fast to be run in real-time. Nevertheless, the knowledge of the physics of the problem can be exploited to build reliable and fast models, informed by the physics. In this thesis, the proposed DL approach is based on the so-called Physics Informed Neural Networks (PINNs) [30].

In the following paragraphs, the fundamental of the Machine Learning and Deep Learning methods, used in this thesis, have been synthesized.

2.2 Generative Topographic Mapping

The GTM [27] belongs to the class of the so called "generative models", which are based on the idea that the data of interest lies on a low-dimensional manifold, embedded in the high-dimensional space. Generative models try to model the distribution of the data by defining a density model with low intrinsic dimensionality in the data space. The method preserves the topology of the input data space: this

means that points close to each other in the data space will be mapped still close in the latent space. Through a nonlinear mapping from the latent space to the data space, the algorithm generates a mixture of Gaussians, whose centres are constrained to lie on the nodes of a low dimensional space embedded in the high-dimensional one (see Figure 2.1). The gaussians model the uncertainty in the low dimensional representation of the data space [27], and are iteratively fitted to the data through the Expectation Maximization (EM) algorithm. This method basically extends the principle of the most popular Self-Organizing Map (SOM) [31] but with a more explicit formulation of the mathematical properties of the data. In fact, the GTM has these advantages compared to the SOM:

- it explicitly formulates a density model over the data.
- it uses neighbourhood parameters to ensure topographic ordering
- it uses a cost function that quantifies how well the map is trained.
- it uses a sound optimization procedure (EM algorithm).

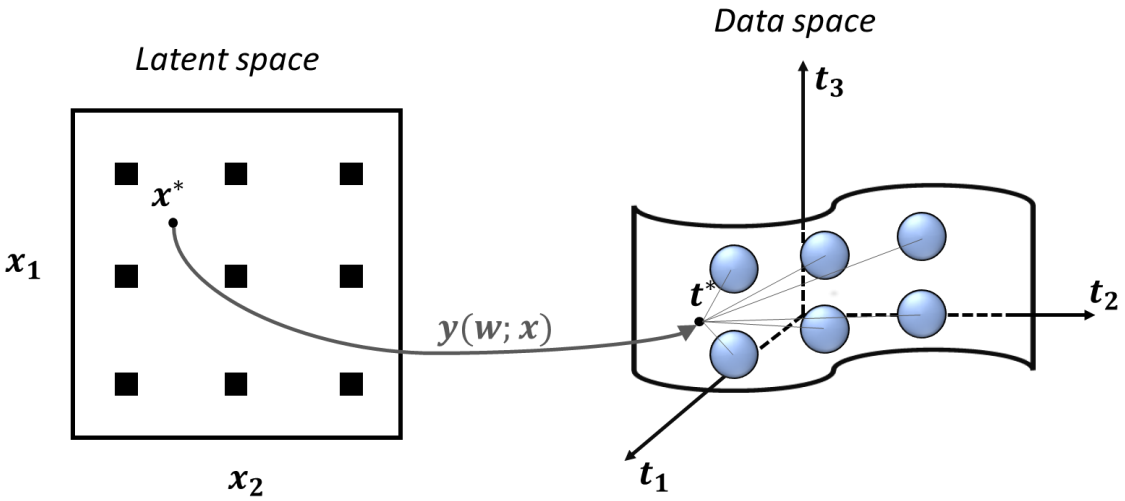

Figure 2.1: GTM mapping and manifold: each node located at a regular grid in the latent space is mapped to a corresponding point $y(x;W)$ in the data space and forms the centre of a corresponding Gaussian distribution. In Figure 2.1 the correspondences between a data point in the manifold embedded in the data space and the mean of the posterior distribution in the latent space x* is also shown.

More formally, let $\mathbf{X} = \{\mathbf{x}_1; \mathbf{x}_2; \cdots \mathbf{x}_K\} \in \Re^L$ be a regular grid of nodes in the latent space, and $\boldsymbol{T} = \{\boldsymbol{t}_1; \boldsymbol{t}_2; \cdots \boldsymbol{t}_N\} \in \Re^D$ be the training data set in the data space. The GTM algorithm performs a parameterized nonlinear mapping $y(\boldsymbol{x};\boldsymbol{W})$ from $\mathbf{X}$ to $\boldsymbol{T}$ consisting of a linear combination of Radial Basis Functions (RBF) Φ (but in principle other non-linear functions could be defined) with weighting coefficients $\boldsymbol{W}$:

$$y(\boldsymbol{x},\boldsymbol{W}) = \boldsymbol{W} \cdot \Phi(\boldsymbol{x}) \qquad (2\text{-}1)$$

Another internal hyperparameter is the width σ of the RBFs, which allows to control two important properties of the manifold iteratively fitted to the data: smoothness and flexibility. As said before, the uncertainty related to the assumption

that the actual data points lie on an embedded low dimensional manifold is modelled through symmetric Gaussian probability density functions, whose centres correspond to the mapped latent points into the T-space. This Gaussian noise assumption added to the model gives rise to a mixture of Gaussians where β is the inverse of the noise variance:

$$p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{W},\beta) = \left(\frac{\beta}{2\pi}\right)^{-\frac{D}{2}} \mathrm{e}^{-\frac{\beta}{2}\{\|y(\boldsymbol{x},\boldsymbol{W})-\boldsymbol{t}\|^2\}} \tag{2-2}$$

The probability distribution over the T-space is integrated over the $\boldsymbol{X}$ distribution under the assumption that the latent variable distribution is modelled as a superposition of delta functions associated to the K nodes of the regular grid in the latent space. Through the steps described in [27], the final formulation for the distribution function over the T-space can be obtained:

$$p(\boldsymbol{t}|\boldsymbol{W},\beta) = \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{t}|\boldsymbol{x}_k,\boldsymbol{W},\beta) \tag{2-3}$$

The GTM, therefore, defines a parametric probability density model that is fitted to the data by maximizing the log likelihood function by means of the Expectation Maximization (EM) algorithm [32], [33]:

$$\max_{\boldsymbol{W},\beta} \mathcal{L} = \sum_{n=1}^{N} ln\left(\frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{t}_n|\boldsymbol{x}_k,\boldsymbol{W},\beta)\right) \tag{2-4}$$

The adaptive hyperparameters of the model ($\boldsymbol{W}$ and β) are updated during such an iterative learning to compute the final values $\boldsymbol{W}^*, \beta^*$. At the end of the iterative procedure, the GTM defines a probability distribution over the data space conditioned on the latent variable, whereas the visualization of the resulting mapping is possible only in the low dimensional latent space. The corresponding posterior distribution over the latent space can be computed through the Bayes' theorem referring to the prior distribution of the latent variable $p(\boldsymbol{x})$:

$$p(\boldsymbol{x}_k|\boldsymbol{t}_n) = \frac{p(\boldsymbol{t}_n|\boldsymbol{x}_k,\boldsymbol{W}^*,\beta^*)\cdot p(\boldsymbol{x}_k)}{\sum_{p=1}^{K} p(\boldsymbol{t}_n|\boldsymbol{x}_p,\boldsymbol{W},\beta)\cdot p(\boldsymbol{x}_p)} \tag{2-5}$$

In order to visualize the whole data space on the map, the posterior probability distribution over the latent space is usually summarized through a statistical measure such as the mean or the mode:

$$\boldsymbol{x}_n^{mean} = \sum_{k=1}^{K} \boldsymbol{x}_k * p(\boldsymbol{x}_k|\boldsymbol{t}_n) \tag{2-6}$$

$$\boldsymbol{x}_n^{mode} = \sum_{k=1}^{K} argmax\, p(\boldsymbol{x}_k|\boldsymbol{t}_n) \tag{2-7}$$

2.3 Artificial Neural Networks

Artificial Neural Networks (ANN) try to mimic the way biological nervous systems, such as the brain, process information. Their structure is typically a series of interconnected layers: the most common model is the fully connected (FC) neural network which, as visible in Figure 2.2, is made of an input layer, an output layer and an arbitrary number of hidden layers, each containing processing elements (neurons).

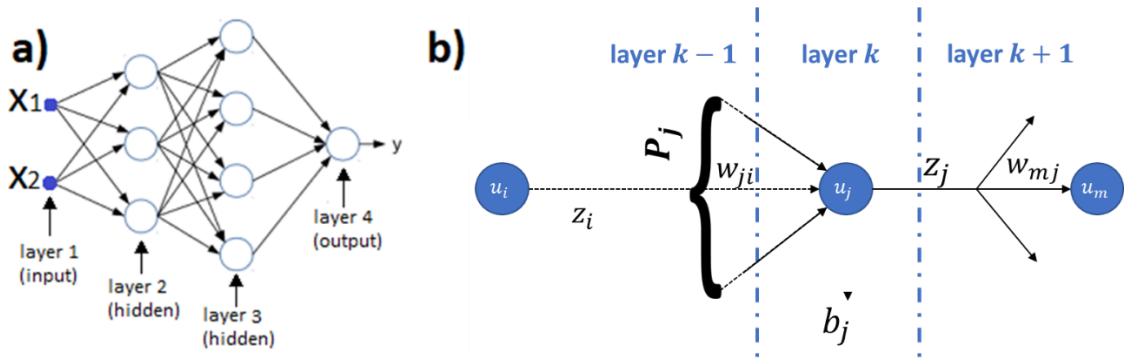

Figure 2.2: a) An example of feed-forward neural network; b) the neuron u_j is the basic processing unit of an artificial neural network

An artificial neural network takes a vector of inputs $\boldsymbol{x}$ and processes it through a number of K layers of neurons to produce the output y (Figure 2.2.a). A neuron of the layer k is the basic processing unit (u_j), visible in Figure 2.2b, which weighs its input data and produces normalized output through its activation function. An artificial neuron u_j takes a set of inputs $\boldsymbol{Z_j}$ from the neurons of the previous layer to which it is connected. These connections are weighted with the matrix $\boldsymbol{W_j}$, which is used to compute a weighted sum of the inputs $\boldsymbol{P_j}$. To this sum, a bias term b_j is added, and the neuron then merely computes its output $z_j = f(\boldsymbol{W_j^T Z_j} + b_j)$, where $f(\cdot)$ is the neuron activation function. The activation function is a non-linear function, which is fixed in the architecture among a set of commonly used ones, such as the logistic function, the sigmoid, the hyperbolic tangent (*tanh*) and the rectified linear unit (ReLU). In a FC-NN, all neurons of a layer are connected to all the neurons in the following one, as in Figure 2.2a. To calculate how well the network is approximating the array of target outputs t, an error function E is used; an example of this function is the Mean Squared Error (MSE), which is dependent on the weights and the biases of the network:

$$MSE = \frac{1}{n}\sum_{k}(NN(\boldsymbol{x}_n) - t_n)^2 \qquad (2\text{-}8)$$

where $NN(\boldsymbol{x}_k)$ is the neural network output for the input $\boldsymbol{x}_n$, and t_n is the target output for the n_{th} input. The cost function goes to 0 when the neural network can correctly map the input output relationship for all the data in the dataset. The output of the network can be a single numerical value or a vector (as in one-hot encoding for classification problems); in the latter case, vectorial distances can be

used to compute the MSE. The MSE or other kinds of loss functions can be minimized using the gradient descent algorithm. The number of misclassified samples cannot be used directly to evaluate the weights because the relation of a specific weight with the misclassified pattern is not straightforward (if compared to the single perceptron): changing the weight might not produce an immediate change of the number of correctly classified data, but can reduce the distance from the desired output array $\boldsymbol{t}$ and, consequently, ease the improvement of the classification by small steps; this algorithm exploits the gradient descent and is called backpropagation.

Backpropagation is the learning algorithm for NNs and exploits the continuity and the differentiability of the output. The influence of a weight change in a hidden layer can be calculated knowing the error of the following layers, so that, using as a reference Figure 2.2a, the output is computed starting from the inputs at the left to the right (forward phase), while the back-propagation computes and updates the weights from the right to the left (backward phase). In practice, considering a FC neural network, as in Figure 2.2b, u_j is the neuron of layer k, with its inputs $\boldsymbol{Z_j}$ and its output z_j, w_{ji} denotes the weight of the connection from unit u_i at layer $k-1$ to unit u_j at layer k. The weighted inputs of u_j are $\boldsymbol{P_j} = \boldsymbol{W Z_j}$. A gradient descent approach is then used to update the weights: starting from a random $\boldsymbol{W}$, when E is greater than 0, the algorithm changes the weights to reduce E

$$w_{ji(new)} \leftarrow w_{ji(old)} - \eta \frac{\partial E}{\partial w_{ji}}, i = 0,1, \dots d \tag{2-9}$$

where d is the total number of neurons at the layer $k-1$. This update reduces, in a first order approximation, the loss function E. Considering an improvement on the single sample n, the term $\frac{\partial E}{\partial w_{ji}}$ can be rewritten as:

$$\frac{\partial E}{\partial P_k} = \frac{\partial E}{\partial \boldsymbol{P_j}} \frac{\partial \boldsymbol{P_j}}{\partial w_{ji}} \tag{2-10}$$

Where $\boldsymbol{P_j}$ depends on the outputs $\boldsymbol{Z_j}$ from the I neurons of the previous layer:

$$\boldsymbol{P_j} = \sum_I w_{jI}\, z_I \tag{2-11}$$

so that $\frac{\partial \boldsymbol{P_j}}{\partial w_{ji}} = z_i$. The term $\frac{\partial E}{\partial \boldsymbol{P_j}}$, instead, is the influence of the input of u_j to the error. This is called δ_j and can be computed depending on the position of the neuron in the network. There are two cases:

- u_j is an output neuron: the input $\boldsymbol{P_j}$ comes from a hidden layer, while the output z_j is equal to the output of the network or to one of its components (for simplicity, here

$z_j = y$). Remembering that z_j is calculated with a continuous update function, here we use the logistic function $f = \left(1 + e^{-\boldsymbol{P_j}}\right)^{-1}$, we can calculate the MSE

$$E_n = \frac{1}{2}[y - t_n]^2 = \frac{1}{2}\left[z_j - t_n\right]^2 = \frac{1}{2}\left[f(\boldsymbol{P_j}) - t_n\right]^2 \quad (2\text{-}12)$$

So that

$$\delta_j = \frac{\partial E_n}{\partial \boldsymbol{P_j}} = \left[f(\boldsymbol{P_j}) - t_k\right]\left[1 - f(\boldsymbol{P_j})\right]f(\boldsymbol{P_j}) = \left[z_j - t_n\right]\left[1 - z_j\right]z_j \quad (2\text{-}13)$$

- u_j is a hidden neuron: the input $\boldsymbol{P_j}$ comes from an input or from another hidden layer and the output z_j feeds the M neurons of the following layer. Then δ_j is

$$\delta_j = \frac{\partial E_k}{\partial \boldsymbol{P_j}} = \sum_M \frac{\partial E_k}{\partial p_M}\frac{\partial p_M}{\partial \boldsymbol{P_j}} \quad (2\text{-}14)$$

In this expression $\frac{\partial E_k}{\partial p_M}$ is the influence of the input to the M units of the next layer with respect to the total error. This is equal to the sum of the δ_m of the following layer. The term $\frac{\partial p_M}{\partial \boldsymbol{P_j}}$, instead, can be split into the contribute $\frac{\partial p_M}{\partial z_i}$ and $\frac{\partial z_i}{\partial \boldsymbol{P_j}}$, where the first one depends on the input weights of the following layer and the second one is the same derivative of the activation function.

$$\frac{\partial E_k}{\partial p_M} = \sum_M \delta_M \quad (2\text{-}15)$$

$$\frac{\partial p_M}{\partial p_j} = \frac{\partial p_M}{\partial z_j}\frac{\partial z_j}{\partial p_j} = w_{mj}\frac{\partial f(\boldsymbol{P_j})}{\partial p_j} = w_{mj}z_j(1 - z_j) \quad (2\text{-}16)$$

This method allows you to update all the weights, starting from the deeper layer to the first one. There are some differences depending on the network structure, on the error function, on the activation function, and on the gradient descent algorithm chosen. These parameters can be edited to achieve different results.

2.4 Convolutional Neural Networks

Convolutional neural networks are inspired by the visual system structure [34] and revolutionized the field of computer vision, due to their capability of exploiting local patterns in the image data. The deep architecture of a CNN normally consists of a cascade of blocks of different layers which performs a filtering of an input image to extract significant features from it. The features are produced by a cascade of filtering blocks, interconnected through nonlinear activation functions as in the FC neural networks (typically a Rectified Linear Unit), and a FC layer combines them to produce the output of the network.

The processing blocks can be of different kinds [35]:

- Convolutional Layer:

  This layer exploits the convolution operation to extract feature maps from the input image. During the forward step, the kernel or filter is convolved with the image-producing the image representation of that receptive region. The convolution produces an activation map. In the backward step, the filter weights are tuned with backpropagation [36]. The kernel has several design parameters which must be set, such as the kernel or filter size, stride, padding and dilation. The stride is the sliding size of the filter. The image can be padded at the sides to adapt the output size. The number of filters determines the number of activation maps produced. In a traditional convolutional layer with an input of size L x L x D and D_{out} filters with a spatial size of F with stride S and amount of padding P, then the output size will be L_{out}, computed as:

$$L_{out} = \frac{L - F + 2P}{S} + 1 \tag{2-17}$$

  The convolutional layer can have extra parameters, such as the stride of the convolution, the dilation, which is the creation of empty spaces among the kernel elements. The dilation parameter is used to keep the kernels compact while focusing on larger areas of the input image. Other variations of the convolutions are described in [29].

- Batch Normalization Layer:

  This layer has been introduced in a recent paper [37] and it is added to normalize the input of the previous layer before passing them to the following one. This layer has two learnable parameters (β, γ), and two estimated or learned parameters, the mean moving average and the standard deviation moving average $(\mu_{mov}, \sigma_{mov})$. The two moving averages are used to normalize the data according to the equation:

$$x_j^{l'} = \frac{x_j^{l-1} - \mu_{mov,j}^l}{\sigma_{mov,j}^l} \tag{2-18}$$

  So that the data has unitary standard deviation and 0 mean. $x_j^{l'}$ is the output of this intermediate step. Then, the data is rescaled and shifted using β and γ:

$$x_j^l = \gamma x_j^{l'} + \beta^l \tag{2-19}$$

  Where x_j^l is the output of the batch normalization layer. This layer is placed before the activation layer in the original publication [37], but other ones propose to move it after the nonlinear activation.

- Activation Layer:

  In this layer, a nonlinear activation function g is applied to the input feature maps. As in the FC neural networks, this layer follows a processing layer such as the convolutional layer.

- Subsampling/Pooling Layer:

This layer is used to down sample the input feature maps. The down sampling is applied independently for each input map, preserving the third dimension. In general, this layer contains a kernel which computes a max or an average function sliding over the input maps. For example, if a 2x2 kernel is employed, the output dimensions will be half of the input ones.

- Fully Connected Layer

  This is the fully connected layer which computes the score of each class from the extracted features from a convolutional layer in the preceding steps. The final layer feature maps are represented as vectors with scalar values which are passed to the fully connected layers

- Classification/Regression Layer

  This layer is the output layer of the CNN. If the task is a classification, then a SoftMax operation is performed to the input from the fully connected network and then a criterion is applied to classify the image. If the layer performs a regression, generally a simple linear layer produces the final output.

This structure is very general, and several variations can be applied depending on the actual task. This architecture is very flexible and can be employed in a plethora of different problems and contexts. The CNN is able to process local spatial properties among the features and is considered the state of the art in computer vision tasks. It is usually employed in a supervised learning framework, despite self-supervised applications are being developed [38], [39]. An example of CNN structure is visible in Figure 2.3.

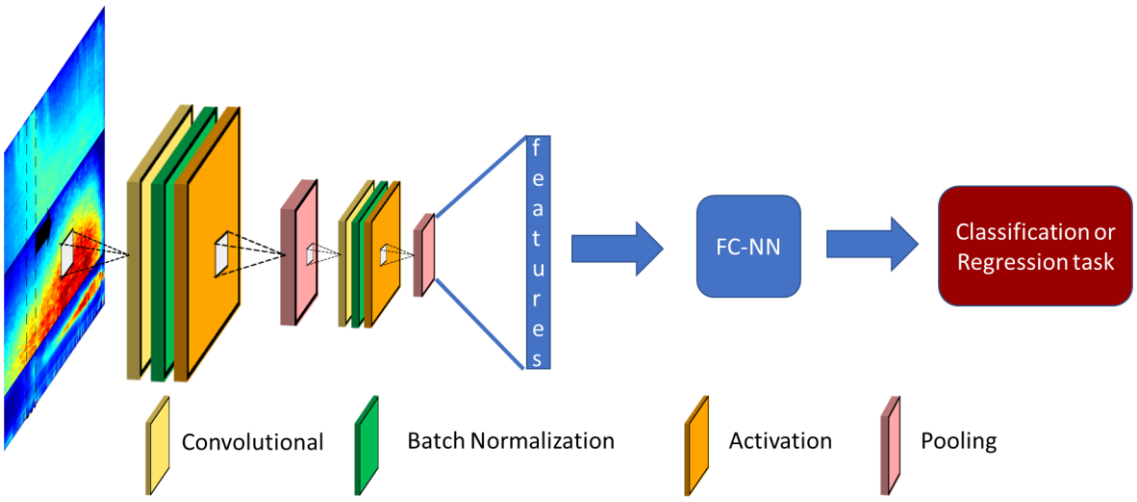

Figure 2.3: The overall architecture of the Convolutional Neural Network. Includes an input layer, multiple convolutions, activations (ReLU) and pooling layers and one fully connected layer [40]

2.5 Physics Informed Neural Networks

In the past years, the research community increased the understanding of multiscale physics problems in a variety of applications, by implementing more efficient solvers for Partial Differential Equations (PDEs). The solvers can be based

on finite differences, finite elements, spectral and meshless methods. Nevertheless, the use of classical analytical and computational tools may not be sufficient to model and predict the evolution of large nonlinear multiscale systems or may require prohibitive costs. Finally, the solution of real-life physical problems with missing or noisy boundary conditions through traditional approaches is very complex. In this context, machine learning can be implemented, due to its capability of analyzing large design spaces, exploiting correlations in multi-dimensional domains and being suitable to ill-posed problems. Deep learning approaches may be used for feature extraction purposes when a large amount of observation is available. Unfortunately, despite their flexibility and the high performances obtained in several fields [41], most data-driven approaches currently are not interpretable and may require huge amount of data to handle multidimensional domains. Since the solution of multiscale physics problem may require the use of very complex models, the target data generation may be a bottleneck in the traditional approach.

Moreover, in the absence of a regularizing term, even though purely data-driven models may fit observations very well, their predictions may be physically inconsistent or implausible. Therefore, fundamental physical laws and domain knowledge could be integrated in the machine and deep learning models, providing a physics-based regularizing term additional to the observational one [42]. Physics-informed learning is the process by which prior knowledge derived from our observational, empirical, physical or mathematical understanding of the world is exploited to improve the performance of a learning algorithm. Physics-informed neural networks (PINNs) are the most investigated example of this learning paradigm: a class of deep learning algorithms which integrate data and abstract mathematical operators, including PDEs with or without missing physics. Constraining the neural network with prior knowledge or a physics bias has several advantages:

- Since the model will also learn the underlying physics principles linked to the observations, the model results are robust in the presence of noise and the extrapolation power of the model will be higher.
- The model output will be more interpretable, and the regularizing term could provide a metric for measuring the performances of the model in the inference.
- The presence of a regularizing term reduces the number of necessary data for the model training, as visible in Figure 2.4. The better knowledge of the problem physics allows to build the model using fewer data, up to the possibility of training PINNs without experimental data [43].

Physics-informed neural networks [30] can integrate the information from both the measurements and partial differential equations (PDEs) by embedding the PDEs into the loss function of a neural network using automatic differentiation [44]. The PDEs could be integer-order PDEs , integral-differential equations [45], fractional PDEs [46] or stochastic PDEs [47], [48].

The generic equation:

$$N_{x,t}(u,\eta) = 0, x \in \Omega, t \in [0,\mathrm{T}] \tag{2-20}$$

with a suitable initial condition u_0 for t = 0

$$u(x,0) = u_0(x), x \in \Omega \tag{2-21}$$

and Dirichlet boundary conditions $u_b(x_b,t)$ at the boundary of the spatial domain

$$u(x,t) = u_b(x_b,t), x_b \in \partial\Omega, \mathrm{t} \in [0,\mathrm{T}] \tag{2-22}$$

can be solved in with numerical methods such as finite difference methods (FDMs) finite element methods (FEMs) or finite volume methods (FVMs).

In the case of the PINN, considering a set of points $T(x,t;\hat{u})$, where $\hat{u}$ is the PDE solution computed with a traditional solver, a traditional neural network can be trained by regressing $\hat{u}$ in the training set, as in Figure 2.4a. Instead, a PINN can be trained by computing the gradients of $\hat{u}$ with automatic differentiation, and then by constraining the model solution to solve the PDE respecting the boundary and initial conditions, as in Figure 2.4b. If the problem is completely defined and well-posed, then the solution is unique, and the PINN can also be trained without additional training data. Of course, a mixed approach can also be used where measurement data are available and the PDE is fully known.

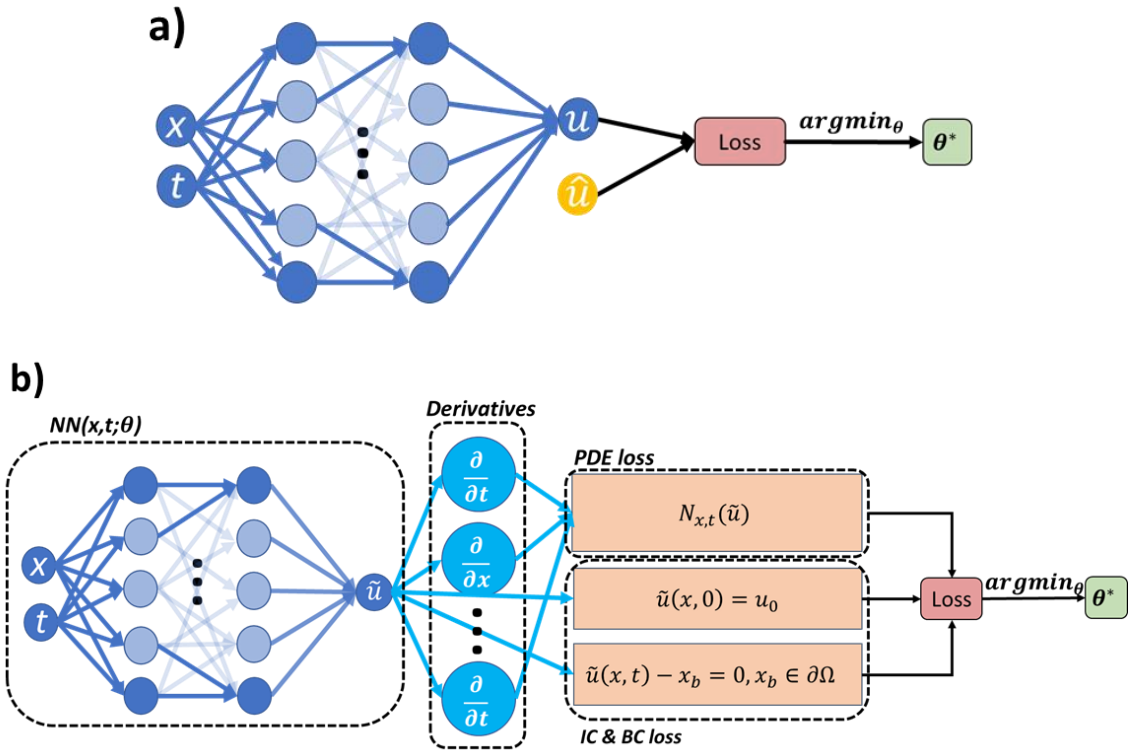

Figure 2.4: a) Traditional Neural Network solving the PDE problem; b) PINN solving the problem without using target data (only physics loss). The network output can be automatically differentiated with respect to the inputs using automatic differentiation, enabling the satisfaction of the PDE. The other components of the loss are the boundary and initial conditions of the PDE. Adapted from [45]

A general algorithm for training the PINN is the following [42], [45]:

Construct a neural network (NN) $u(x,t;\theta)$ where θ is the set of trainable parameters (weights and biases). The network output is differentiable with respect to the inputs of the network, so that it is possible to compute the PDE loss. Moreover, if boundary points are provided, also the boundary condition loss can be computed by enforcing the values of the solution and its derivatives at the boundary. Finally, if real measurement or solver data $\{x_i, t_i, \hat{u}_i\}$ for u and the residual points $\{x_j, t_j\}$ are available, it is also possible to compute the data loss. The total loss L is obtained by summing the weighted data, boundary and PDE losses. Train the NN to find the best parameters θ^* by minimizing the loss L with backpropagation until a stop condition is matched.

The weights of the losses are not necessarily kept equal for the entire training, and they are crucial for the PINN convergence; it was shown that the order of magnitude of the PDE loss components can be different from the one of their gradients or derivatives and how their scale can slow down or prevent convergence [49], [50]. Nevertheless, the problem of PINN convergence is still under research, as several techniques can be used to improve the learning of the PDE. For instance, other works [51]–[53] exploit the fact that any point within the domain can be sampled since there is no need to have a target value to compute the loss. Actually,

several approaches adopt resampling strategies to refine the quality of the solution in specific points of the domain until convergence [52], [53]. PINNs have several advantages with respect to the other numerical PDE solvers: they can be used to regress nonlinear PDE operators[43], [54]–[56]; they are mesh-free and can handle irregular domains; they are able to exploit the parallel computing capabilities of Graphical Processing Units (GPUs) [42], [43], [54]. It is a very recent learning paradigm which is applicable to several classes of contexts and problems where additional information on the problem is known.

Part 2: Disruption prediction at JET

Chapter 3
Disruption prediction and state of the art

3.1 Disruption prediction methods in Tokamaks

In tokamaks, the forces produced during an unmitigated disruption can seriously threaten the machine. A disruption is characterized by a sudden collapse of the plasma energy, both in terms of thermal energy (plasma temperature) and plasma current. The thermal quench initiates a major plasma disruption event, but as the plasma resistivity increases as its temperature decreases, the plasma cooling causes the current quench. The loss of the plasma current may also cause an abrupt drift of the plasma column, making it collide with the conductive vessel and compromising the integrity of the first wall and of the coils. For this reason, since the 90s, the ITER Physics Basis [57] review paper discussed the idea of radiating the plasma energy before the disruption, mitigating the consequences of a disruption. In a following issue [58], the areas of disruption prediction and mitigation were established. Unfortunately, the prediction of disruption is a complicated task, since the thermal quench is not affecting global plasma parameters but its local properties, and, despite advances in the non-linear MHD simulation codes as JOREK [59], NIMROD [60] and M3D [61], the process is not yet fully understood. Several physical limits, which constrain the operational space of the magnetic confinement devices, have been found. The high density and the high beta limits, the safety factor at the edge and the Greenwald limit have been investigated. It is also well known that an asymmetric perturbation of the toroidally symmetric magnetic field can induce large MHD modes in low density and low q plasmas and cause a disruption. Moreover, a large increase of the impurity density can also lead firstly to a radiative collapse and ultimately to a thermal quench. Nevertheless, the first work that provided a systematic study of the disruption statistics is [62], where a statistical analysis of JET disruptions is made. The work aimed at explaining the trend of the disruption rate (i.e., the percentage of pulses that disrupted) over a long period of time. The work demonstrated that a better understanding of the device operation helped in reducing technical failures and errors. Moreover, it was shown that, in critical experimental campaigns, such as the first D-T campaign at JET, the more careful operation led to a lower disruption rate. A follow up publication [10] investigated the root cause of 2309 JET disruptions to provide the chain of events which determined the disruption, a list of possible technical issues and several limits related to JET operation. The analysis was made manually, with the help of measurements and logbook reports of the experiments. This work identified the main root causes behind disruptions and influenced the research on physics-based and data-driven disruption prediction methods.

3.1.1 Physics-based methods

One of the approaches to the prediction of disruptions is the development of physical models for disruption prediction and avoidance. The physical model should be able to identify the plasma equilibrium by fixing as boundary condition the measurements by the pick-up coils, then to predict the evolution of the plasma state in real-time, sufficient to adopt avoidance or mitigation actions. This task is very complicated and, despite a lot of progress in the MHD modelling of the plasma [63], a code able to include the full physics related to the problem and sufficiently fast to be adopted in real-time is not available yet [64]. For this reason, the "physics-based" approaches available either make use of simplified physics models or aim to the detection of specific "off-normal" events to enable their use in real-time. For instance, the Disruption Event Characterization and Forecasting (DECAF) algorithm, is a framework where different physics models, each targeted to detect a relevant physics event are combined to estimate the risk of a disruption [65], [66]. The code can detect rotating MHD modes, resistive wall and Edge Localized modes among other events. A real-time version of the code is being developed for the KSTAR tokamak [65], [67]. On the other hand, the detection of specific events related to the physics is also adopted for the avoidance and mitigation purposes. For instance, the parameters such as the radiation peaking, the locked mode signal or other indicators of off-normal MHD activity have been exploited to trigger safe stop procedures or mitigation at JET [68]. These approaches have the advantage of providing a response to the specific physical mechanism which is destabilizing the discharge and of being more interpretable. On the other hand, most of these models are either not sufficiently fast to be ported in real-time or they cannot address all the various kinds of events which can determine a disruption. For this reason, complex schemes, which handle the simultaneous detection of different events, are being developed for several devices [14], [69], [70].

3.1.2 Data-Driven Methods

The lack of a general physical model for the prediction of disruptions motivated the development of data-driven models from the experiments. Data-driven methods are usually characterized by the lack of a precise mathematical model that describes the plasma state, which is instead analysed in terms of the univariate or multivariate statistics of the experimental data. There is a plethora of data-driven approaches, each one with specific advantages or disadvantages. The general idea is that, for a set of input data $\boldsymbol{x}$, which should clearly describe the state of the system, the model should learn a function $f(\boldsymbol{x})$ which provides the level of disruption risk. In contrast with the physics-based methods, the $f(\boldsymbol{x})$ function is not known beforehand, but it can be learnt from labelled or unlabelled experimental data. The following paragraphs describe the main methods adopted in the disruption prediction task.

3.1.3 Statistical Methods

Statistics is pivotal for evaluating the soundness of the any data-driven methodology and provides insights on the data. For this reason, univariate or multivariate analyses are usually the first approach before developing a machine learning model [71], [72]. Moreover, in the literature, the statistical analysis has been applied to identify the start of the pre-disruptive phase [15], [71], [73]–[75] or to find cross-machine disruption indicators [76]. Moreover, an explorative analysis is necessary to select experiments from specific scenarios [77] compatible with the ones developed for the next generation tokamaks such as ITER or SPARC. In [78], the statistical method of Survival Analysis is integrated with Random Forest (RF) binary classification of disruptions for time-to-disruption studies. In [79], Discriminant Analysis (DA) is used as the main approach to disruption prediction on AUG. A log-linear discriminant function, constructed with five 0-D plasma parameters is derived for the edge cooling (EC) disruptions. DA [80] falls into the multivariate statistics approaches, and it makes use of a training data set $(\boldsymbol{x}_i, y_i)$, $i = 1, \cdots, I$, to determine a boundary between two response classes. The assumption of this method is that the input points of each class belong to a $n - D$ normal distribution. The work finds a linear boundary between the disruptive and the non-disruptive regions, describing the plasma state using the 0-D following plasma parameters: internal inductance l_i, Greenwald density ne_{GW}, Loop Voltage U_{loop}, poloidal beta β_{pol}, and the fraction of the radiated power over the total input power P_{frac}. The Logistic Regression method has been used in [75] to develop a disruption predictor for ASDEX Upgrade. Logistic regression is a nonlinear regression method which models the probability $p(y = 1|\,\boldsymbol{x})$ of a vector $\boldsymbol{x}$ of independent variables to be classified into one category of the dependent variable $y \in \{0, 1\}$. Depending on the probability p, the value is classified with a binary classification scheme. Moreover, in [81] an auto regressive exogenous inputs model (ARX) is built using P_{frac}, l_i, and β_{pol} from the non-disrupted pulses of ASDEX Upgrade. This model can be trained using only safe pulses, which is promising since ITER can only tolerate a few percent of disruptions.

3.1.4 Machine learning and Deep learning methods

The first techniques to be applied to the disruption prediction task were Support Vector Machines (SVM) and Neural Networks (NN). The SVM is a supervised machine learning algorithm, which projects the data in a higher dimensional space to find a linear boundary between two classes of vectors $\boldsymbol{x}$. The boundary is found by maximizing the distance between the separating hyper plane and the closest training data points $\boldsymbol{x}_i$ (named support vectors). The kernel function is a nonlinear function used to increase the domain dimensionality and is a fixed hyperparameter of the SVM. In [82] a first application of the SVM is proposed as a disruption predictor at JET for mitigation purposes using plasma diagnostic signals.

Then, the SVM has been also adopted in [83], [84], and posteriorly upgraded and installed in JET real-time network in 2012 (APODIS) [84]–[87]. Moreover, SVM is implemented at JT-60U for disruption mitigation [88], [89].

As previously cited, Fully Connected Neural Networks can approximate any continuous function of one or more variables [90], and if the network is trained with a binary target, its outputs can be interpreted as posterior probabilities. This is very useful for classification tasks, as it gives a certainty measure on the classification performance [28]. The earliest applications of FC-NN based disruption predictors are in [91]–[95] and new models based on FC-NN are still developed [96], [97]. The input of these networks are plasma diagnostic signals from disruptive and/or non-disruptive pulses. In [92], [93], [98], a disruption prediction tool based on a FC-NN neural network, ideally suitable for real-time application, has been implemented and tested over the flat-top phase of JET and AUG discharges. In [93], [99], the AUG predictor has been adaptively trained whenever it triggers a missed alarm.

More recently, a FC-NN model has been developed also for the J-TEXT tokamak, focusing on the prediction of density limit disruptions [96]. Moreover, in [100] a hybrid two-stage neural network architecture is proposed: the first stage is a custom network, which uses time series diagnostics as inputs to predict plasma density, and the second stage is a FC-NN to predict the probability of density limit disruptions. Finally, in [97] the FC-NN is employed to determine whether an equilibrium data point from NSTX is below or above the no-wall stability limit.

In the literature, also decision trees have been widely adopted for developing interpretable disruption predictors. Among these, the Random Forest algorithm has been employed to classify disruptions at DIII-D and Alcator C-Mod [71], [73], [78]. The forests are grown by developing parallel sets of predictors, thus collecting a large number of independent and identically distributed, de-correlated decision trees [101]. The trees are usually fully grown, and the final prediction is aggregated, using majority voting, from a large number of trees.

Among the unsupervised machine learning algorithms there are Manifold Learning Algorithms such as Self Organizing Maps (SOMs) [9], [81], [102], and Generative Topographic Maps (GTMs) [15], [72], [103], which have been used for both disruption prediction and classification with very successful results. This class of algorithms aims to group data exploiting only the properties of the data itself. Therefore, they can be used for an exploratory analysis of the data space, or for feature extraction and selection. In this context, they were exploited for mapping and visualizing the high dimensional space, typical of the disruptive process, in a lower dimensional space. Moreover, the projected discharge trajectory on the map allows to clearly visualize the principal differences among the chain of events which cause the disruptions [15], [72].

In [103] the k-Nearest Neighbours (k-NN) is used as a comparative method for evaluating the Generative Topographic Mapping results on disruption classification. In fact, the k-NN error tends to the Bayes error when the size of the training set

tends to infinity, giving a statistical lower bound on the error achievable for a given classification problem and associated data.

Finally, recently, deep learning methods have been adopted to develop disruption predictors. DL consists in the use of deep neural networks to address complex problems. CNN and Recurrent Neural Networks (RNNs) are extremely popular architectures of this field. As previously cited, CNNs are mostly applied to image analysis. Their main advantage is the ability to achieve 'spatial invariance', which implies that they can learn to recognize and extract image features independently on their position in the image. Moreover, thy can act as feature extractors, simplifying the cumbersome feature engineering process, which is necessary in traditional machine learning pipelines. RNNs instead are used to process time series and dynamical data by introducing recurrent connections which cause the network output to be determined from the current inputs and the previous ones. Long-Short Term Memory (LSTM) neural network is the most common RNN architecture. CNNs have been employed for developing disruption predictors either alone [104], [105] or in combination with RNNs [100], [106]–[109].

3.1.5 Adaptive learning

Data-driven approaches are affected by the problem of ageing, which is the deterioration of their performance when the device changes or undergoes profound modifications (e.g., the change of the JET wall). To deal with this issue, researchers developed data-driven models from scratch with data from the machine itself [110] or incorporate a small quantity of new data into the previously trained model [106]. This implies the generation of predictors that require at least one disruptive discharge and one non-disruptive discharge to be able to distinguish between the two plasma behaviours [111]. However, these models created with so much scarcity of information need to be updated in an adaptive way to incorporate relevant information as new discharges are produced [82], [92], [93], [99]. The adaptive procedure includes not only gaining knowledge (i.e. to learn) about disruptive and non-disruptive features of the parameter space but also accomplishing a de-learning process (i.e. to remove training samples that are no longer valid due to their statistical irrelevance from the evolution of the experimental program of the device).

The need of at least one disruptive discharge to create a first predictor from scratch can be avoided by using anomaly detection methods [75], [112]. These methods are based on recognizing off-normal behaviours in plasma quantities that are related to potential incoming disruptions.

Chapter 4

JET Database

4.1 Introduction

A tokamak discharge is generally divided into 3 phases, depending on the plasma current trend: ramp-up, which is the phase when the plasma current rises to the chosen value; flat-top, which is the phase when the control system keeps the current stable; ramp-down is the final phase of the discharge, when the current decays in a controlled way, until the experiment terminates. A general correctly terminated (named here as "safe") pulse develops by passing through all the 3 phases, with only minor problems which do not compromise the overall plasma stability. A disrupted pulse, instead, is characterized by the destabilization of the plasma equilibrium by a series of events of variable complexity; the disruption may occur in any of the phases, but the plasma during the flat-top yields generally much more energy, due to the higher current and temperature values. In Figure 4.1, a schematic is reported, which describes the development of the plasma current of a generic safe (blue) and disrupted (red) discharge.

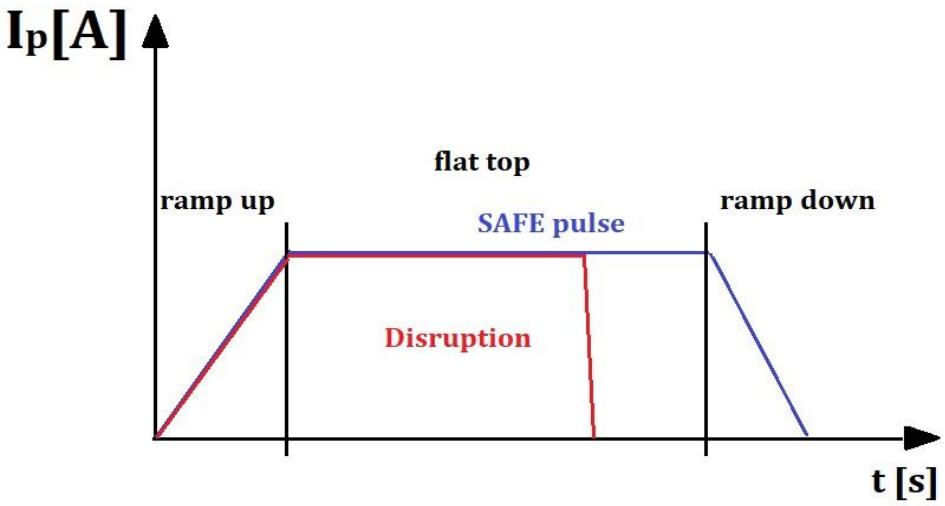

Figure 4.1: Sketch representing the 3 phases of a pulse: ramp-up, flat-top and ramp-down

The construction of a database is a mandatory step for the development of whatever data-driven technique. The analysis of nuclear fusion data is particularly challenging due to the high dimensionality of the operational space of the devices, to the large amount of available data and to the expertise required for the selection of the set of relevant diagnostics in a specific application. The existing database maintained by the University of Cagliari, contains several disrupted and regularly terminated pulses from the first JET ITER-Like Wall (ILW) campaigns. In particular, the database was adopted for disruption prediction and classification [113] studies and contains non-intentional and unmitigated disruptions. During the three years of PhD research program, the database has been updated with the

discharges coming from the JET experimental campaigns from 2016 to 2020, focusing on the C36 (2016) and the C38 (2019-2020) campaigns. Since these campaigns were carried out using higher input power, the number of unmitigated disruptions was very low. For this reason, also mitigated disruptions have been considered, adopting as disruption time the time of mitigation. The overall database for this work contains a total of 198 disrupted and 219 regularly terminated discharges having a flat-top plasma current higher than 1.5 MA, and a flat-top length greater than 200 ms. The analysis of the pulses refers to the flat-top phase. In particular, for each selected discharge, the flat-top starting time has been assumed as the first time instant where the plasma is in X-point configuration. For the disrupted pulses, the flat-top ending time (t_{end}) is assumed as the time of the valve activation for those terminated by Massive Gas Injection (MGI), and as the disruption time (t_D), corresponding to the drop of the core temperature and the start of the plasma current spike, for the unmitigated ones. Disruptions caused by Vertical Displacement Events have been excluded at all from the data set. These criteria are widely employed in disruption prediction and avoidance studies to select relevant experiments [15], [106]. The considered database covers a wide set of experimental conditions, starting from the earlier campaigns with the ILW until the recent experiments where high power experiments were carried out. It has been grouped in 3 datasets, as detailed in Table 4.1, following the different experimental campaigns.

Table 4.1: Database composition

Dataset	Years	Campaigns	Composition	
			Disruptions	Regular pulses
I	2011÷2013	C28-C30	132	115
II	2016	C36	29	41
III	2019÷2020	C38	37	63

The database was analysed with univariate and multivariate statistics. Table 4.2 reports the diagnostics used for the preliminary analyses, the development and the testing of the disruption predictors presented in the thesis. Among these diagnostics, the first 5 were used to carry out the main exploratory analyses of the database, while the remaining ones were used to extract relevant features for the disruption prediction task. The differences among the three datasets in terms of explored regions in the operational space are shown in Figure 4.2 where, from top-left to bottom-right, the distributions of the plasma current (I_p), the toroidal field (B_T), the normalized beta (β_N), the total input power, the line integrated density, and the edge safety factor (q_{95}) are reported for the regularly terminated discharges. It can be noted that the Datasets I and II share the same parameter ranges even if, for some parameters (such as I_p, B_T or q_{95}), their distributions slightly differ. Instead, the Dataset III, which is related to experiments aiming to study the baseline scenario

suitable for sustained high D-T fusion power, is characterized by higher currents, density and input power, also exceeding the range of the other two datasets.

Table 4.2 Diagnostic signals, acronyms and units.

Plasma signal	Acronym	Diagnostics	Dimension
Plasma Current	I_p	MAGN	0-D
Poloidal Beta	β_p	BetaLi	0-D
Normalized Beta	β_N	BetaLi	0-D
Line Integrated density	$n_{e,int}$	FIR interferometer	0-D
Safety factor at the edge	q_{95}	EFIT	0-D
Toroidal magnetic field	B_T	MAGN	0-D
Electron Temperature	T_e	HRTS	1-D
Electron Density	n_e	HRTS	1-D
Radiated Power	P_{rad}	Bolometer	1-D
Total Radiated Power	$P_{rad-TOT}$	Bolometer	0-D
Total Input Power	P_{TOT}	BetaLi	0-D
Internal Inductance	l_i	BetaLi	0-D
Normalized locked mode	LM_{norm}	LMS	0-D
Mirnov Coil signal	M_{signal}	Mirnov Coil	0-D
Spectrogram	S	Mirnov Coil	1-D

Previous studies [15], [72], [73], [106] demonstrated the importance of the plasma profiles for the early detection of unstable plasma states. For this reason, the profile quantities such as electron temperature, density and radiation have been exploited for the development of disruption predictors. Figure 4.3 shows the area covered by the HRTS diagnostic installed at JET with respect to the poloidal section, while Figure 4.4 shows the lines of sight of the horizontal and vertical bolometer cameras.

Moreover, other quantities were taken into consideration such as the internal inductance (l_i), which provides information on the current profile, and the mode lock signal normalized by the plasma current (ML_{norm}), which is commonly used to predict the onset of a mode-locking before the final destabilization of the plasma [15], [76], [114]. Moreover, to anticipate the alarms provided by the mode lock signal, a signal coming from the Mirnov coil diagnostics was analyzed with the aim of detecting the presence of slowing down MHD modes.

As detailed in the next Part, for the analysis of the 1-D plasma profiles, two main approaches were used. One involves the synthesis of 0-D indicators, which could represent the spatial distribution of the profile quantity, the other is the creation of 2-D images by processing the profile data and extracting the profile signal with a sliding window, producing a spatiotemporal representation of the profile evolution.

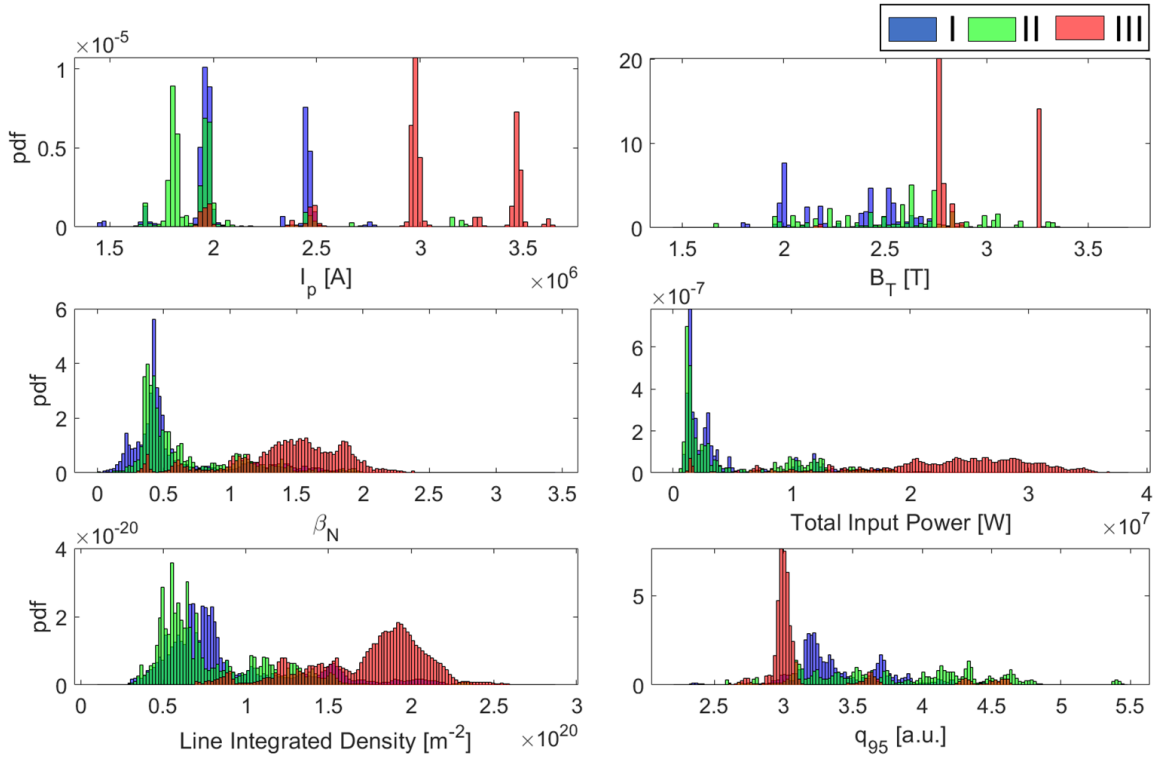

Figure 4.2: Distributions of the main parameters of the regularly terminated discharges in the Dataset I (blue), Dataset II (green) and Dataset III (red) for (from top left to bottom right): plasma current, toroidal field, normalized beta, total input power, line integrated density and edge safety factor.

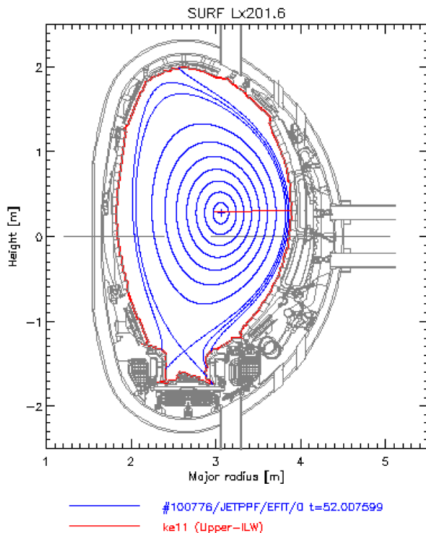

Figure 4.3: Area covered by the HRTS at JET (in red), and the reconstruction of the magnetic surfaces with EFIT in shot #100776

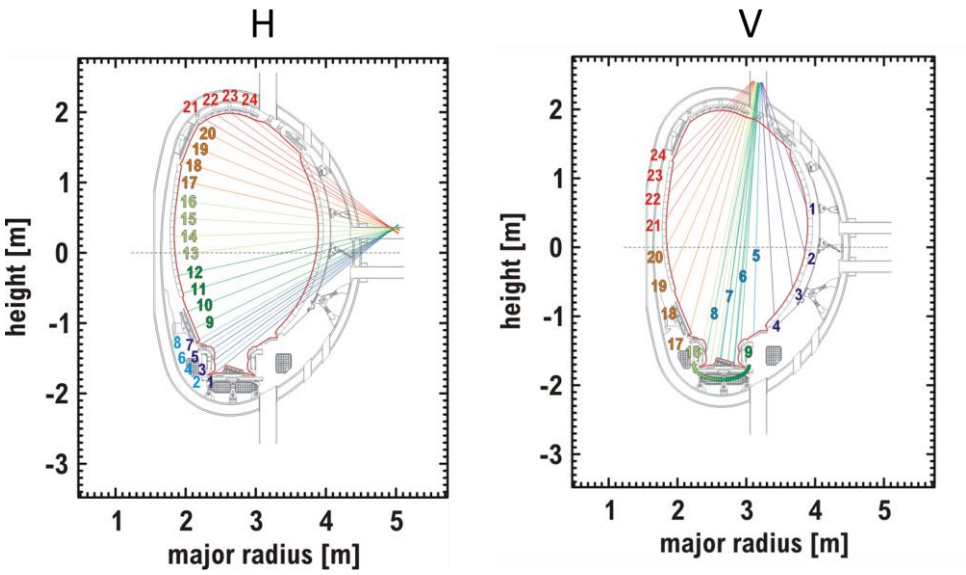

Figure 4.4: View of the JET bolometer camera system: horizontal camera (left side) and vertical camera (right side)

4.2 Processing the plasma profiles

The profiles represent the time evolution of fundamental plasma quantities such as the electron temperature, the electron density, and the radiation. In [72] these 1-D profiles have been processed to synthesize physics-based indicators called "peaking factors" (PFs). The peaking factors demonstrated very useful to discriminate between a non-disruptive plasma state and a disruptive one. The peaking factors for both density and temperature have been considered as features defined with a "core versus all" metric; they are computed as the ratio between the mean value of the considered radial profile (temperature, radiation, density) around the magnetic axis and the mean value of the measurements over the entire radius. The radial interval to define the "core" with respect to the magnetic axis is the 25% of the radial coordinate (the minor radius for poloidal mid-plane measurements) in the case of electron temperature (Te_{pf}) and density (Ne_{pf}) peaking factors. Regarding the radiated power, in [15] the same authors computed the peaking factors using the main-vessel bolometric camera with a horizontal view of the plasma cross-section (Bolo H). Two different peaking factors have been derived splitting the information carried out by the global poloidal radiation distribution. Firstly, the core is defined as 4 channels of the Bolometer, from 13th to 16th, and the divertor as 8 channels, from 1st to 8th. Then, the two peaking factors, the Rad_{pf-CVA} and the $Rad_{pf-XDIV}$ have been computed: the first one is the ratio between the average radiation in the core, and the average radiation in the entire plasma excluding the divertor area. The $Rad_{pf-XDIV}$ is instead computed as the ratio between the average radiation in the divertor and the average radiation in the entire plasma excluding the area of core (20 channels).

Note that, the peaking factors are obtained by heuristically defining the core of the plasma with either a set of channels or as a percentage of the minor radius of the plasma, and the construction of the synthetic feature causes a loss of information. On the contrary, it is possible to analyze the full spatiotemporal information contained in the electron temperature and density, and radiated power profiles by converting the same set of 1-D diagnostics used in [15], [72], [115] in 2-D images.

In order to generate images from the profiles input data, the following steps have been implemented:

1. Firstly, data from the High-Resolution Thompson Scattering (HRTS) for the electron temperature (T_e) and density (N_e), and the horizontal lines of sight of the bolometer for the radiated power (P_{rad}), is resampled at the same JET real-time network sampling time of 2 ms. Since the objective is developing real-time compatible disruption predictors, the resampling is made by only exploiting current and past values information. This operation allows the system to work with signals at the same time scale, as these diagnostics have different sampling times, which vary from 10^{-4} to 10^{-2} s. Note that, the raw measures from these diagnostics are used.
2. Once the resampling has been carried out, the 1-D profile data is processed to create a set of input images, as detailed in the following and sketched in Figure 4.5a-c:

– a pre-processing is applied to each diagnostic to remove outliers. In particular, for the HRTS diagnostic, the pre-processing consists in the comparison of the measurement with the diagnostic estimated error [116]. As some shots, both for HRTS and Bolometer profiles, presented corrupted measures, a pre-processing procedure has been developed, based on the correlation between the measure of each line of sight and those of their neighbours. The corrupted measures are replaced by the interpolated values between the closest ones. From an inspection of the training dataset, the outer 9 lines of sight (from major radius greater than 3.78 m) of the HRTS are discarded as, at least on the selected dataset, they usually provide unreliable data. For the Bolometer data, no estimation of the measurement error was available, so negative power values have been substituted with null values, whereas unreliable positive ones are saturated to a fixed threshold empirically fixed to 1MW/m^2;
– for the HRTS diagnostics, the lines of sight are ordered from the inner (R=2.96m) to the outer one (R=3.78), where R is the major radius. For the bolometer diagnostic, the lines of sight are ordered as labelled in Figure 4.4. Then a spatiotemporal matrix is built, whose elements assume the value of the measure in the corresponding line of sight and the corresponding time sample. The obtained images are shown in Figure 4.5b;
– the three images are vertically stacked, and their ranges are normalized with respect to the signal ranges in the training set. After retrieving the maximum and the minimum values from each diagnostic in the training set, its value x is normalized between [-1,1] by

$$x_{norm} = \frac{(2x - x_{max} - x_{min})}{x_{max} - x_{min}}$$

obtaining the final image in Figure 4.5c;

– the final image is segmented using an overlapping sliding window of 200 ms, as sketched by the dashed black line in Figure 4.5c. In this way, each segment corresponds to an image of $N_{CHANNELS}$x101 pixels, where $N_{CHANNELS}$ is the number of input channels.

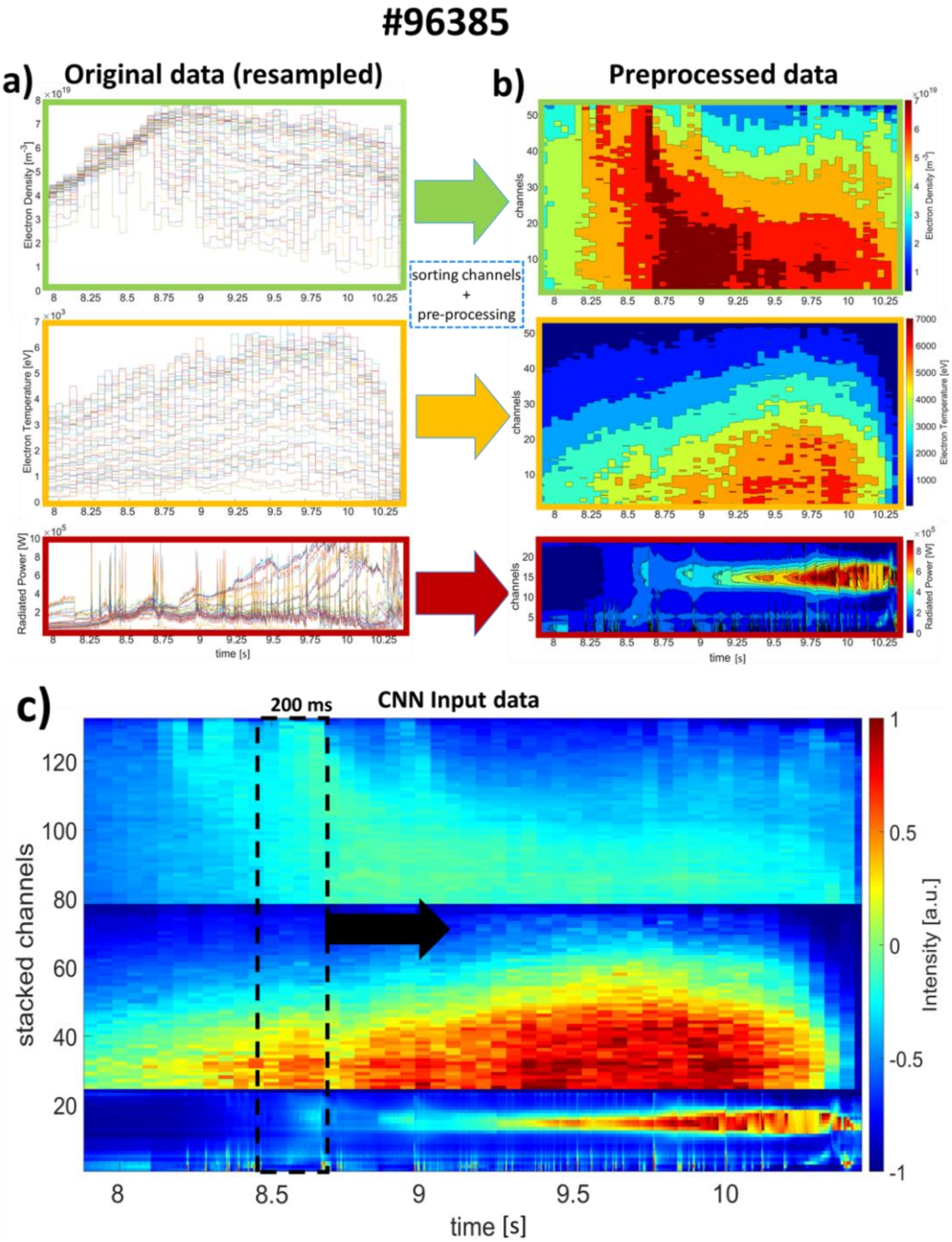

Figure 4.5: Sketch of the pre-processing steps applied to pulse #96385 to generate the input images: a) Original data from the HRTS and Bolometer diagnostics; b) Pre-processed data are converted into images; c) Input data, obtained by normalizing the data at point b) with the training set ranges and by vertically stacking the lines of sight. An overlapping window of 200 ms produces the segmented images to feed the prediction models.

4.3 Processing the Mirnov signals

Rotating magneto-hydro-dynamic (MHD) modes in tokamaks may grow in amplitude and slow down during plasma operation; eventually, they will stop rotating and remain in a certain position. This phenomenon is known as mode locking and leads very often to disruption or, in any case, to a degradation of confinement. Tangential pick up or Mirnov coils can be used to measure the mode propagation velocity in real-time, and specific signals, such as the mode-lock signal, have been synthesized to measure the amplitude of the radial component of the magnetic field. The locked mode amplitude is calculated from the saddle coils and it is normalised by the plasma current (I_p) [117]. When the mode is locking the its amplitude can be evaluated through the measure of the voltage induced in the saddle coils. In JET and in other tokamaks, the mode locking is a very late but extremely reliable disruption precursor, and several data-driven approaches aiming to disruption mitigation have adopted it as input feature [76], [82], [85]. Moreover, the frequency of the magnetic amplitude oscillations detected by the Mirnov coils has been studied by means of Singular Value Decomposition (SVD) [118], [119], Wavelets [120], [121], Kalman Filters [122]. In this thesis, to test the feasibility of the use of 1-D MHD information in disruption prediction, the Mirnov coil signals have been processed as in [123], [124] to create images for a Convolutional Neural Network disruption prediction model.

Figure 4.6 reports the list of the JET's main magnetic diagnostics, where H301-H307 is an array of high resolution Mirnov coils with a sampling frequency of 2 MHz. Among these coils, the signals from H302 and H305 were selected as possible input signals for MHD analysis.

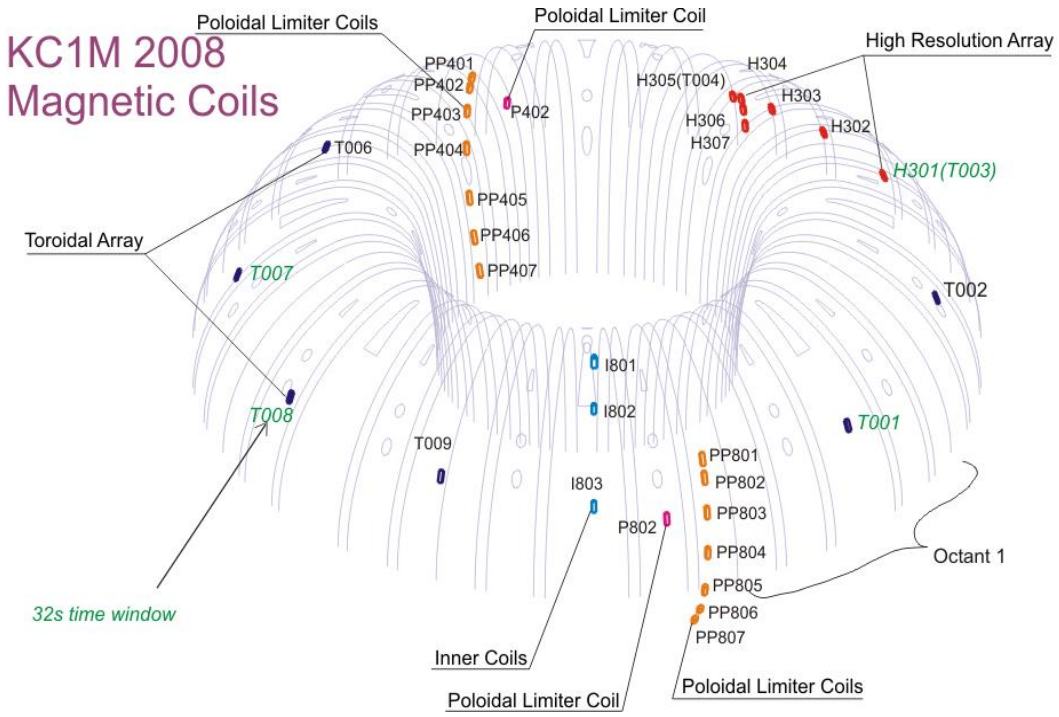

Figure 4.6: Overall view of the JET's main magnetic diagnostics for MHD analysis

The raw signal from the Mirnov coil is firstly down sampled to 125 kHz. This frequency is anyway much larger than the ones of interest for the application, which are usually below 40 kHz [123]. To build the spectrogram, a short-time Fourier Transform is then applied with a length and a step of 256 points, but with an overlap of 6 samples. The overlap allows to have a sampling time of 2ms to synchronize the spectrogram sampling time with the other diagnostics. The amplitudes of the power spectrum are converted in decibels (dB), the values are saturated between [-80,-10] dB, and the frequencies higher than 40 kHz are removed. Images of size 81x101 are obtained by extracting 200 ms windows from the spectrogram, as shown in Figure 4.7. It is possible to see that the locking of the mode is visible at around 59.5 s, before the disruption time.

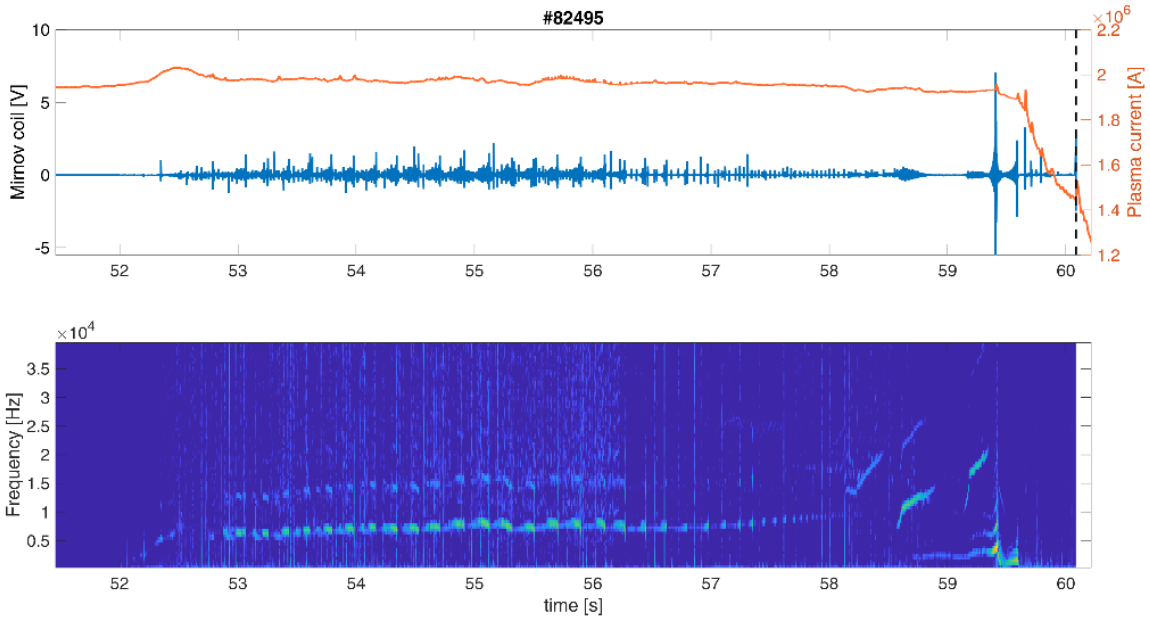

Figure 4.7: top) Plasma current (orange line) and Mirnov coil signal (blue line). The dashed black line highlights the disruption time; bottom) spectrogram of the Mirnov coil.

4.4 Processing the 0-D parameters

The 0-D parameters include both the synthetic ones obtained by processing the plasma profiles, and the l_i, P_{frac} and the LM_{norm} which are used for prediction purposes. The signals were resampled at the same time samples of the 1-D data. Moreover, P_{frac} data may include outliers, especially in the transients close to the switch-on or switch-off of the Neutral Beam Injector (NBI). For this reason, values higher than 9 are considered outliers and saturated to 9.

4.5 Creation of a disruption prediction dataset

The disruption prediction models developed in this thesis require the use of labelled data, which is necessary for the training step. To this purpose, for each disruption in the training set, it is mandatory to identify, as precisely as possible, the time of precursors, named here $t_{pre-disr}$, which determines the moment when the final chain of events destabilizes the plasma, i.e., the beginning of the precursors phase. The precursor time is the moment where the disruption predictor should trigger the alarm, to avoid a premature detection of the disruption. This task, far from being easy, has been solved in most of the literature, assuming the same value for all the disruptions on the base of statistics or heuristics, inevitably introducing contradictory information in the prediction models. As an alternative, a manual identification of the pre-disruptive times ($t_{pre-disr-MAN}$) can be done, as in [15], but this task is very time consuming and complicated and can also be uncertain due to the possible interplay of many different mechanisms. In this thesis, an algorithm to automatically estimate the pre-disruptive times ($t_{pre-disr-AUT}$) on unlabeled disruptive discharges has been developed, starting from a set of 0-D diagnostics including the peaking factors, the l_i, the P_{frac} and the LM_{norm}. The algorithm is described in detail in the following Chapter.

Chapter 5
Disruption Prediction with Generative Topographic Mapping

5.1 Introduction

As reported in the literature presented in Chapter 2, several machine learning and deep learning algorithms have been developed for predicting disruptions in tokamaks. On the other hand, the community is moving towards the use of these automatic systems for the identification of disruptive events in real-time control systems for avoidance and prevention actions, as well as mitigation. The use of machine learning disruption predictors for avoidance poses new challenges. First of all, the warning times of the data-driven models should be sufficient for implementing an avoidance strategy. Secondly, the output of the predictor should be interpretable or at least linked to a well-defined chain of physical phenomena, in order to enable a specific strategy for recovering the discharge, or to safely terminate it. Finally, there is a strong effort to build common data sets and metrics for the comparison of the different disruption prediction models. For addressing the first two issues, it was observed that the use of parameters connected to the plasma profiles could allow an earlier detection of an unstable plasma state. For this reason, synthetic parameters called “peaking factors”, or 2-D images from the 1-D plasma profiles, described in Chapter 1, were developed to inform the data-driven model on the electron temperature, electron density and radiated power profiles. In this context, the Generative Topographic Mapping (GTM) model is an unsupervised machine learning algorithm which allows to project a very high dimensional feature space in a lower dimensional latent space. The mapping is exploited for several purposes. Firstly, it allows to easily visualize patterns in the feature distributions and how disrupted plasma state differs from the regular one. For instance, the use of the GTM allowed to highlight the presence of a compact normal operational space of the JET experiments. Secondly, it allows to develop a disruption predictor by projecting a new discharge on the map and associating the disruption risk to the composition of the node of the map where the discharge is projected. The GTM, however, needs a labelling of the disruptive samples in order to be adopted as a predictor. Commonly, in the literature the pre-disruptive time was heuristically or statistically defined and fixed for all the disrupted discharge. In [15], [72], the manual detection of the pre-disruptive phase reduced the amount of ambiguous information provided to the data-driven model and allowed to identify a destabilization time which is different for every disruption. The pre-disruptive time identifies the beginning of the phase where the destabilization of the discharge is detectable, and a corrective action may be taken. However, the manual labelling of tokamak disruptions is a time-consuming task and prevents the application of

automatic procedures for the retraining of disruption prediction algorithms. For this reason, In the framework of the PhD, an algorithm has been implemented to automatically identify a pre-disruptive time for each disruption for the creation of disruption prediction databases. In order to validate the algorithm outcomes the performance of the GTM prediction model trained with the manual times has been compared with the one trained with the automatic times with very good results. Moreover, the algorithm allowed to extend the GTM training set to include more recent pulses.

5.2 Generative Topographic Mapping

In this thesis, the GTM mapping has been made from a latent 2-D space to the 6-D set of plasma parameters reported in Table 5.1. This fact enables the exploration of the low dimensionality domain, also called the latent space, to visualize properties of the input data space. For instance, in Figure 5.1a the Unified distance matrix (U-matrix) representation [125] of the JET operational space is reported. This matrix, a standard way of representing the latent space, visualizes the Euclidean distance among adjacent clusters of the map by using different shades of grey. In this way, the U-matrix allows one to display the similarity of data elements into one cluster with respect to the data into nearest ones. With this representation, it is possible to detect if there are macro-clusters of data and to judge if they are well separated or not.

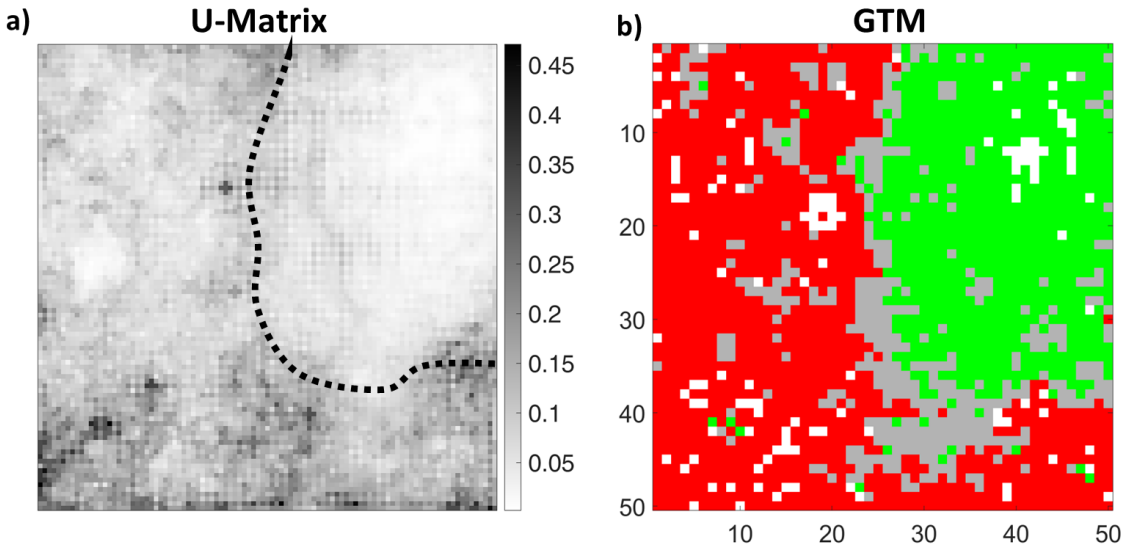

Figure 5.1: a) Example of U-Matrix of a GTM. b) Example the same GTM coloured using information from the labelling of the disruptive and non-disruptive samples

Moreover, the training instances $\{\boldsymbol{x}, y\}$ can be used to assign a different label, or color, to the map units depending on their composition. As previously cited, labelling of the training disrupted discharges is critical for the performance of the map. The estimation of the beginning of the precursors phase is difficult, and several works defined it with a time before the disruption time t_D, usually statistically or

empirically fixed [71], [82], [126]. Of course, this definition is not suited to represent the different physical timescales of the several mechanisms involved in the disruptive process. In Figure 5.1b, an example of a GTM is shown where each unit is colored using the manually detected pre-disruptive phases. The colors of the map units depend on the samples associate to it: green units are associated to samples labeled as non-disrupted (y=0), red units are associated to samples labeled as disrupted ($y = 1$), whereas grey units are associated to both disrupted and non-disrupted samples. The white units are empty. It is possible to see that the green region on the right in Figure 5.1b is a compact area representing the space of JET non-disruptive operation. This area is surrounded by a grey boundary where the transition between non-disruptive and disruptive behavior could be located and the rest is occupied by disruptive clusters, together with a few grey ones. In Figure 5.1a it is possible to confirm that the boundary obtained by looking at the distance between the clusters in the U-matrix matches very well with the green region of Figure 5.1b. Once the GTM model has been trained and successively colored, it can be used to track the dynamics of a new discharge, by projecting the temporal sequence of its samples on the map. In Figure 5.2a, the trajectory of the discharge is represented with a dashed line, the color of which darkens during the time evolution up to the tip of the arrow representing the end point. Usually, a disrupted discharge evolves in the green region until disruption precursors appear, moving the trajectory towards the red disruptive region. The disruptive likelihood of the discharge is obtained by evaluating the percentage of disrupted samples contained in the units visited by the trajectory, as the one represented in Figure 5.2b. In disruption prediction literature, the GTM has been implemented for disruption classification [15], [127], [128] and prediction [15], [72], [73] and it is implemented in the PETRA control system at JET [129].

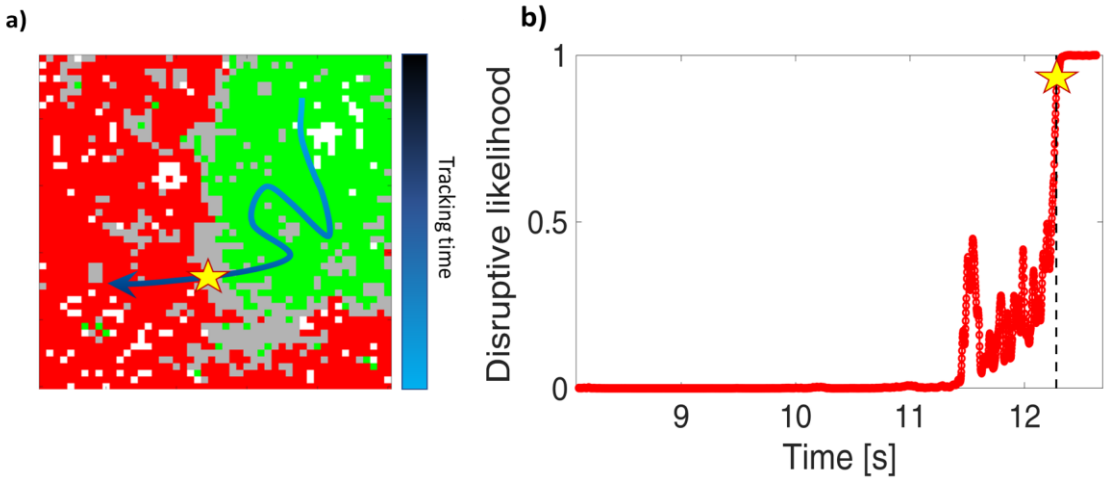

Figure 5.2.a) 2-D GTM map of a n-D space, coloured on the basis of the unit composition, and trajectory of a disrupted pulse. b) Disruptive likelihood computed depending on the composition of the node where the pulse is tracked. The star indicates the overcoming of an alarm threshold and the triggering of an alarm

In the disruption prediction literature, there is a general effort towards the standardization of the reference times considered for the prediction. For instance, in [130] the authors made a first step, developing a tool for the automatic definition of important times and parameters of the disruptions, such as the thermal quench and the current quench times, the time of disruption (t_D) and the Mode Lock time (t_{LM}), which is the time where the locked mode amplitude starts to rise [130]. Similarly, in [15], [72] the authors manually identify the so-called reference pre-disruptive time of a disruption, which provides a reference time to identify the start of the chain of events destabilizing the discharge and leading to the disruption. The introduction of consistent pre-disruptive times ($t_{pre-disr}$) is doubly beneficial. Firstly, the pre-disruptive times allow to identify the pre-disruptive phase, which is used to describe the disrupted operational space. Secondly, being the pre-disruption time strongly linked to the observation of physical properties of the plasma state, the predictor should provide a response, which can be explained by the knowledge of the physical phenomena. Moreover, as disruption research is moving from the process of disruption mitigation to the use of the prediction for avoidance purposes, the predictors should provide its response hundreds of milliseconds prior the disruption, and it should allow distinguishing among the different type of destabilizing chain of events. The key to a successful prediction model is therefore the capability, for each disrupted discharge in the training set, to discriminate among the non-disrupted and pre-disruptive phases following standard and coherent criteria, linked to the observed physical mechanisms. However, this classification requires a very time-consuming manual analysis [15], [72]; hence, adopting it to classify tens of thousands of shots would be highly impractical. Therefore, in this thesis, an algorithm for the automatic identification of early plasma instabilities, which may lead to the disruption, has been developed.

Starting from the training discharges, described in Chapter 1, the algorithm is trained by providing a set of input points extracted from the pre-disrupted part of the disruptions and from the regularly terminated pulses. The pre-disrupted part of disruptions is identified automatically with the algorithm developed in [115] and described later in this Chapter. Since the number of non-disrupted samples is much higher than the disrupted ones, the non-disrupted samples are subsampled by taking one point every 18 ms, while the disrupted samples are not down-sampled.

Table 5.1: Plasma parameters: parameter names, Acronyms, optimized weights.

Parameter name	Acronym	Weight
Peaking Factor of Temperature	Te_{pf}	1
Peaking Factor of Electron Density	Ne_{pf}	1
Peaking Factor of the Radiation (excluding the contribution of the X-point/divertor region)	Rad_{pf_CVA}	0.8
Peaking Factor of the Radiation (excluding the contribution of the core region)	Rad_{pf_XDIV}	0.5
Internal Inductance	l_i	1
Fraction of the Radiated Power	P_{frac}	0.7

After labelling the disruptive data, the GTM can be used in prediction by projecting a discharge on the latent space. A typical regular trajectory will stay within the green region of the map, while a disruptive trajectory will exit the green region and then terminate in the red area as in Figure 5.2. A disruption risk can then be assigned to each time sample of the experimental discharge by evaluating the percentage of disrupted samples contained in the cluster where is projected and its neighborhood [113] and alarming at the exit of the green region (yellow star in Figure 5.2b). Moreover, the use of synthetic parameters representative of the shape of the plasma profiles and of manually selected times for identifying the beginning of the unstable phase for each disruption allowed to better interpret the physical mechanism, which is causing the disruption, as discussed later in the examples. The identification of early pre-disruptive training times allows the possibility to develop avoidance schemes or to terminate the experiment with a soft stop rather than using the mitigation valve (DMV) which is instead a last resort system used to mitigate the disruption effects on the vessel.

The GTM model, developed for disruption prediction purpose in this thesis, has 2500 latent points, 400 radial basis functions with variance $\sigma = 0.8$. The hyperparameters have been assumed equal to the ones in [15], as well as the training set, which contains the same 89 disrupted shots and 70 regular terminations and was instead trained using the manually identified pre-disrupted phase.

In order to detect possible sudden events due to locked mode disruptions or other events with fast time scales, or false alarms due to transients, the multiple conditions alarm scheme reported in Figure 5.3 has been optimized starting from the implementation in [15] to trigger the alarm. In particular, the GTM model triggers an alarm when the trajectory stays in a disruptive or a mixed cluster containing a percentage of disruptive samples DS > 99.75% (likelihood>0.9975) for at least d seconds. The total assertion time d has been assumed to vary with the time evolution of the discharge with the exponential law: $d = 60 + 300 \cdot e^{-5(t-T_0)} ms$ where T_0 is the time when the plasma assumes the X-point configuration. Conversely, if the operating point lays in a mixed cluster with a percentage of disruptive samples

$50\% < DS < 99.75\%$, also the locked mode amplitude signal, normalized with respect to the plasma current, is considered to trigger the alarm. A threshold of 0.43 mT/MA is used to trigger the mode-lock branch. The alarm criteria parameters have been optimized by minimizing the total prediction error of the GTM on a validation set composed by all the training discharges. This set is different from the set used to train the GTM model, since the phase of disruptive discharges before $t_{pre-disr}$ is not used in the training, whereas regarding the non-disruptive discharges, the training set, because of the down-sampling, contains only approximately 10% of the total samples.

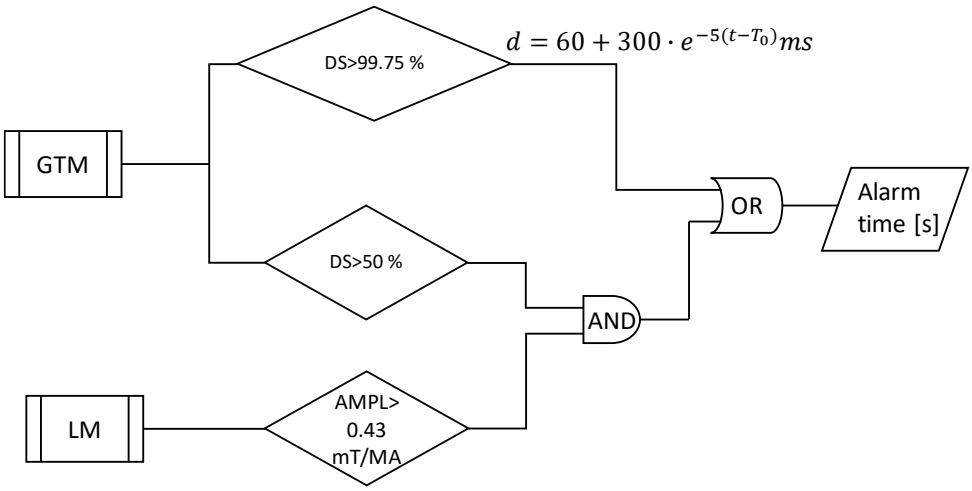

Figure 5.3: Multiple condition alarm scheme of the disruption predictor [15]. DS is the percentage of disrupted samples in the cluster where the discharge trajectory stays for at least d consecutive milliseconds (d is the assertion time). T_0 is the starting point of the flat-top.

5.3 Automatic detection of the pre-disruptive time

In this Chapter an algorithm for the automatic identification of the pre-disruptive phase of tokamak discharges is presented. The challenge of the understanding of very complex high dimensional spaces led researchers to the use of manifold learning techniques such as Self-Organizing Map [131], [132] and Generative Topographic Mapping [15], [72], [127]. Especially with the latter, the encouraging results led to the application of the method in a real-time framework [129]. On the other hand, these models need labelled data, which is necessary for the training step. To this purpose, for each disruption in the training set, it is mandatory to identify, as precisely as possible, the time of precursors, $t_{pre-disr}$, which determines the moment when the final chain of events destabilizes the plasma, i.e., the beginning of the precursors phase. In this thesis, using a set of features, synthetized to detect some of the main known disruption precursors in fusion experiments, an algorithm for the automatic identification of the precursors times has been developed and tested. The algorithm is based on the use of similarity measures between distributions, and it weights the contribution of each input

feature to construct a *Precursors Time Indicator* (*PTI*). The study of the *features* distribution in the regular discharges allows to optimize a coherent threshold value for the identification of the pre-disruptive times. Once the value of $t_{pre-disr,AUT}$ is determined, the discharge samples in the time window from $t_{pre-disr,AUT}$ to the disruption time t_D, are labelled as disrupted. Note that, for disruptions mitigated by massive gas injection (MGI), the time of the valve activation, t_{valve}, is considered in place of t_D. The samples of disrupted discharges, from the beginning of the flat-top to $t_{pre-disr,AUT}$, and all the flat-top samples in the regularly terminated discharges are labelled as non-disrupted.

5.3.1 Statistical analysis

A univariate statistical analysis has been firstly performed to evaluate the power of each selected feature in discriminating between disruptive and non-disruptive behaviour. This analysis has been performed on the first set of discharges of the data base (C28-C30 data set). Figure 5.4 reports the probability density functions (*pdf*) of the six parameters in Table 5.1 for the non-disruptive pulses (blue) versus the non-disrupted phase of the disruptive pulses (red). Here, the manual selected pre-disruptive times have been used to discriminate between the non-disrupted and pre-disruptive phases of the disrupted discharges. The results of the analysis, reported in Figure 5.4, refer to phases that can be considered in a non-disrupted condition. It can be observed that there is an overlap between the *pdf* of the parameters of non-disrupted (or safe) discharges and the non-disrupted phase of disrupted ones. Figure 5.5 reports the *pdf* of the parameters of the non-disruptive pulses (blue) versus the pre-disruptive phase of the disrupted pulses (red) for the same parameters in Figure 5.4. Looking at Figure 5.5, it can be seen that the parameters distribute differently during the unstable phase (i.e., after the pre-disruptive time $t_{pre-disr,MAN}$) with a wider range of parameter values. Moreover, the *pdfs* of the pre-disruptive phases of the disrupted discharges shift with respect to the stable phases. The orange arrows in Figure 5.5 highlight the shifts. Summarizing, during the non-disrupted phase of the disrupted discharges the distribution of the parameters is remarkably similar to the distribution of the regularly terminated discharges, while during the pre-disruptive phase, the values are distributed quite differently.

The main idea of the proposed algorithm is to introduce distance/similarity measures between these probability density functions when the pre-disruptive time varies, in order to automatically identify the moment when a disrupted discharge starts its pre-disruptive evolution. For instance, Figure 5.6 compares the distribution of the temperature peaking factor of the non-disrupted pulses (blue) in the database with the *pdf* of a window of 500 ms, centered at different time instants, of the disruptive discharge #81916. From a) to d) the time instant is getting closer and closer to the time of disruption, where in c) the time instant is the closest to the

manually selected pre-disrupted time ($t_{pre-disr,MAN}$) (about 50.07s) [15]. The time evolution in Figure 5.6 clearly shows that, approaching to the actual pre-disruptive time, the overlap of the two distributions reduced.

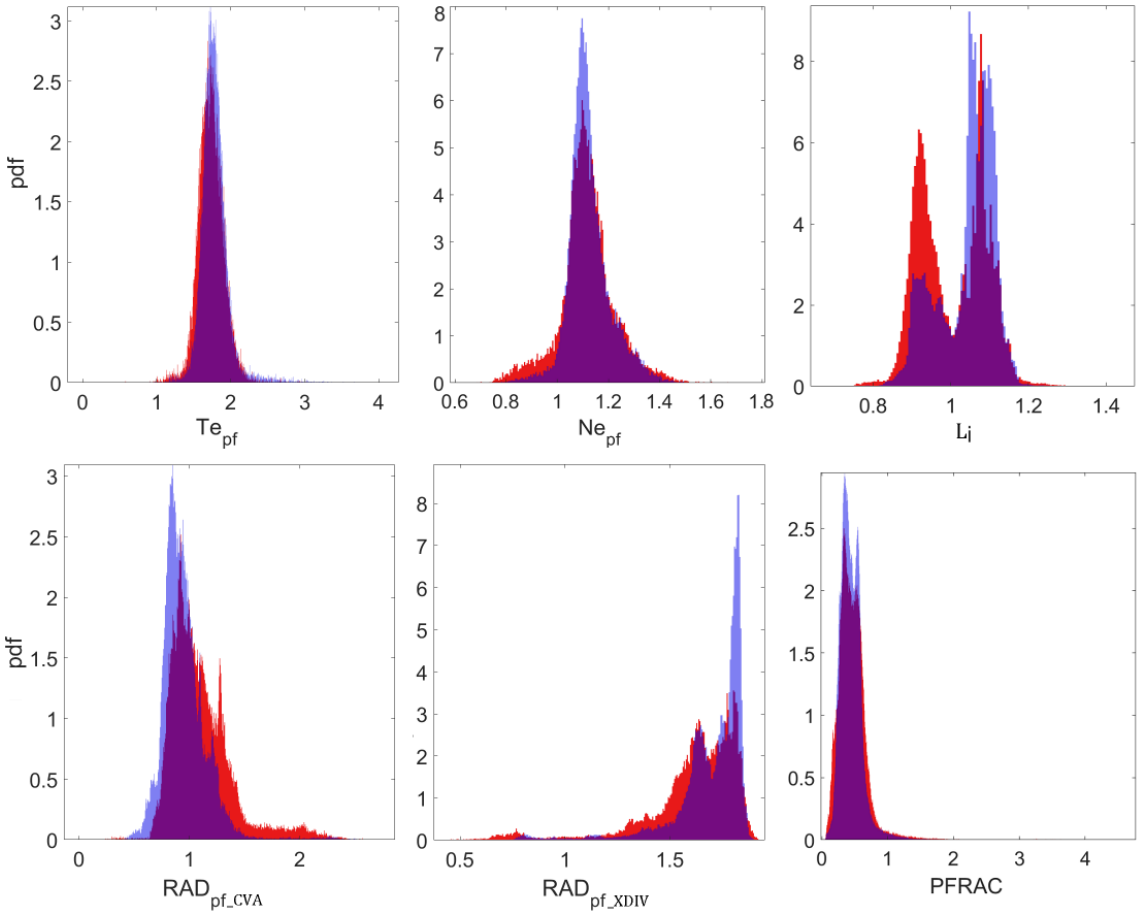

Figure 5.4: C28-C30 data set: Probability density functions of the parameters of the safe pulses (blue) versus the non-disrupted phase of the disrupted pulses (red) for (from top left to bottom right): electron temperature peaking factor, electron density peaking factor, internal inductance, radiation at the core peaking factor, radiation at the edge peaking factor, fraction of radiated power.

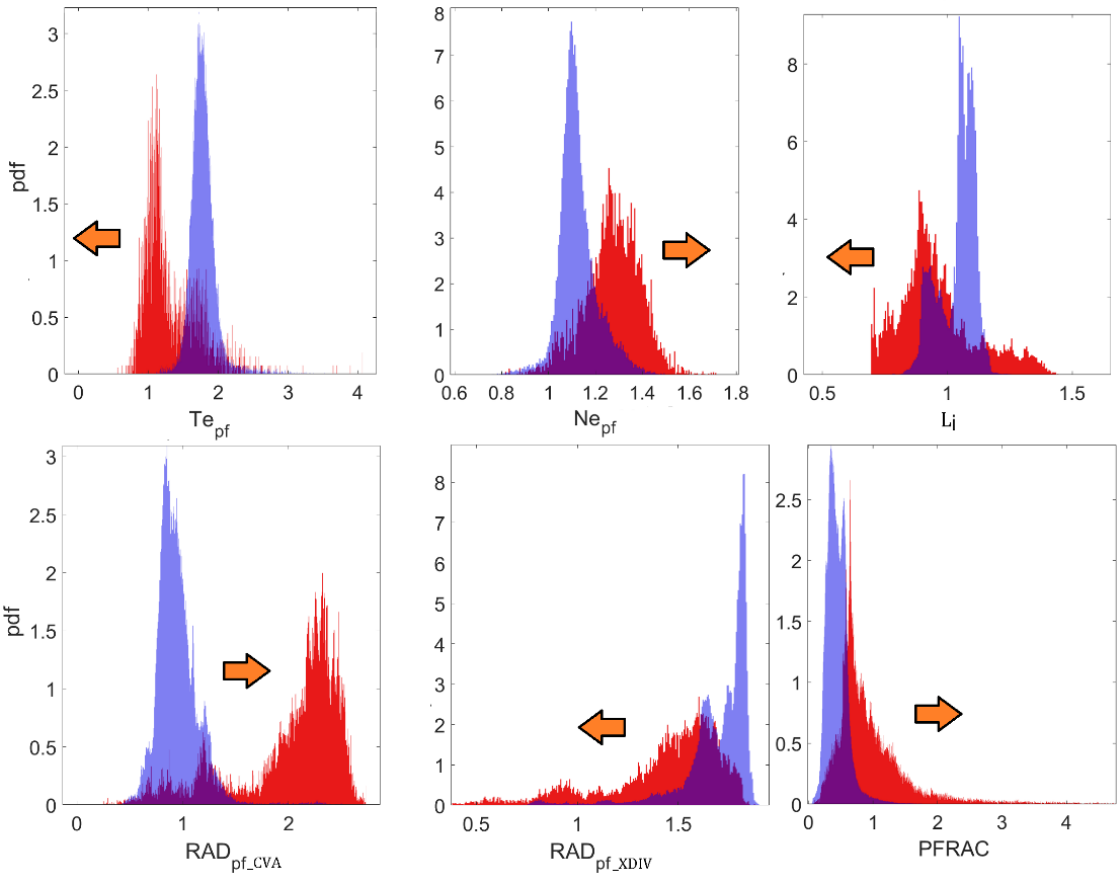

Figure 5.5. C28-C30 data set: Probability density functions of the parameters of the safe pulses (blue) versus the pre-disruptive phase of the disrupted pulses (red) for (from top left to bottom right): electron temperature peaking factor, electron density peaking factor, internal inductance, radiation at the core peaking factor, radiation at the edge peaking factor, fraction of radiated power. The shift of the distributions is marked with an orange arrow.

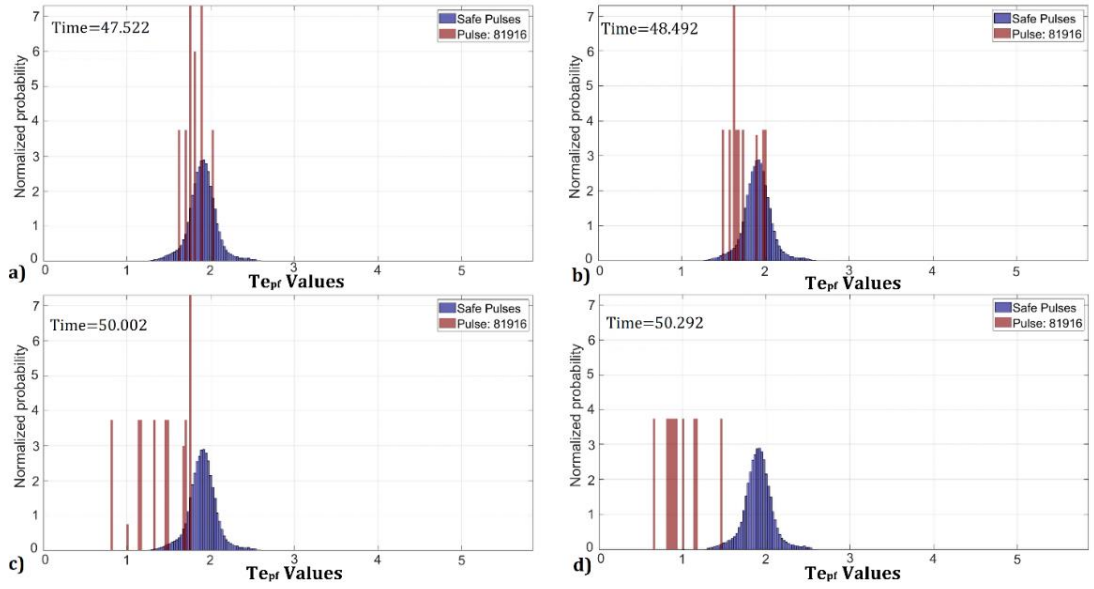

Figure 5.6: Probability density functions of the temperature peaking factor (Te_{pf}) of the safe pulses (blue) in the C28-C30 data set versus the pdf of a 500 ms window, centered at different time instants (indicated on each subplot), of the disruptive discharge #81916. From a) to d)

the time instant is getting closer and closer to the time of disruption, where in c) the time instant is the closest to the manually selected warning time ($T_{pre-disr,MAN}$) [13].

5.3.2 Features weights

Table 5.1 reports the nondimensional plasma parameters considered to develop the proposed algorithm for the automatic identification of the pre-disruptive times $t_{pre-disr}$: the last column of the table reports the weights assigned to the parameters as a result of an optimization of the algorithm, which will be detailed in the following. Literature [15], [72] proved that the selected features discriminate well between safe and disrupted pulses. Figure 5.7 and Figure 5.8 report these features for the regularly terminated discharge #83747 and the disrupted discharge #81916 respectively. The disrupted discharge is a high-Z impurity accumulation (or Radiation Peaking RPK) disruption [72], [133] with warning time $t_{pre-disr,MAN}$ manually set at 50.07 (highlighted with a red vertical line in Figure 5.8). It can be noted that, in the regularly terminated discharge, the variation range of the signals is generally smaller than in the disrupted one; while this remark may be valid in most of the cases, it is not necessarily true for all the discharges. Moreover, looking at Figure 5.8, it can be seen that the peaking factors characterize well the typical Radiation Peaking (RPK) evolution: the ne_{pf} shows an increase of the density in the plasma core correlated with a temperature drop. Moreover, the peaking factor of radiation at the core rises, as well as the overall fraction of radiated power, while the internal inductance starts to decrease. This chain of events starts from the penetration of high-Z atoms in the core of the plasma that produces a destabilization of the MHD equilibrium in the plasma itself. The proposed algorithm weighs the variations in these signals' distributions to identify the start of the chain of events leading to disruption. This is done by comparing the distribution of each signal in the regularly terminated discharges in different time instants with the distribution of the same parameter of the single disrupted discharge, as detailed in the following section.

The data base includes datasets I from Table 4.1. This first set has been used to perform the statistical analysis and to assess and optimize our algorithm. In order to test the generalization capability of the algorithm, a second data set (C36 data set) has been used, which includes 29 disrupted and 41 regularly terminated pulses within the more recent (2016) high performance campaigns both in baseline and hybrid scenarios. In this case, the suitability of the algorithm to correctly identify the pre-disruptive phase of the disrupted discharges has been evaluated in terms of the composition of the GTM that maps the more recent plasma operational space, i.e., in terms of its capability to discriminate between disrupted and non-disrupted regions.

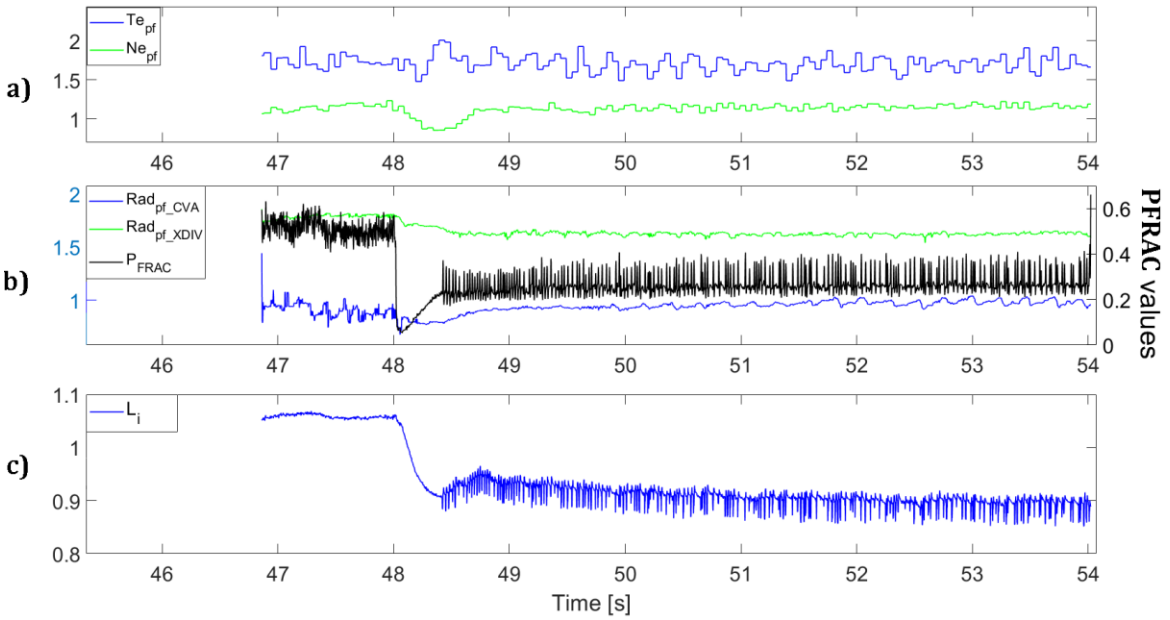

Figure 5.7: The input features for the algorithm for the JET regularly terminated discharge #83747: a) the peaking factors of the temperature (Te_{pf}, in blue) and density (Ne_{pf}, in green); b) the radiation peaking factors with the metric "Core Vs All" (Rad_{pf-CVA}, in blue), which excludes the divertor, and with metric "Edge Vs All" ($Rad_{pf-XDIV}$, in green), which excludes the core, and the Power Fraction (P_{FRAC}, in black); c) the internal inductance (l_i, in green).

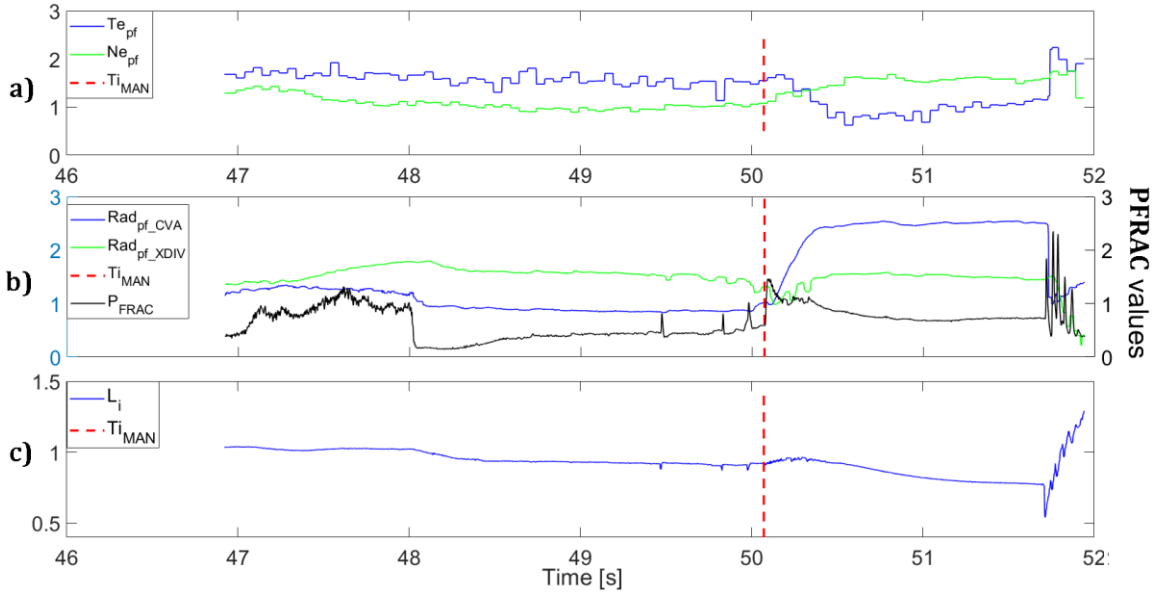

Figure 5.8: The input features for the algorithm for the JET disrupted discharge #81916: a) the peaking factors of the temperature (Te_{pf}, in blue) and density (Ne_{pf}, in green); b) the radiation peaking factors with the metric "Core Vs All" (Rad_{pf-CVA}, in blue), which excludes the divertor, and with metric "Edge Vs All" ($Rad_{pf-XDIV}$, in green), which excludes the core, and the Power Fraction (P_{FRAC}, in black); c) the internal inductance (l_i, in green). A vertical red line marks the manually detected warning time $T_{pre-disr,MAN}$.

5.3.3 The algorithm

As discussed in [15], [72], the selection of the pre-disruptive time $t_{pre-disr,MAN}$ required a tedious and time-consuming analysis of several events and parameters, additional to the ones used as inputs for the proposed algorithm, and not necessarily available in real time. A *Predisruptive Time Indicator* (*PTI*) has been built here that can be used to automatically detect the pre-disruption time in the disrupted discharges.

As previously mentioned, the algorithm is based on the comparison of the distributions of the selected plasma parameters in the regularly terminated and in the disrupted discharges. In particular, it is assumed that, before the onset of the chain of events leading to disruption (before the actual pre-disruption time $t_{pre-disr}$), the distributions of the parameters in the non-disrupted phase of a disruption be close to those of the safe discharges, whereas they become more and more dissimilar while approaching the disruption time. Hence, for each plasma feature in Table 5.1, the distribution of the safe pulses (*SAFE_distr*) has been considered as the reference distribution. Then, for each discharge and for each time instant *t*, the algorithm scans every parameter from the beginning to the end of the flat-top, identifying two different distributions:

- *LEFTpart_distr*: the distribution before *t*
- *RIGHTpart_distr*: the distribution after *t*

and computes the distance/similarity between these two distributions to the *SAFE_distr*.

Note that, for time instants at the beginning (at the end) of the flat-top, a very small number of samples is available for the *LEFTpart_distr* (*RIGHTpart_distr*). This creates a border effect at the beginning (at the end), which has been partly compensated by padding the first 125 ms of the initial and final part of each signal. The padding has been done by simply replicating the respective part of the signal, so that at the beginning of the flat-top and at its end, the distributions could be represented by more values.

In order to evaluate the distance/similarity, several metrics have been considered [24], based both on the computation of misclassification probability, such as Bhattacharya, Hellinger, Kullback-Leigler Divergence and Matusita and on the computation of the distribution similarities, such as those belonging to the inner product family. Among all the tested metrics, in this thesis, the final choice was the Cosine similarity metric, which basically implements the normalized inner product:

$$s_{Cos} = \frac{\sum_{i=1}^{B} P_i Q_i}{\sqrt{\sum_{i=1}^{B} P_i^2}\sqrt{\sum_{i=1}^{B} Q_i^2}}$$

where, P and Q are the two probability density functions, each composed by the same number B of bins.

This metric is itself normalized between 0 and 1 and allows to add the measures referred to different parameters without rescaling them regardless of their range of variation.

Hence, two similarity measures have been evaluated for each parameter: the similarity of the left part of the discharge with the safe operational space (*LEFTpart_simil*) and the similarity of the right part of the discharge again with the same safe operational space (*RIGHTpart_simil*). For a disrupted discharge, when approaching the actual pre-disruptive time $t_{pre-disr}$, it is expected that the right part distribution has similarity value close to 0, and the left part has similarity value close to 1. In fact, in such a case, in the left part the discharge is still in the non-disrupted phase, whereas in the right part it already shows a disruptive behavior.

These similarity measures are normalized with respect to the similarity of the whole flat-top phase (*Total_simil*), then truncates the values to 1; this adjustment makes the algorithm work for the shots where the signal range is very different from the safe one, even during the non-disrupted phase.

Subsequently, the normalized left part similarity is subtracted from the normalized right part similarity and the negative values are truncated to 0.

Then, the standard deviation of each plasma parameter is computed in a sliding window of 500 ms width, when the flat-top phase lasts more than 500ms, otherwise it is set equal to half flat-top length. Since the parameters may have different ranges, they are normalized between 0 and 1 before computing the standard deviation.

For each plasma parameter, an indicator is evaluated by weighing its standard deviation with the difference of the similarities. Hence, the parameter variations which do not produce a destabilization of the discharge are neglected.

Figure 5.9 shows, as an example, the construction of the indicator for the Rad_{pf_CVA} signal of the pulse #81916. Figure 5.9a) reports the signal $\mathrm{Rad_{pf-CVA}}$ (blue) and the same signal padded at the beginning and at the end (red dashed line) to avoid border effects processing the signal. Figure 5.9b) reports the normalized left part similarity (in blue), the normalized right part similarity (in red), and the difference between the blue and red signals (in yellow), where negative values are truncated to 0. Figure 5.9c) reports the Rad_{pf_CVA} standard deviation computed in the sliding window (red) and the Rad_{pf_CVA} indicator (in blue), computed as a time-by-time product between the yellow signal in Figure 5.9b) and the standard deviation.

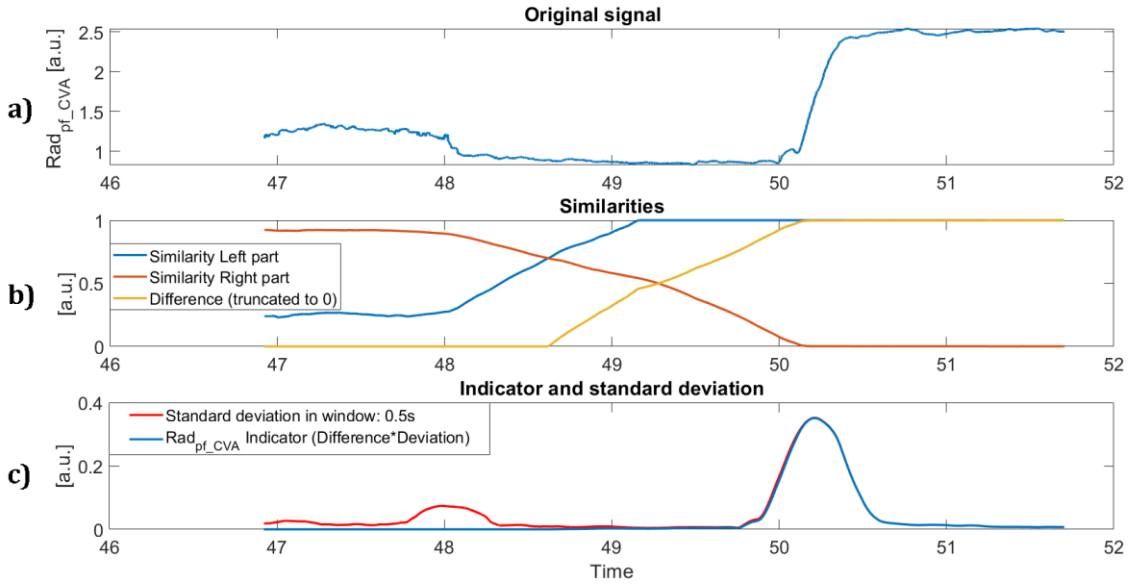

Figure 5.9 Construction of the indicator for the parameter Rad_{pf-CVA}, of the disrupted shot #81916: a) Rad_{pf-CVA} (blue), and Rad_{pf-CVA} padded at the beginning and at the end (red dashed); b) normalized LEFTpart_simil (blue), normalized RIGHTpart_simil (red), and their difference (yellow), where negative values are truncated to 0; c) standard deviation computed in a sliding window of variable length, adjusted depending on the signal length (maximum value is 0.5s) (red) and the indicator (blue).

It can be noted that, at around 48 s, the original signal varies and produces some peaks in the windowed standard deviation; these variations of the signal, on the other hand, are not moving the signal distribution outside the safe one: this determines a low value of the similarity difference and hence a low value of the indicator for the $\mathrm{Rad_{pf-CVA}}$. This is not true for the following variation at around 50s, which is the time when there is the beginning of the chain of event leading to the disruption. The indicator highlights the points where there is both a variation from the safe operational space and a variation in the signal trend. This is the reason why, in Figure 5.9c, the indicator grows at around 50.3s and then drop afterwards, due to the drop of the standard deviation.

Finally, an overall indicator (*Pre-disruptive Time Indicator* or *PTI*) is evaluated as the weighted sum of the single plasma parameter indicators. To set the parameter weights an optimization procedure has been performed, as described in the next subsection. Table 5.1 (last column) shows the finally adopted weights.

Figure 5.10 shows the *PTI* for the regularly terminated discharge #83747(a) and for the disrupted discharge #81916 (b), already considered in Figures 5.8 and 5.9. Note the different range of variation.

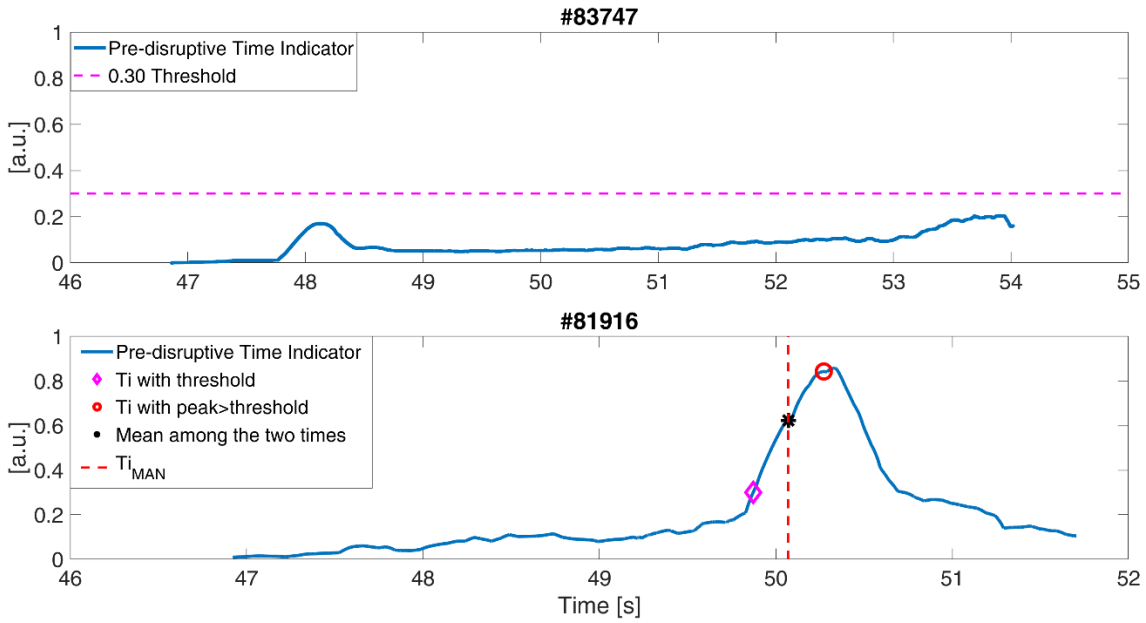

Figure 5.10. Overall Indicator: a) regularly terminated pulse #83437; b) for the disrupted pulse #81916.

Figure 5.11 reports the pseudo-code of the algorithm to construct the *PTI*.

```
PTIndicator = 0
For every signal:

  #Defining the window
  window_length = min(Signal_length/2, 500 ms)
  Window = [instant_t - window_length/2, instant_t + window_length/2]
  signal = padded_signal
  SIGNALweight = optimized weight of the indicator for this signal
  Total_simil = similarity(SIGNAL_distr,SAFE_distr)
  for every instant_t in time:
     #Defining the similarity
     LEFTpart_distr = distribution of signal before instant_t
     RIGHTpart_distr = distribution of signal after instant_t
     LEFTpart_simil = similarity(LEFT_part_distr,SAFE_distr)/Total_simil
     RIGHTpart_simil = similarity(RIGHTpart_distr,SAFE_distr)/Total_simil
     LEFTpart_simil and RIGHTpart_simil are truncated to 1
     SIMILmeasure = LEFTpart_simil-RIGHTpart_simil
     Negative values of SIMILmeasure are truncated to 0
     STD_win = standard deviation of signal in Window
  SIGNAL_indicator = SIMILmeasure * STD_win
  PTIndicator = PTIndicator + SIGNAL_indicator*SIGNALweight
```

Figure 5.11 Pseudo-code for the PTI

5.4 Thresholding the PTI

As expected, the ranges of variation of the *PTI* are very different among the safe and the disrupted pulses. Moreover, looking at Figure 5.10b, it can be noted that

the *PTI* highlights the moment when the features are varying, so that a threshold can be used to identify the onset of the chain of events leading to disruption.

Figure 5.12 shows the distribution of the values of the *PTI* for the safe pulses in the C28-C30 data set where the values 0.3 corresponds to the 99th percentile. Using this value as a threshold on the *PTI*, a pre-disrupted time of 50.02 s is obtained (magenta star in Figure 5.10b). Other criteria have been taken into consideration to detect the pre-disrupted time, such as the time corresponding to the first local maximum of the *PTI* greater than 0.3 (red star in Figure 5.10b), or the mean between the previous two.

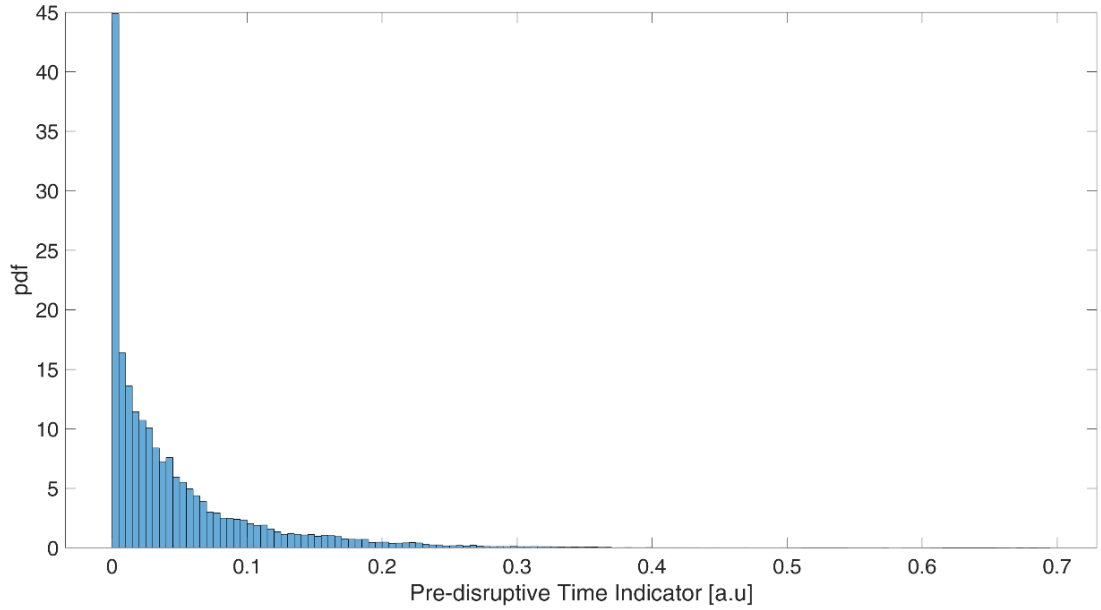

Figure 5.12: Probability density function of the PTI values for the regularly terminated pulses in the C28-C30 data set.

The best criterion is the mean between the time detected using the threshold equal to 0.3, with an assertion time of 20 ms, and the time of the first peak of the *PTI* greater than 0.3. It has been chosen in order to maximize the degree of separability of disrupted and non-disrupted regions in the GTM map.

Moreover, in order to consider disruptive processes characterized by fast time scales, which cannot be identified through the proposed statistical method, the mode locking occurrence has been also considered.

Finally, the pre-disruptive time has been identified as the lower time between the mode locking occurrence and the time obtained with the *PTI*. In this case, the value of the *PTI* may be greatly lower than the threshold.

Assuming such criterion on the *PTI*, the corresponding pre-disrupted time $t_{pre-disr,AUT}$ for the pulse #81916 is 50.075s, which is very close to the manually selected pre-disruptive time $t_{pre-disr,MAN}$ (50.07s) (see Figure 5.10b where this time is identified by the black star, whereas $t_{pre-disr,MAN}$ corresponds to the vertical red

dashed line). Furthermore, no pre-disrupted time is detected for the regularly terminated discharge #83747 (see Figure 5.10a).

5.5 Optimization of the algorithm parameters

As previously mentioned, the *PTI* is obtained as a weighted sum of the indicators of the plasma parameters in Table 5.1. Varying the weights leads to different pre-disrupted times, and therefore to different GTM maps. The optimal weights are reported in the last column of Table 5.1. They have been chosen, again, to maximize the degree of separability of disrupted and non-disrupted regions in the map, which means to minimize the percentage of samples falling in the mixed clusters of the GTM (grey clusters in Figure 5.13).

Figure 5.13a) shows the GTM ($\mathrm{GTM_{C28-C30-AUT}}$) trained using the pre-disrupted times $t_{pre-disr,AUT}$ obtained with the optimal weights reported in Table 5.1. Figure 5.13b) reports the GTM trained using the manually identified pre-disrupted times $t_{pre-disr,MAN}$ ($\mathrm{GTM_{C28\text{-}C30\text{-}MAN}}$).

The six parameters listed in Table 5.1 have been used to train both the GTMs. For the sake of comparison, the GTM hyperparameters, such as the number of latent points (2500), the number of radial basis functions (400) and their variance $\sigma = 0.8$, have been assumed equal to the ones used in [15], as well as the training set, which contains the same 89 disrupted shots and 70 regular terminations used in this thesis. However, unlike in [15], in Figure 5.13a) the pre-disruptive phase of the disrupted discharges has been identified using $t_{pre-disr,AUT}$ instead of $t_{pre-disr,MAN}$. It can be seen that, in both the maps, there is a well-defined separation between the two regions representing the disruptive (red) and non-disruptive (green) operational space. Moreover, the shape and the compositions of the two maps are quite similar (see Table 5.2): the percentage of samples falling in the mixed grey clusters differs by about 3% and the percentage of white clusters differs less than about 1%. Hence, it is expected that the two maps have quite similar performances when used as disruption predictors, as it will be shown in the next section.

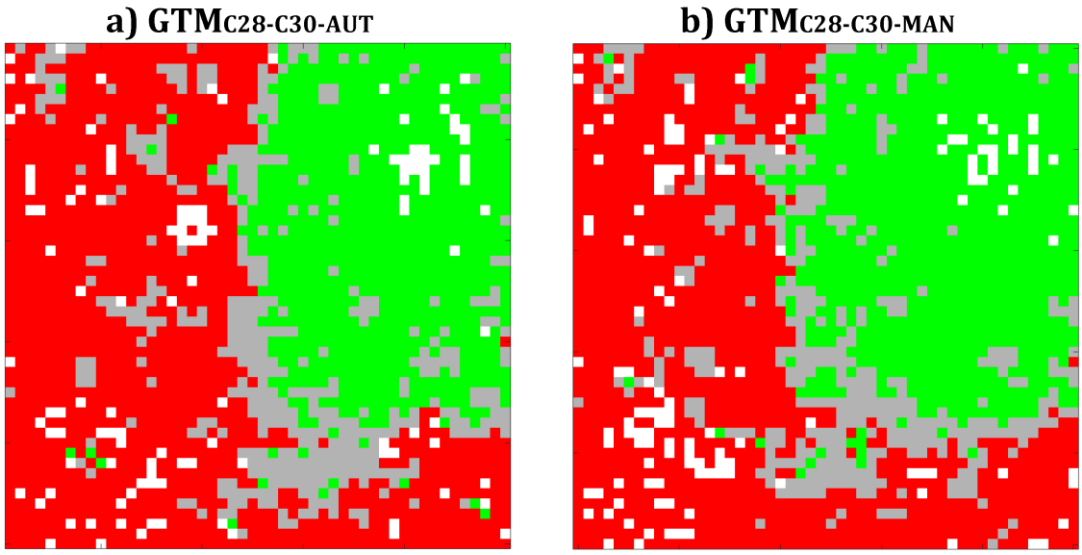

Figure 5.13: a) $GTM_{C28-C30,\ AUT}$ of the 6 plasma dimensionless parameters obtained using $t_{pre-disr,AUT}$ to determine the pre-disruptive samples; b) $GTM_{C28-C30,MAN}$ of the same parameters obtained using $t_{pre-disr,MAN}$.

Table 5.2: GTMs composition (using $T_{pre-disr,AUT}$ and $T_{pre-disr,MAN}$)

GTM	% safe samples belonging to safe (green) clusters	% disr. samples belonging to disr. (red) clusters	% samples in the grey clusters	% empty clusters
$\mathbf{GTM_{C28-C30-AUT}}$	75.25	81.51	21.47	4.92
$\mathbf{GTM_{C28-C30-MAN}}$	79.17	84.74	18.12	5.52

5.6 Prediction performances

In the disruption prediction literature, the performance of a predictive model is evaluated in terms of:

- Successful predictions (SP): pulses correctly predicted by the system (alarm for disruptions and no alarm for regularly terminated discharges);
- Missed alarms (MAs): disruptions for which the system does not provide any alarm;
- Tardy Detections (TDs): disruptive discharges where the detection is less than 10 ms before the disruption time;
- False alarms (FAs): regularly terminated discharges for which the system provides an alarm

Since disruption prediction systems are being developed especially for avoidance purposes, metrics such as premature alarms, defined as the alarms triggered at a prefixed time before the disruption, have become less significant in the definition of the system performance. Nowadays, the goal of an avoidance system is to associate the alarm to the presence of a destabilizing mechanism in the plasma, regardless of the distance of such event to the ending time t_{end}. Instead, the

prediction capability within the scope of avoidance and/or disruption control can be evaluated in terms of warning time, which represents the distance of the model alarm from t_{end}. A well-timed warning time allows the control system to react to the presence of an instability, while with a short warning time the disruption is generally mitigated by MGI. Thus, the premature alarm rate is replaced by the cumulative warning time distribution.

Figure 5.14 reports the cumulative warning time, that is the difference between the disruption time and the manual pre-disruptive time $t_{pre-disr,MAN}$ (in black) and automatic $t_{pre-disr,AUT}$ (in magenta). As can be noted, they follow quite the same trend confirming the validity of the proposed algorithm. Note that, in the construction of the algorithm, the pre-disruptive times $t_{pre-disr,MAN}$ have not been used. They were considered only as benchmarks values to evaluate the performance of the algorithm.

The same Figure 5.14 reports the cumulative warning time provided by the two GTMs in Figure 5.13 when used as disruption predictors on the entire C28-C30 data set adopting the same multiple condition alarm scheme in [15], shown in Figure 5.3. The cumulative warning time distribution reports the fraction of the shots that has an alarm time larger than a selected value. In particular, Figure 5.14 reports, in blue, the cumulative warning time provided by the GTM trained with the manually detected pre-disruptive times $T_{pre-disr,MAN}$ and, in orange (dashed), the alarms obtained using $t_{pre-disr,AUT}$. The cumulative warning times are almost overlapping with comparable prediction performance: the GTM trained with $T_{pre-disr,AUT}$ presents one missed alarm (0,7%), one tardy detection, and 3 false alarms (2,6%) on the datasets I and II, whereas the GTM trained with $t_{pre-disr,MAN}$ has one missed alarm, one tardy detection and 6 false alarms (6%) on the same dataset.

Figure 5.14 reports also the cumulative Locked Mode time, evaluated as the difference between disruption time and Locked mode onset time (in green). Note that, the alarm time is well in advance with respect to the time needed by the disruption mitigation valve (DMV, highlighted with a red vertical dashed line in Figure 5.14) to intervene, with more than 55% of the discharges predicted more than 1 second before the disruption time. Furthermore, very often, the proposed predictor is able to activate an alarm well in advance with respect to the Locked Mode trigger.

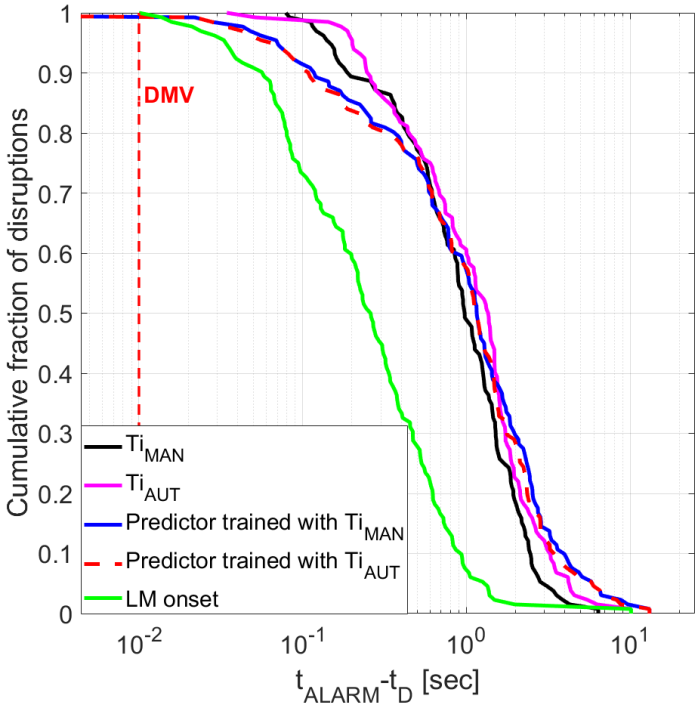

Figure 5.14: Cumulative warning time distributions for all the disrupted discharges in the training and test set of C28-C30 data set (the red vertical dashed line points out the DMV time, which allows to identify tardy detections).

The generalization capability of $\mathrm{GTM}_{\mathrm{C28-C30-AUT}}$ as disruption predictor has been evaluated on the C36 data set by projecting the 29 disrupted and 41 regular terminated discharges on the map. As expected, the prediction performance deteriorates with about 86% correct disruption predictions (1 missed alarm and 2 tardy detections) and 12% false alarms. Note that, 3 of the 5 false alarms are triggered by an abnormal increase of P_{frac} due to interruption of the additional heating system and could be avoided by inhibiting GTM response when this event occurs. On the other hand, we did not observe any premature detection of disrupted discharges generated by this issue. For the two tardy detections, it is observed a very late locked mode as disruption cause.

The deterioration is commonly observed in whatever data-based model, and even more so in the present case, due to the variation of the operational scenarios performed in the more recent campaigns. Figure 5.15 reports the probability density functions of the selected plasma parameters of the regularly terminated discharges in C28-C30 (blue) versus those in C36 (green) data sets, which show the variation of the plasma state especially for what concern Rad_{pf_XDIV}. This evidence confirms the need to regularly update the GTM model with data from more recent campaigns. To this purpose, the pre-disruptive times can be evaluated using the proposed algorithm avoiding the complex and time-consuming manual analysis.

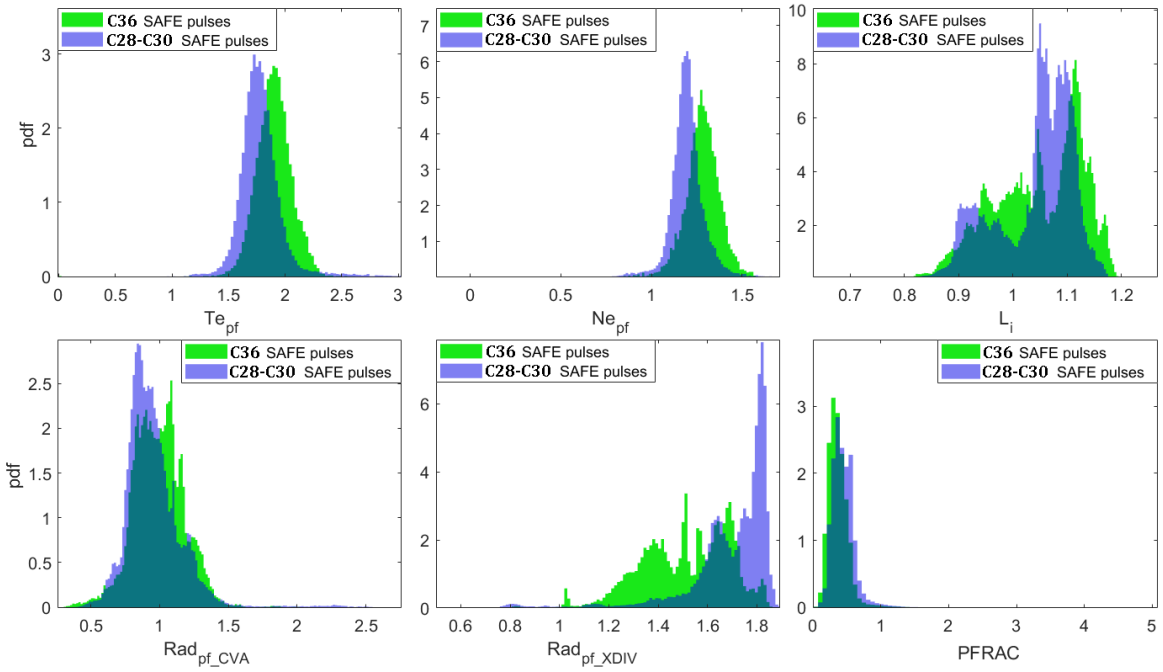

Figure 5.15: Probability density functions of the parameters of the regularly terminated discharges in C28-C30 (blue) versus those in C36 (green) data sets for (from top left to bottom right): electron temperature peaking factor, electron density peaking factor, internal inductance, radiation at the core peaking factor, radiation at the edge peaking factor, fraction of radiated power.

To confirm the robustness of the algorithm for automatically determining the pre-disruptive times, a statistical analysis of the selected plasma parameters has been performed on the discharges in the dataset II. Figure 5.16 reports the *pdf* of the selected parameters for the safe pulses (blue) versus the non-disrupted phase of the disruptive pulses (red), whereas Figure 5.17 reports the *pdf* of non-disruptive pulses (blue) versus the pre-disruptive phase of the disrupted pulses (red). Looking at Figure 5.17, it can be seen that, similarly to what observed in Figure 5.5, the *pdfs* of the pre-disruptive phases of the disrupted discharges shift with respect to the non-disrupted phases. The orange arrows in Figure 5.17 highlight the shifts. Hence, the proposed algorithm has been used to evaluate the pre-disruptive times, $t_{pre-disr,AUT}$, in the disrupted discharges of the dataset II.

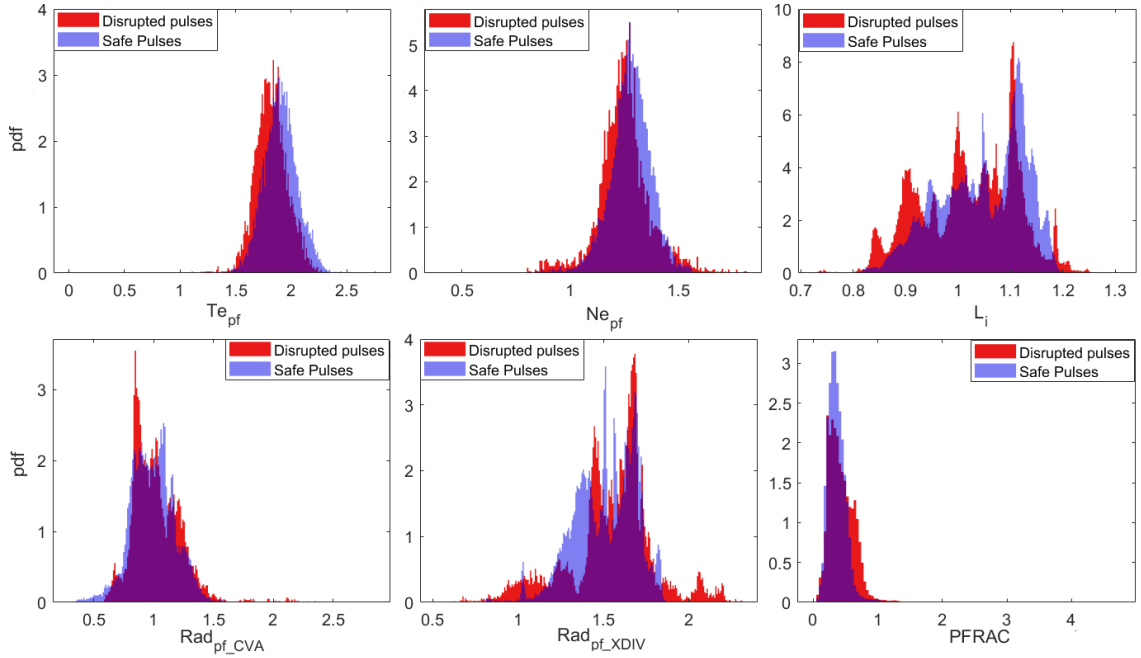

Figure 5.16: C36 data set: Probability density functions of the parameters of the safe pulses (blue) versus the non-disrupted phase (selected with the $t_{pre-disr,AUT}$) of the disruptive pulses (red) for (from top left to bottom right): electron temperature peaking factor, electron density peaking factor, internal inductance, radiation at the core peaking factor, radiation at the edge peaking factor, fraction of radiated power.

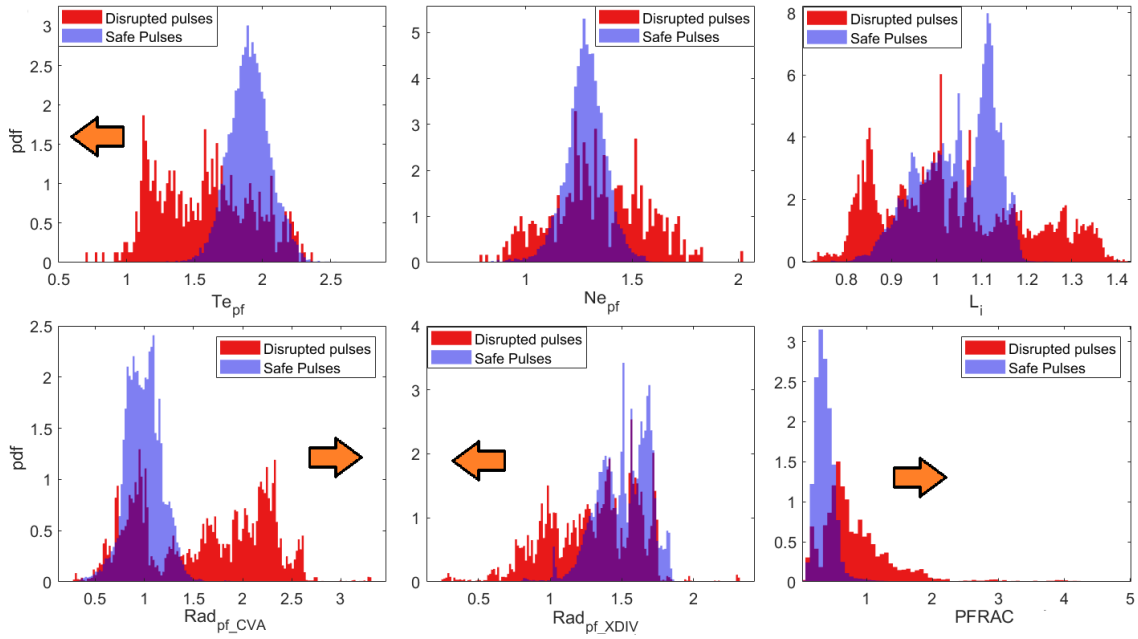

Figure 5.17. C36 data set: Probability density functions of the parameters of the safe pulses (blue) versus the pre-disruptive phase (selected with the $t_{pre-disr,AUT}$) of the disrupted pulses (red) for (from top left to bottom right): electron temperature peaking factor, electron density peaking factor, internal inductance, radiation at the core peaking factor, radiation at the edge peaking factor, fraction of radiated power. The shift of the distributions is marked with an orange arrow.

To validate the obtained $t_{pre-disr,AUT}$, a new GTM ($\mathrm{GTM}_{\mathrm{C36-AUT}}$) has been trained using all the pulses in the C36 data set, except two disruptions where the $t_{pre-disr,AUT}$ was not detected by the proposed algorithm.

Note that, being the GTM an unsupervised algorithm, the data are mapped only exploiting their intrinsic properties. The optimal GTM hyperparameters are the following:

- Number of latent points = 1024;
- Number of radial basis functions = 784;
- Variance σ = 1.2.

Figure 5.18a) reports the U-matrix representation of the GTM of the dataset II where a clear dark boundary between two lighter macro-clusters can be identified (highlighted with a black dashed line). Using the automatically evaluated pre-disruptive times, the GTM has been colored on the basis of the node composition and shown in Figure 5.18b). From Figure 5.18, it can be noted that the boundary in the U-matrix is very similar to the boundary between the green (safe) and the red (disrupted) regions. Moreover, the map performs a clear separation of the safe and disrupted regions with very high discrimination capability as reported in Table 5.3.

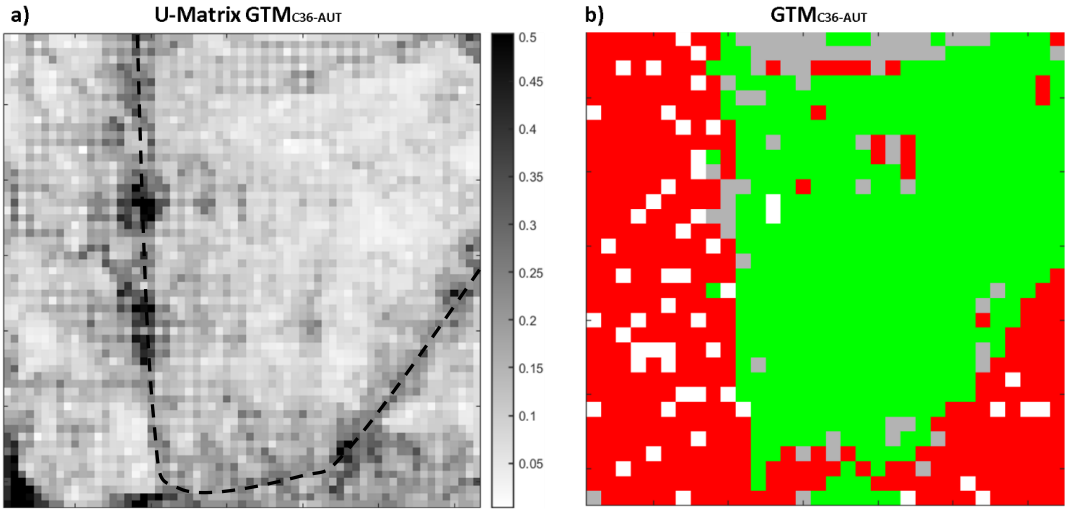

Figure 5.18: a) U-matrix of the $GTM_{C36-AUT}$. Lighter colors indicate smaller distance between clusters, while darker colors indicate higher distances. b) $GTM_{C36-AUT}$ obtained coloring the clusters using the automatically evaluated warning times $T_{pre-disr,AUT}$.

Table 5.3: GTM composition (using $t_{pre-disr,AUT}$)

GTM	% safe samples belonging to safe (green) clusters	% disr. samples belonging to disr. (red) clusters	% samples in the grey clusters	% empty clusters
$\mathbf{GTM_{C36-AUT}}$	96.45	86.34	8.06	5.08
$\mathbf{GTM_{C28-C36-AUT}}$	73.99	80.87	16.79	5.71

As an example, Figure 5.19 reports the temporal evolution of the disrupted discharge #90346, not used to train the $GTM_{C36-AUT}$: a) red (green) disrupted (non-disrupted) class membership function, which represents the percentage of samples of the disrupted and non-disrupted class respectively, in the cluster to which the sample is associated, with respect to the total number of samples in the cluster itself; b) trajectory of the discharge on the map. The circles depicting the evolution in time of the operating point are colored depending on the evolution time. The starting point is green, then the point becomes darker and darker as the discharge is approaching to the final point in red; c) Time evolution of the 6 plasma dimensionless parameters, together with the plasma current and the locked mode; the $GTM_{C36\text{-}AUT}$ alarm is marked with a vertical purple dashed line, the blue dashed line marks the mode lock time, and the red dashed line marks the disruption time t_D. The disruptive discharge starts in a non-disruptive cluster, firstly evolving in the non-disrupted (green) region, enters the disruptive (red) region, returns in the green region and enters, at the very end, in a disruptive cluster, which corresponds to the disruption time. For the considered discharge, the GTM identifies, according to what observed during the experimental session, an impurity accumulation pattern well in advance to the disruption time and triggers the alarm. Moreover, the trajectory on the map highlights the observed subsequent stable phase followed by a very fast disruption due to a mode lock. All the presented results confirm the validity of the algorithm proposed for the evaluation of the pre-disruptive times, mandatory for the updating of the model.

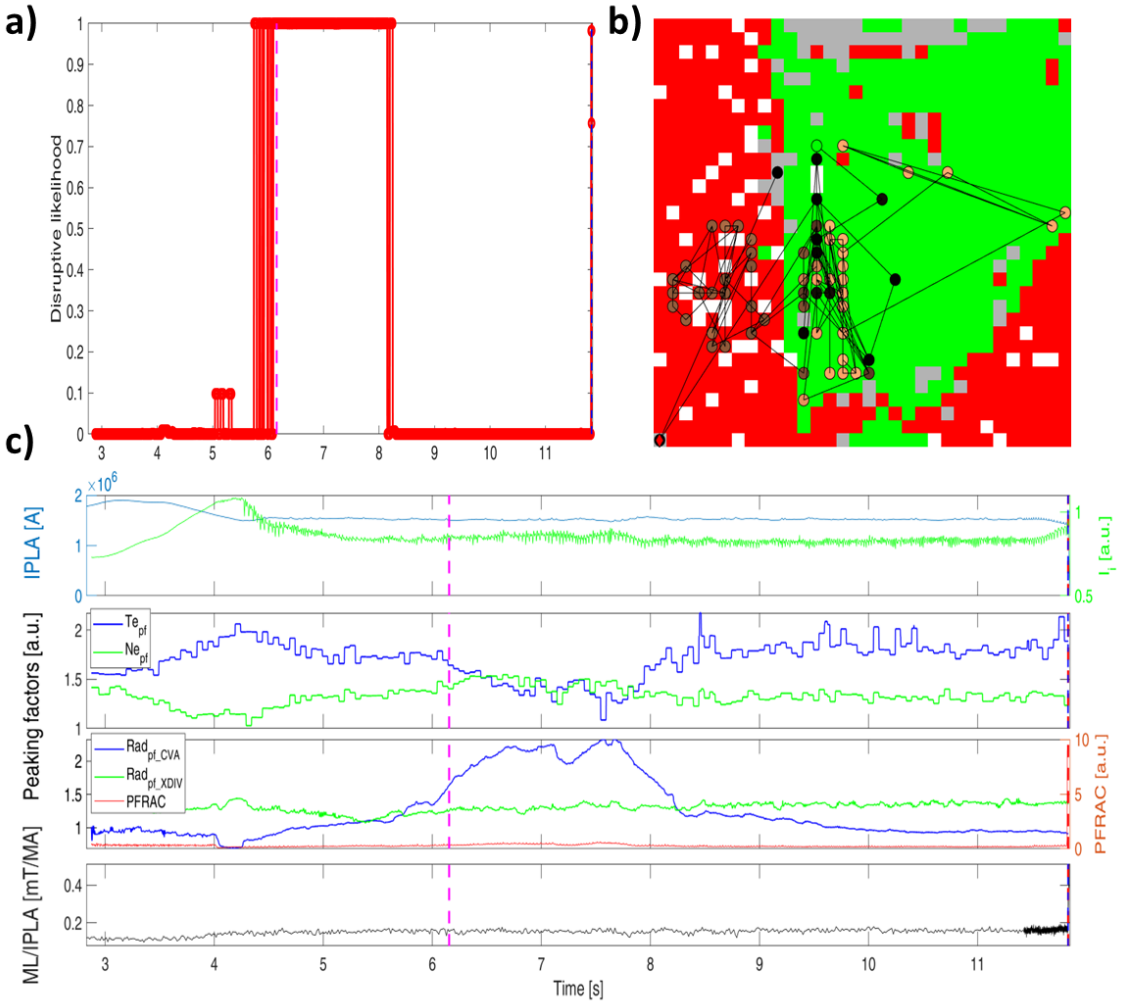

Figure 5.19. Disrupted discharge #90346: a) Disruptive likelihood of the of non-disrupted (green) and disrupted (red) classes; b) Projection on the map; the lighter points correspond to the beginning of the discharge, whereas the darker one corresponds to the end, at the disruption time t_D; c) Time evolution of the 6 plasma dimensionless parameters, together with the plasma current and the locked mode: the $GTM_{C36-AUT}$ alarm, corresponding to an impurity influx, is marked with a vertical magenta dashed line, the blue dashed line marks the mode lock time and the red dashed line marks the disruption time t_D.

5.7 Update of the GTM with C36 campaign data

As every machine learning algorithm, the GTM performance degrades as the operational space of the machine changes. This change can be highlighted by the statistical analysis reported in Table 5.4, which compares some plasma parameters of the regularly terminated discharges in the experimental campaigns performed at JET from 2011 to 2013 (C28-C30), those in 2016 (C36), and those in the more recent 2019-2020 campaigns (C38). As the JET operating scenario is continuously changing, an option to improve the performances is to upgrade the disruption predictor, and the automatic identification of the pre-disruptive phase becomes very useful to accelerate the labelling process and retraining the GTM. A new $GTM_{C28-C36-AUT}$ was developed using the C28-C30 and the C36 datasets, and it was

tested on the C38 campaign. Figure 5.20 shows the GTM map and Table 5.3: GTM composition (using tpre−disr, AUT) its composition. It is possible to see that the green area of the $GTM_{C28-C36-AUT}$ covers a portion of the space which was grey in the lower central part of the $GTM_{C28-C30-AUT}$. Moreover, the grey area is slightly extended towards the red region and more grey clusters cover the center of the map. Regarding the performance, Figure 5.21 shows the accumulated fraction of detected disruptions against the warning time and Table 5.5 shows the rate of MAs, FAs and TDs of the 2 GTMs with the same alarm scheme shown in Figure 5.3.

Table 5.4: Ranges of the plasma parameters over the three considered sets of regularly terminated discharges

	C28-C30		C36		C38	
Plasma Parameter	Min	Max	Min	Max	Min	Max
Plasma Current [MA]	1.448	2.983	1.633	3.273	2.261	3.545
Poloidal beta [a.u.]	0.096	0.971	0.125	0.760	0.126	0.669
Total Input Power [MW]	0.715	21.676	0.196	30.453	1.277	36.010
Total Radiated Power [MW]	0.100	7.715	0.100	12.657	0.532	22.608
Safety factor q95 [a.u.]	2.328	4.917	2.571	5.476	2.936	3.810
Line Integrated Density [10^{19} m^{-2}]	2.763	22.099	2.876	23.632	3.296	23.570
Temperature peaking factor [a.u.] *	1.157	3.051	1.109	2.613	1.442	2.395
Density peaking factor [a.u.] *	0.762	1.625	0.706	1.714	1.097	1.753
Radiation peaking factor: Core-Versus-All [a.u.] *	0.441	2.278	0.365	1.580	0.627	1.704
Radiation peaking factor: Divertor [a.u.] *	0.760	1.896	0.803	1.857	0.609	1.730
Internal Inductance [a.u.]	0.836	1.224	0.822	1.190	0.780	1.105

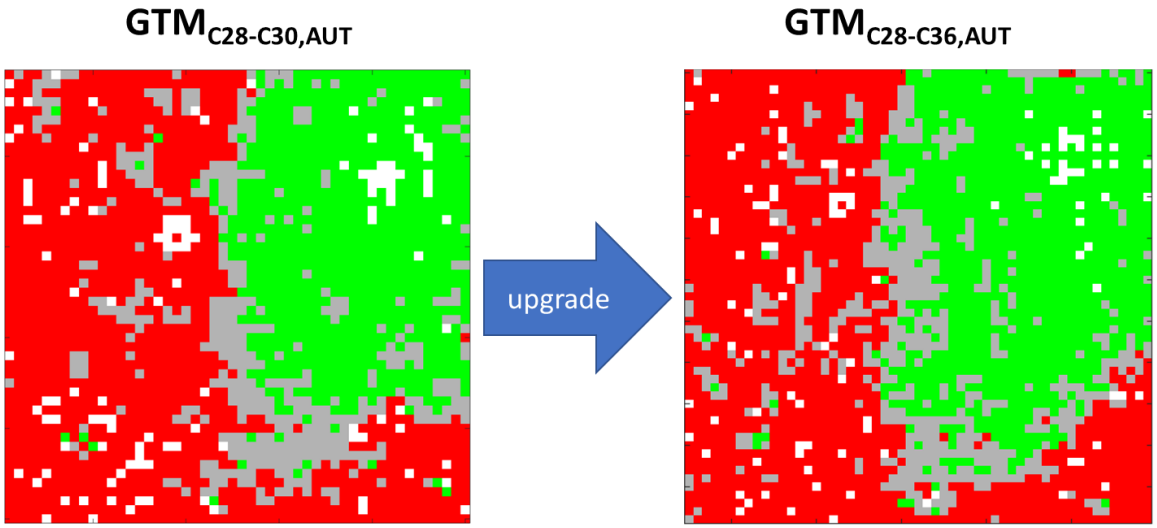

Figure 5.20: upgrade of the GTM with the pulses from the dataset II

Table 5.5: Performances of the GTM in the Dataset III

GTM	TD	MA	FA
$GTM_{C28-C30-AUT}$	0%	0%	49.20%
$GTM_{C28-C36-AUT}$	2.7%	0%	11.11%

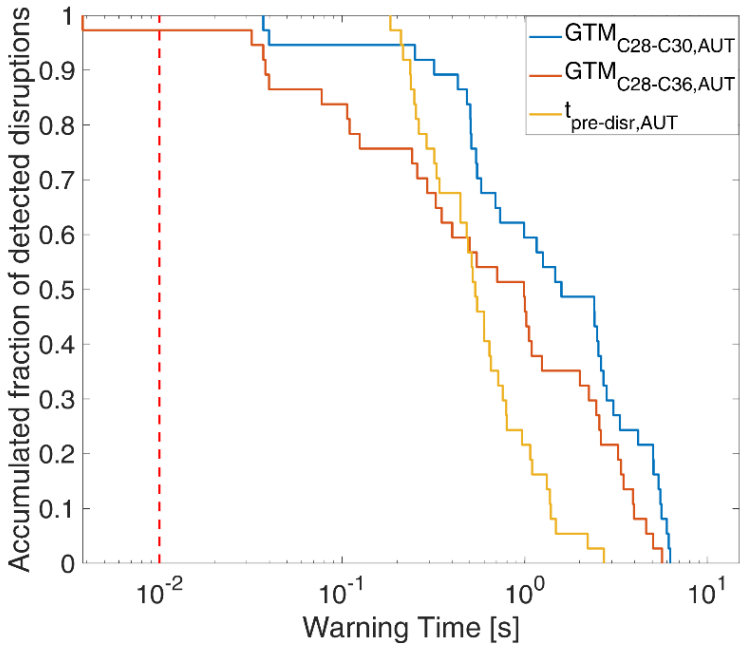

Figure 5.21: Cumulative warning time distributions for all the disrupted discharges in the Dataset III (the red vertical dashed line points out the DMV time, which allows to identify tardy detections).

5.8 Conclusions

In this Chapter the disruption prediction model based on the GTM Machine Learning method has been presented and an algorithm for the automatic identification of the pre-disruptive phase of tokamak discharges has been proposed. Presently, a general physical model for clearly recognizing disruptive behavior does not exist, and this sometimes produces ambiguity on the manual classification task as well. Hence, the interest is not only towards the classification task (as a plethora of different models exist, and many of them provide satisfying performances) but also in the properties of the parameter space where the relevant disruption physics takes place, its visualization and interpretative analysis. The encouraging results led to the use of the automatic pre-disruptive times as the new inputs of the GTM algorithm, in place of the manually detected ones. The shape and the composition of the GTMs trained with the manual and the automatic ones were comparable, as well as the data distribution obtained with the mapping and univariate analysis of the signals.

The results obtained with the GTM confirm the efficacy of the method and validate the proposed algorithm. The general principle of the algorithm seemed to work quite well, leading to a coherent discrimination of the non-disrupted and pre-disruptive phases of discharges, also referring to more recent experimental campaigns. Machine Learning models generally suffer from ageing whether the operational space of the machine changes, and this is also valid for different experimental campaigns, as the machine is reconfigured for new experiments. The presented results, together with the map composition, confirms the possibility to avoid the cumbersome and time-consuming identification of the pre-disruptive times and to implement a continuous learning system. The algorithm for the automatic identification of the precursors times, together with a set of data analysis and clustering algorithms, has also be used for retraining the GTM model extending the training space to the C36 data, improving the performance on a later campaign (C38, Dataset III).

Chapter 6
Disruption prediction with Fully Connected Neural Networks

6.1 Introduction

This Chapter discusses the development of Fully Connected Neural Networks (FC-NN) models for the disruption prediction task. The model reaches very high performance in the task and is a widely adopted one in disruption prediction [92], [96], [98]–[100].

6.2 Data preparation

The list of signals provided as input to the FC-NN is presented in Table 6.1. The peaking factors of the electron temperature, density and radiation are obtained as described in Chapter 1. Moreover, the plasma internal inductance provides information on the current profile, the fraction of radiated power is a dimensionless indicator of the power balance, and the normalized locked mode allows to detect the insurgence of mode locking.

Table 6.1 Diagnostic signals, acronyms and units

Plasma signal	Acronym	Diagnostics
Electron Temperature Peaking Factor	Te_{pf}	HRTS
Electron Density Peaking Factor	Ne_{pf}	HRTS
Peaking Factor of the Radiation (excluding the contribution of the X-point/divertor region)	RAD_{pf-CVA}	Bolometer
Peaking Factor of the Radiation (excluding the contribution of the core region)	$RAD_{pf-XDIV}$	Bolometer
Plasma Internal Inductance	l_i	BetaLi
Fraction of radiated power	P_{FRAC}	Bolometer, BetaLi
Normalized Locked Mode	LM_{norm}	Saddle Coils

6.3 Training of the model

To train a FC-NN model, examples of both disrupted and non-disrupted plasma states (discharge time samples) have be collected, as for the previously presented models, using the algorithm presented in Chapter 5 determining a consistent value of $t_{pre-disr}$ ($t_{pre-disr-AUT}$) for the different disruptions [74].

Referring to the binary classification of disrupted or non-disrupted samples, the FC-NN can model the non-linear relationship among the input feature vector $\boldsymbol{x}$ and the corresponding output y, which encodes the classification of the discharge sample. The training set is obtained starting from almost the same discharges (85 disrupted shots and 70 regular discharges) of the JET-ILW campaign used to train the GTM, but 22 disrupted and 16 non-disrupted discharges are removed from the

training set to build a validation set. The inputs provided to the FC-NN are the same as the one provided to the GTM but including the ML_{norm} in the set of input features. The output of the FC-NN is the disruptive likelihood. A threshold on the likelihood is optimized by analyzing the FC-NN performances in the training and validation sets. In Table 6.2, the training parameters of the FC-NN model are reported, where the parameter s determines the change in the weight for the second derivative approximation, and the parameter l regulates the indefiniteness of the Hessian [134]. The architecture of the FC-NN has an input layer, one hidden layer with sigmoid activation function and an output layer with 2 neurons and a SoftMax activation function. The hidden layer size has been optimized by scanning the number of neurons and optimizing the validation performance. The threshold of the FC-NN model is also set to 0.995 by optimizing the performance on the validation and training sets.

Table 6.2 FC-NN training parameters.

Parameters	Value
Optimizer	Back Propagation and Scaled Conjugate Gradient algorithm [134]
Number of input neurons	7
Number of hidden neurons	10
Number of output neurons	1
Weights Initialization	Random
Learning rate (s, l)	$(5 \cdot 10^{-5}, 5 \cdot 10^{-7})$
Best epoch	23
Validation stop (consecutive evaluations)	75

6.4 Model performance

In Figure 6.1a the input features are shown for a test disrupted discharge (#94775) belonging to a recent JET campaign, temporally far from those used in the model training. The disruptive likelihood is reported in the same Figure 6.1b. An alarm is triggered when the disruptive likelihood overcomes the threshold. The FC-NN has no Assertion Time, which can be introduced to avoid wrong alarms due to spikes in the disruptive likelihood. This means that optimizing the threshold of the model is sufficient to make the MLP response robust, in terms of detecting the presence of disruption precursors, while keeping the number of false alarms low. The vertical dashed line in Figure 6.1 identifies the alarm time t_{alarm}, resulting in a warning time $\Delta t_{warning}$= 408 ms. The overall performance of the model, computed in terms of Missed Alarms, False Alarms and Tardy Alarms, is reported in Table 6.3.

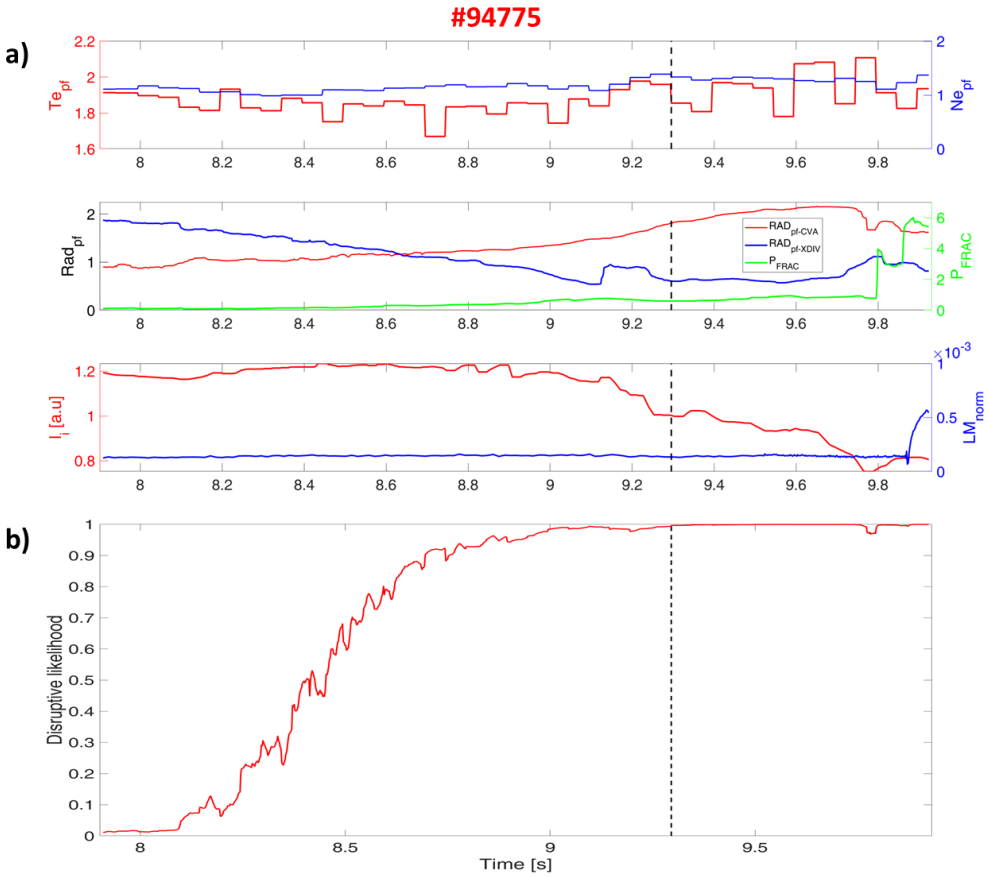

Figure 6.1. JET disrupted discharge #94775: a) Time evolution of the seven plasma dimensionless parameters: temperature (Te_{pf}), plasma density (ne_{pf}), and radiated power (Rad_{pf-CVA}), and $Rad_{pf-XDIV}$ peaking factors, internal inductance l_i, fraction of radiated power P_{frac}, normalized Locked Mode amplitude LM_{norm} signal; b) Disruptive likelihood of the disrupted discharge #94218 supplied by MLP. The dashed black line identifies the alarm time.

Table 6.3: Performance of FC-NN in the Training and Test sets

Set	SP [%]	MA[%]	FA [%]
Train	99.15	0	1.85
Test	96.11	2.78	3.36

Chapter 7
Disruption prediction with Convolutional Neural Networks

7.1 Introduction

As discussed in Chapter 6, FC-NN allow to reach very good performance in the disruption prediction task in a database spanning several years of JET operation. However, the inputs of the FC-NN are synthetic features extracted from the plasma profiles, whose definition may vary depending on the device. In recent years, the use of deep learning in research has increased significantly, due to the improved capability of computers in processing massive amounts of data and to the ability of deep neural networks in producing high accuracy performances even without a feature extraction procedure. Among the architectures in deep learning able to process images, CNNs are the most used [35], [135]. For this reason, the PhD activity also addressed the development of Convolutional Neural Networks (CNNs), which can process the full information from the plasma profiles. Since the neural networks are supervised algorithm, the training data has to be classified before the training of the model. A label has to be explicitly assigned to each sample in the dataset. All the segments belonging to the regularly terminated discharges have been labelled as “stable”. For each disruptive discharge, the labelling of the “unstable” has been carried out by automatically identifying the pre-disruptive phase by means the algorithm discussed in Chapter 5.

7.2 CNN data generation-processing subsampling

In order to reduce the unbalance between the stable and unstable classes, caused by the different duration of the two phases, the overlap times of the sliding window for the regularly terminated and disrupted discharges have been differently chosen. Due to the low time resolution of the HRTS, only one segment every 24 ms has been extracted from the pre-disrupted phase of the disrupted discharges, whereas one segment every 150 ms has been retained from the regularly terminated discharges. Note that, during testing, a sliding window of 200ms with a stride of 2ms, for all discharges (regularly terminated and disrupted), has been used.

Table 7.1 reports the total number of pulses and time slices sampled for the train, validation and test sets. The validation set was used to monitor the training performance during the training and to perform an early stop if the performance on the validation data would not improve. The validation set discharges were randomly sampled among the 85 disrupted discharges and the 70 regularly terminated ones from Dataset I.

Table 7.1: number of pulses and time slices in the training, validation and test sets

Set	Disruptions		Regular pulses	
	Pulses	Time slices	Pulses	Time slices
Training	63	3698	54	4239
Validation	22	1191	16	1381
Test	108	313392	149	588143

7.3 Early Fusion Architecture

As previously cited, the deep architecture of a CNN normally consists of a cascade of blocks of different layers which performs a filtering of an input image to extract significant features from it [34]. The features are produced by a cascade of filtering blocks, interconnected through nonlinear activation functions (typically a Rectified Linear Unit), and a multi-layer perceptron combines them to produce the output of the network. A dropout layer is usually inserted before the multi-layer perceptron in order to reduce overfitting on the training set and improve generalization. The architecture of the proposed CNN is shown in Figure 7.1.

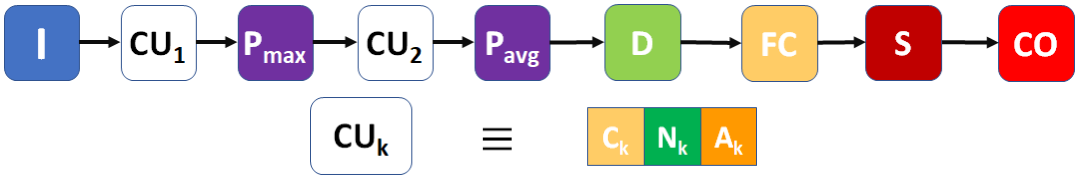

Figure 7.1: CNN architecture, where: I is the image input; CU_k is the kth convolutional unit, composed by the cascade of a convolutional layer (C_k), a batch-normalization layer (N_k) and a nonlinear activation layer with ReLU functions (A_k); P_{max} and P_{avg} are the max-pooling and average-pooling layers, respectively; D is a dropout layer; FC is a fully-connected layer; S and CO are the SoftMax and classification output layers, respectively.

A first convolutional unit (CU_1) followed by a max pooling layer (P_{max}), with pool size and stride 8×1, filters out vertically (along the "spatial" dimension) the input image by reducing the size from 132x101 to 16x101. A second convolutional unit (CU_2) followed by an average pooling layer (P_{avg}), with pool size and stride 1×12, filters out horizontally (along the "time" dimension) the resulting image by reducing the image size to 16x20. The two convolutional units (CU_1 and CU_2) are made out 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 two convolutional layers have one single filter (1-channel kernel) of size 5x1 and 1x11, respectively. The output of the 2nd convolutional layer is then a 16x20 image, which is flattened and provided as input to a fully connected layer (FC). Finally, the FC layer processes the 320 features and feeds a SoftMax layer (S) for classification (CO). A dropout layer with dropout probability of 20% has been included before the fully connected layer in order to reduce overfitting on the training set and improve generalization.

In order to include also the information given by the two 0-D signals, i.e., l_i and ML_{norm}, two segments of size 1x101 have been added as input to the second convolutional unit and concatenated with the output image produced by the max pooling layer (see Figure 7.2). As a result, the output of the average pooling layer has size 18x20 and 40 additional features coming from the two signals are processed by the fully connected layer, combined with the remaining 320 features and used for the classification.

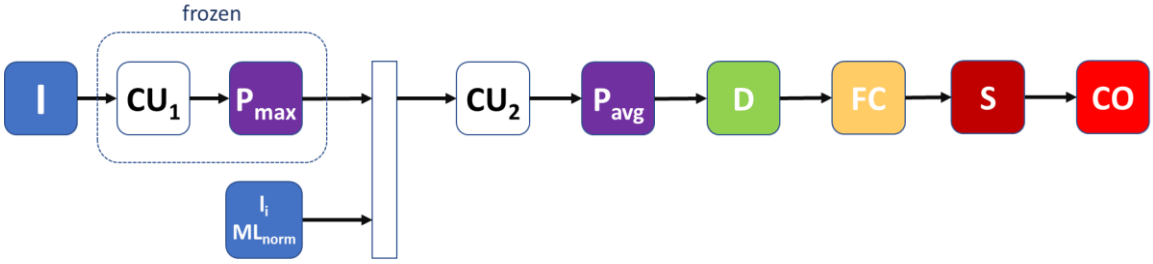

Figure 7.2: Modified CNN architecture, where the internal inductance (li) and the normalized locked mode (MLnorm) are added as input to the second convolutional unit and concatenated with the output image produced by the max pooling layer.

The Soft Max (S) layer produces the likelihood of the input segment to belong to a regularly terminated or a disrupted discharge, and the disruptive likelihood is used to trigger a disruption alarm. As an example, Figure 7.3 shows the disruptive likelihood output for a JET disrupted pulse, which starts to rise in correspondence of the $t_{pre-disr}$ (dashed magenta line) and then it straightforwardly reaches the value 1. The last classification layer (CO) implements a threshold on the disruptive likelihood to perform the final classification. Such alarm threshold has been optimized by means of a heuristic procedure, maximizing the number of correct predictions on the training and validation discharges. The optimal threshold is found to be 0.89 and the alarm time is triggered when the disrupted likelihood overcomes such threshold. In Figure 7.3 the alarm time is identified by the black vertical dashed line.

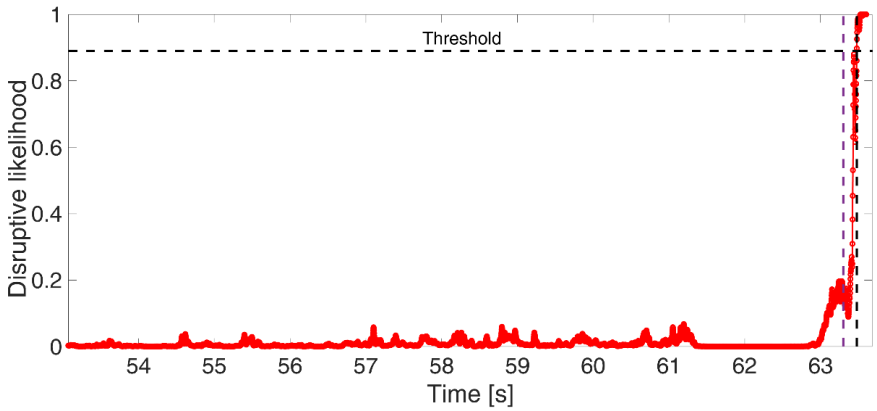

Figure 7.3: CNN likelihood curves for a disrupted pulse, where the red line is the disruptive likelihood. The dashed black line indicates the CNN alarm time, whereas the dashed magenta line indicates $t_{pre-disr}$.

To limit the complexity of the training procedure a first training step has been performed using only the 1-D diagnostics. Then, the CU_1 and the P_{max} blocks were frozen. In a second training procedure, only the second convolutional block and the fully connected layer were trained, using both the 1-D and 0-D diagnostics. This approach greatly reduces the computation time of the training procedure because it reduces the number of parameter updates being made without having a major impact on the network's accuracy [136]. Table 7.2 shows the hyperparameters of the two training procedures.

Table 7.2: Training parameters of the CNN model

Parameters	1st Training (Images only)	2nd Training (Images + signals)
Optimizer	Stochastic gradient descent with momentum	Stochastic gradient descent with momentum
Initial learning rate	2.5e-4	1e-4
Learning rate drop factor	0.1	0.15
Learning rate drop period (epochs)	20	10
Momentum	0.9	0.9
MiniBatch size	16	16
Validation frequency (iterations)	50	50
Validation stop (consecutive evaluations)	100	100
Weight decay (L2 regularization)	1e-4	1e-4

Note that, the network architecture allows us to independently process the two dimensions, spatial and temporal: in fact, the first two blocks (CU_1 and P_{max}) filter only across the spatial direction, while the second two (CU_2 and P_{avg}) filter only across time. This allows to easily concatenate the signals (I_i and ML_{norm}) to the image features processed by the first convolutional and pooling blocks, so that the temporal synchronization is preserved. The vertical kernel size for the convolutional and pooling blocks was designed considering a few constraints: a kernel size equal or larger than 24 would have been larger than the bolometer number of lines of sight, and a small size kernel would reduce the effect of the discontinuity between the stacked diagnostic images. The small kernel size (5x1) allows the network to still identify changes in the spatial dimension of the HRTS scattering profile. Regarding the time filtering, a similar operation was performed: due to the different time resolution of the diagnostics employed, the filter size has been chosen to mainly process the higher frequency signals, such as the bolometer data. The pooling type

was optimized: a network with only average pooling was trained, as well as one with only the max-pooling. Analysing the performances on the training and the validation set, the average pooling had lower performance than the max-pooling, but the max-pooling response was too sensitive to transient changes in the data time traces. Hence, the max-pooling layer was left in the spatial processing block (vertical pooling), while the average pooling was selected for the temporal pooling.

7.4 Predictor performance

In this section, the potentialities of the CNN model to detect a disruptive behavior early enough to enable avoidance actions are presented. The performance in terms of SPs, MAs, and FAs rate of the proposed predictor is reported in Table 7.3, for a training set composed by 63 disruptive and 54 regularly terminated pulses and a test set of 108 disruptive and 149 regularly terminated pulses.

Table 7.3: CNN predictor performance

Dataset	SP	MA	FA
Train	98%	0%	4.28%
Test	93%	3.7%	9.40%

Table 7.3 highlights that the CNN has very good predictive performance with a successful prediction rate of 93% and a FA rate below 10% on the test set. An example of the CNN capability in predicting disruptions can be seen in Figure 7.4, which refers to the test pulse #92226 (outside the training range). The CNN output in Figure 7.4a identifies a rising of the disruptive likelihood at about 11.20 s, accordingly with the visible change of the plasma behavior across the input profiles shown in Figure 7.4d-f. It can be noted that, at the plasma core, the electron temperature collapses (Figure 7.4e) while the electron density peaks (Figure 7.4f). This phenomenon is accompanied by strong radiation (Figure 7.4d). This pattern is well-known as the impurity accumulation disruptive mechanism [72], [137], and it is typical of the JET ILW disruptions due to the penetration of high-Z impurities (such as the Tungsten of the JET divertor) into the plasma core [12].

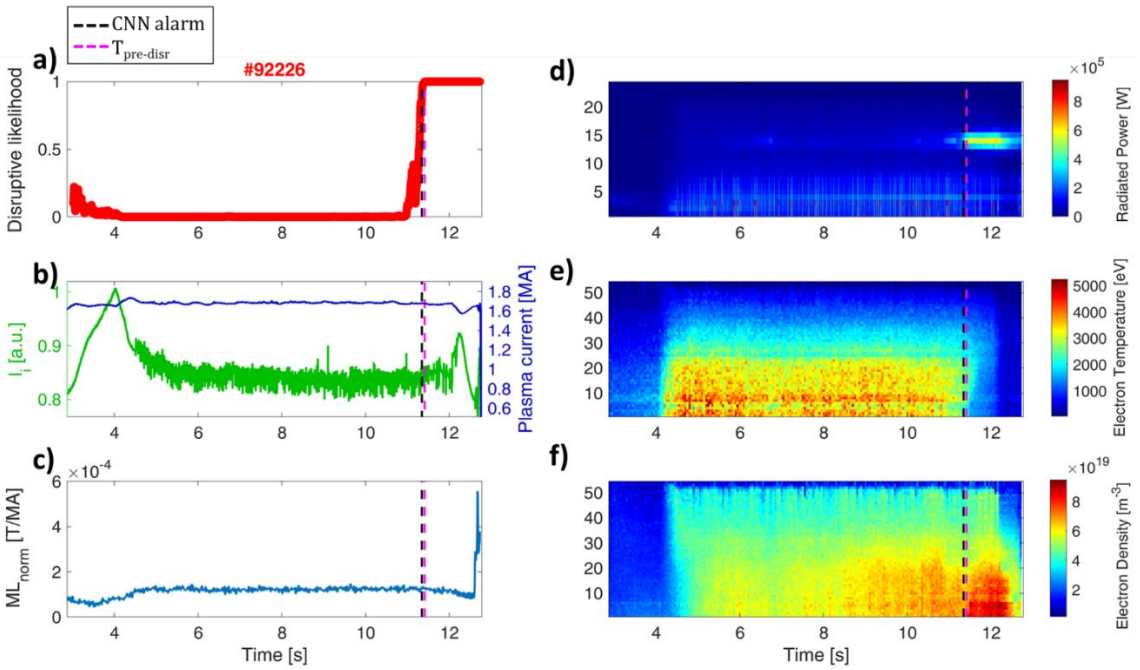

Figure 7.4: JET disrupted discharge #92226 a) CNN disruptive likelihood. The dashed black line indicates the CNN alarm time, while the dashed magenta line indicates the $t_{pre-disr}$; b) Internal inductance, in green, and plasma current in blue; c) mode lock normalized by the plasma current, in blue; d) Radiated power from the Bolometer; e) Electron temperature from the HRTS; f) Electron Density from the HRTS.

A typical regularly terminated discharge, as the one shown in Figure 7.5, is usually characterized by very regular radiated power profiles with low radiation from the central chords of the bolometer horizontal camera as visible in Figure 7.5d. At the plasma core, the electron temperature profile peaks (Figure 7.5e), while the electron density, distributed across the profile, presents slightly higher values (Figure 7.5e). In addition, l_i and ML_{norm} do not reveal any approaching disruption. In agreement, the disrupted likelihood never reaches 0.5.

The cumulative warning time distribution is shown in Figure 7.6. The blue and black lines in Figure 7.6 show the CNN warning times in the training set and test set respectively. Moreover, to evaluate the suitability of the alarm triggered by the CNN predictor, the predicted warning time is compared with respect to the one defined by $t_{pre-disr,AUT}$. In this regard, in the same Figure 7.6, the dark red and yellow dashed lines report the $t_{pre-disr,AUT}$ warning time distribution for the training and test pulses, respectively. The vertical red dashed line indicates the minimum warning time (10 ms) necessary at JET to adopt mitigation actions. Detections made after this line can be considered as tardy alarms.

From Figure 7.6, it is possible to see that the CNN warning times and $t_{pre-disr,AUT}$ ones are quite close, both for the training and the test set. This means that, in most of the cases, the CNN detections are coherent with the instability mechanisms automatically detected with the $t_{pre-disr,AUT}$.

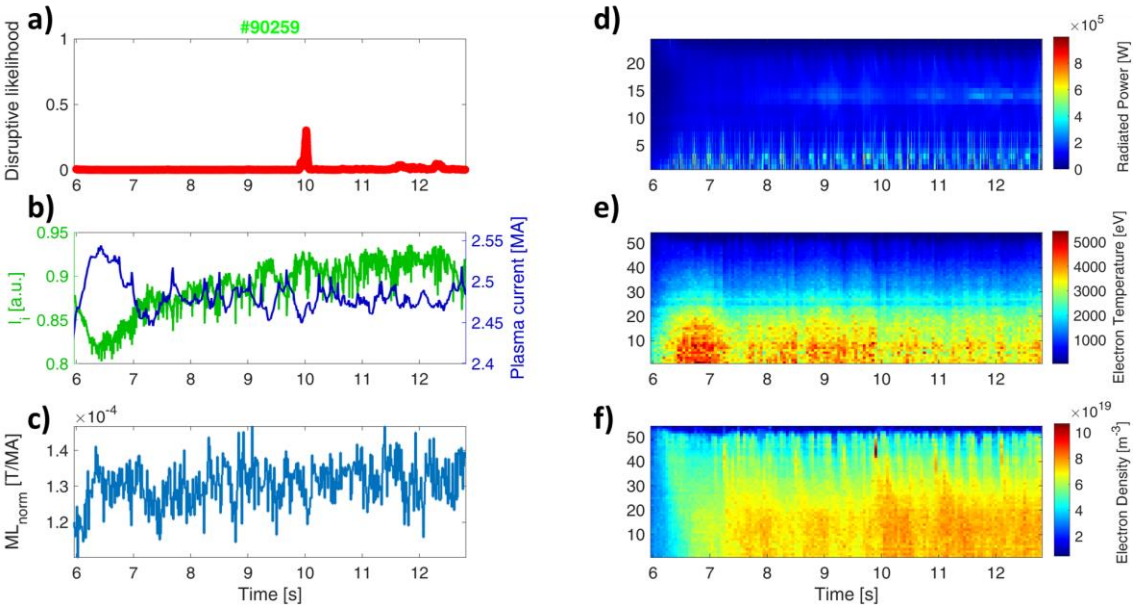

Figure 7.5: CNN output of the JET regularly terminated discharge #90259. a) CNN disruptive likelihood; b) Internal inductance, in green and plasma current in blue; c) mode lock normalized by the plasma current, in blue; d) Radiated power from the Bolometer; e) Electron temperature from the HRTS; f) Electron Density from the HRTS.

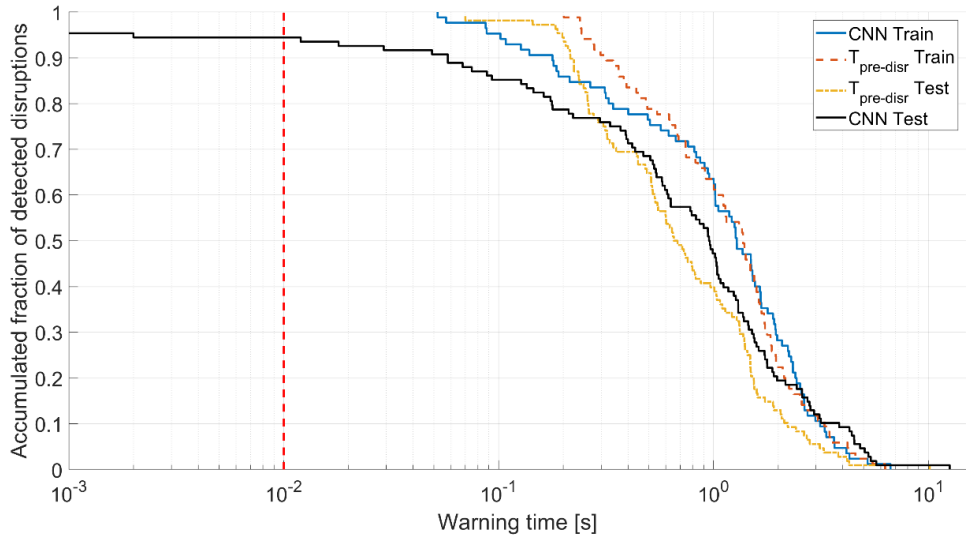

Figure 7.6: Cumulative fraction of detected disruptions by the CNN model versus the warning time in the training and in the test set. The vertical red dashed line allows us to identify tardy detections.

As described in Chapter 4, the database employed in this work includes discharges from several experimental campaigns and different experimental conditions. This motivated an in-depth analysis of the results, to investigate a possible degradation of the CNN performances with the changing of the operating conditions. Table 7.4 reports the SPs, MAs and FAs rate for the test discharges, among the three datasets. As expected, the best performances are reached on the Dataset I, which covers the same pulse range of the training data. Note that, the CNN predictor tested on Dataset II still performs quite well, with a FA rate lower

than 7%, whereas an increase in the MA is observed. The degradation of the MA rate is due to the presence of some discharges which disrupt very abruptly because of a sudden locking mode, as already observed in [115] for the same data set. Conversely, in the Dataset III the errors on the disrupted pulses are extremely low, whereas the false alarm rate is the highest. An explanation for this difference should be sought in the new region of the operational space covered by the regularly terminated pulses belonging to this dataset, which are characterized by higher input power and electron density (as highlighted in Figure 7.7c). Indeed, by comparing the distributions of the features provided to the model in the three datasets, the regularly terminated pulses in the Dataset III are characterized by higher n_e and radiated power values and lower l_i values. In Figure 7.8 a-c, Figure 7.9a-c and Figure 7.10a-c, the probability density functions of the average density across the plasma radius, *li* and the average radiated power are reported for the three considered datasets, for regularly terminated (green) and disruptive (red) pulses, respectively. In addition, for the Dataset III, the related distributions of the false alarms are added to the regularly terminated and disruptive ones (see Figure 7.8c-Figure 7.10c) and identified by a magenta dashed line. It can be noted that the distribution values of the three features for the regularly terminated pulses in Dataset III are shifted with respect to the ones of the previous datasets. Considering that the full training set of the model is contained in Dataset I, the distribution of the values in the Dataset III is then covering ranges poorly represented by the non-disruptive behavior of the training set. This trend is confirmed or accentuated by the distribution of the feature values related to FAs. Moreover, during high power experiments the presence of localized radiation in the outer half of the plasma, not necessarily correlated to the onset of a disruptive mechanism, has been observed [39]. This phenomenon can play a crucial role in the erroneously detection of a disruptive behavior in a regularly terminated pulse, as in the pulse #94785 reported in Figure 7.11. Indeed, at around 11.5 s, despite a non-disruptive behavior shown by the HRTS profiles and the 0-D signals, a high radiation seen from the central lines of sight of the bolometer horizontal camera triggers a FA (see Figure 7.11d). From the radiation profile recorded by the bolometer vertical camera (not provided as input to the CNN) it is possible to localize the radiation blob in the outer half of the plasma, thus not related to any impurity accumulation.

Table 7.4: Performance of the CNN on the test discharges across the three datasets

Dataset I			Dataset II			Dataset III		
SP%	MA%	FA%	SP%	MA%	FA%	SP%	MA%	FA%
96.55	2.38	4.44	91.42	10.34	7.31	91	0	14.28

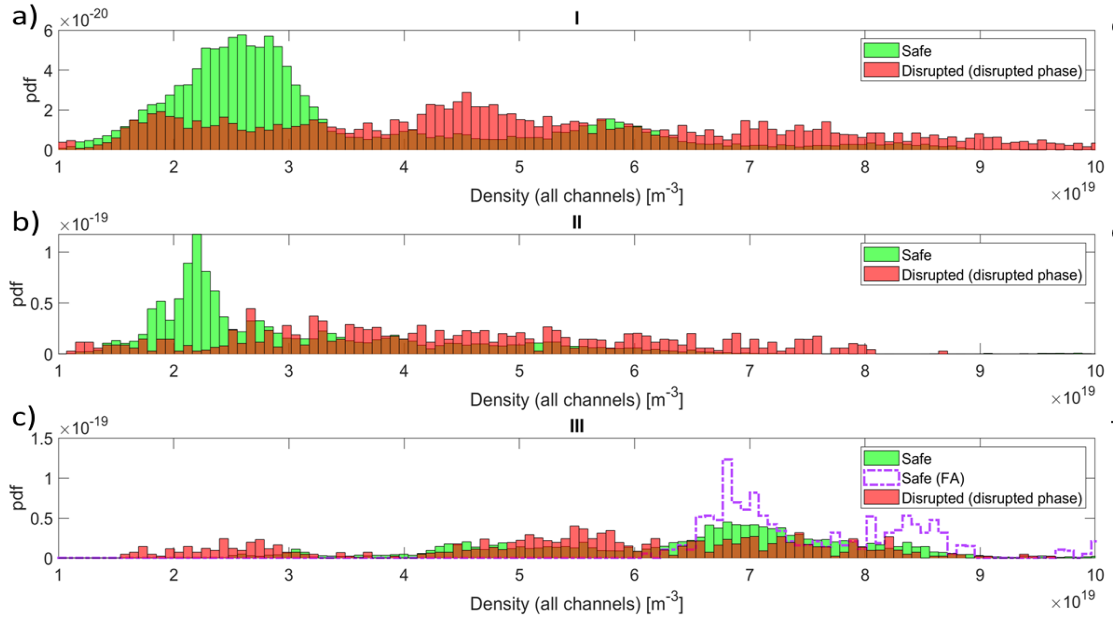

Figure 7.7: a-c) Comparison of the probability density functions of the electron density computed as the mean value across the lines of sight of the HRTS diagnostic for the regularly terminated (green) and the disrupted discharges (red) in the three datasets: a) Dataset I, b) Dataset II, and c) Dataset III. For the Dataset III, the distribution of false alarm values is identified by a magenta dashed line.

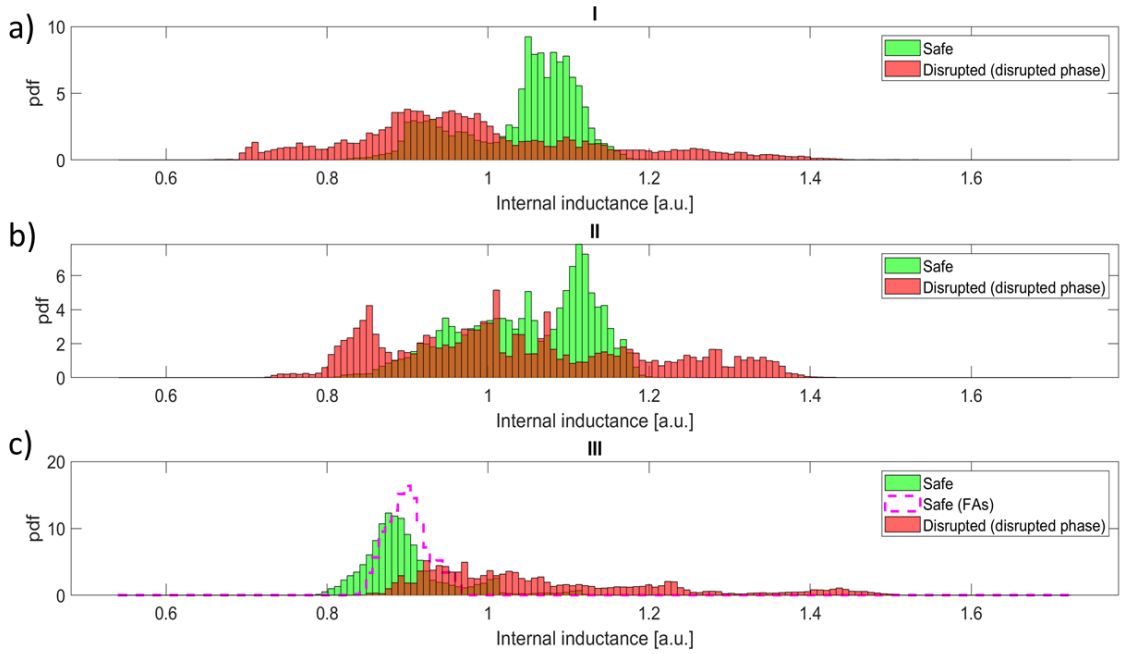

Figure 7.8: a-c) Comparison of the probability density functions of the internal inductance for the regularly terminated (green) and the disrupted discharges (red) in the three datasets: a)

Dataset I, b) Dataset II, and c) Dataset III. For the Dataset III, the distribution of false alarms values is identified by a magenta dashed line.

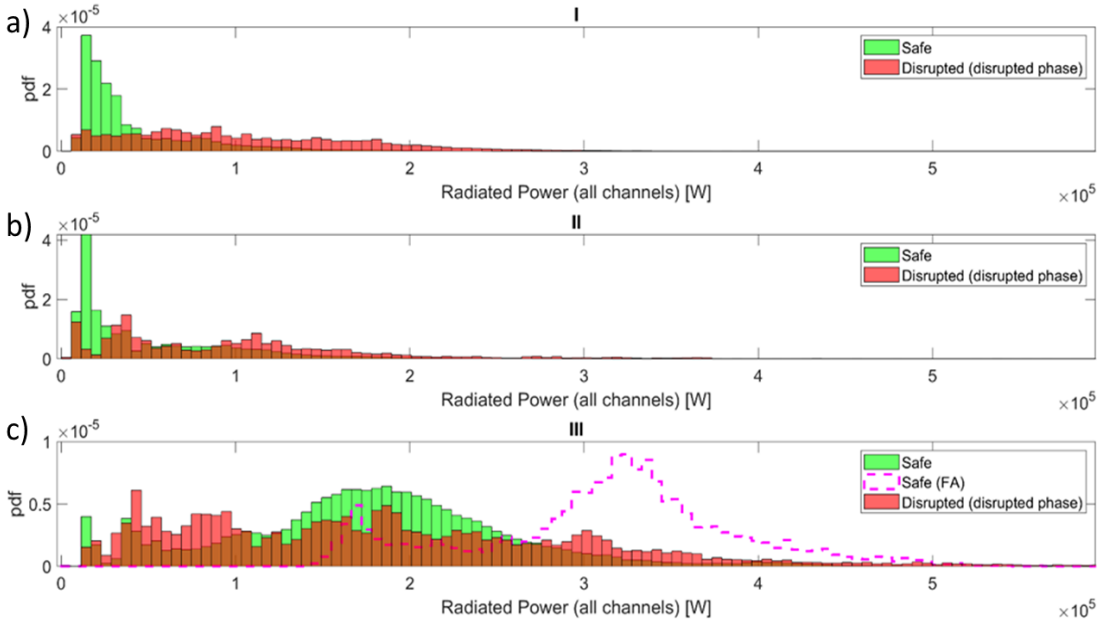

Figure 7.9: Comparison of the probability density functions of the radiated power computed as the mean value across the lines of sight of the HRTS diagnostic for the regularly terminated (green) and the disrupted discharges (red) in the three datasets (from a to c). For Dataset III, the distribution of false alarms values is identified by a magenta dashed line.

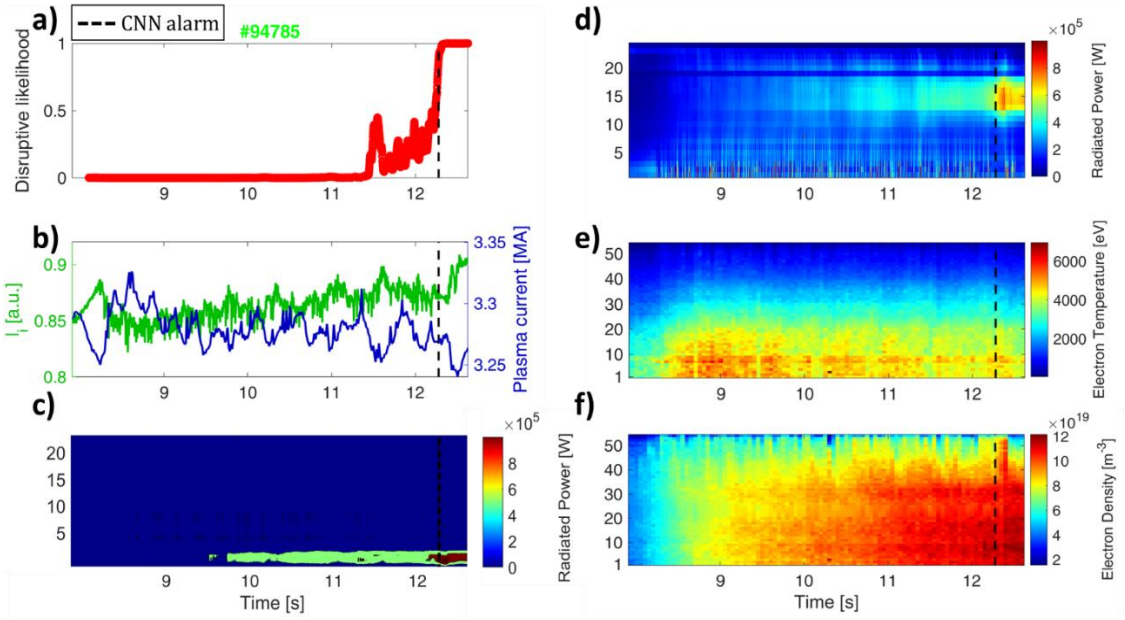

Figure 7.10: CNN output on the regularly terminated discharge #94785. a) CNN disruptive likelihood; b) Internal inductance in green and mode lock normalized by the plasma current in blue; c) Radiated power from the bolometer vertical camera; d) Radiated power from the bolometer horizontal camera; e) Electron temperature from the HRTS; f) Electron Density from the HRTS. The dashed black line indicates the CNN alarm time.

Figure 7.11 reports the response of the CNN to the regularly terminated pulse #95293 from Dataset III. As it can be seen, the CNN triggers a FA nearby a high

radiation from the central lines of sight of the bolometer horizontal camera, together with the decrease of both the electron temperature and the peaking of the electron density at the core. On the other hand, this pattern causes a high number of the false alarms observed in this dataset; in fact, it is very similar to the disruptive mechanism represented in Figure 7.4. It has to be highlighted how, at about 12.5 s, following the temperature and the density flattening over the plasma profile, the CNN predictor reports a gradual reduction of disruption likelihood.

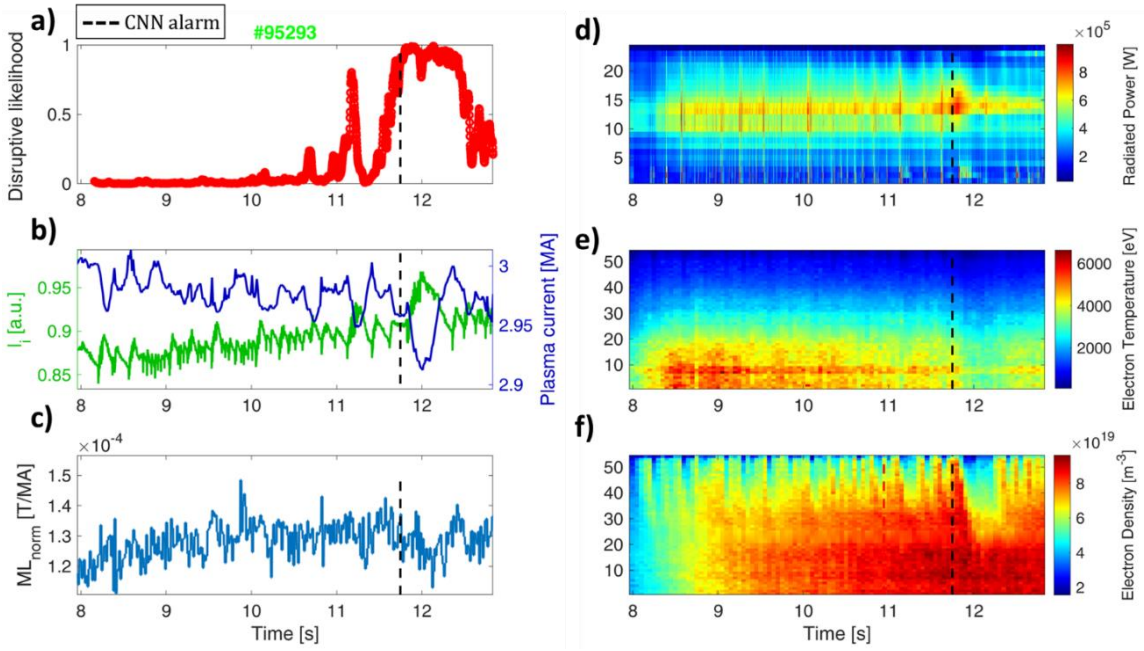

Figure 7.11: CNN output on the regularly terminated discharge #95293. a) CNN disruptive likelihood; b) Internal inductance in green and plasma current in blue; c) mode lock normalized by the plasma current, in blue. d) Radiated power from the bolometer horizontal camera; e) Electron temperature from the HRTS; f) Electron Density from the HRTS. The dashed black line indicates the CNN alarm time.

Instead, Figure 7.12 shows a disrupted pulse belonging to the Dataset III where a disruption due to an edge collapse is detected (#94775). Differently from the impurity accumulation, the edge collapse is characterized by the presence of a blob of radiation in the outer part of the plasma. The radiation causes a localized cooling of the plasma temperature which in turns induces the peaking of the plasma current profiles [137]. This mechanism, visible in Figure 7.12, triggers the alarm at around 9s (black vertical dashed line). In Figure 7.12e it is possible to see the cooling of the plasma between the HRTS lines of sight 12 and 30 (which corresponds to a radial position from 3.13m to 3.46m), together with a high plasma radiation at the central lines of sight of the bolometer horizontal camera (Figure 7.12d) and the rise of the plasma internal inductance (Figure 7.12b). The further analysis of the bolometer vertical camera data allows to localize the radiation blob in the outboard of the plasma (between chords 1-5, see Figure 7.12c). Despite it is not possible to distinguish between the core radiation and the outer low field radiation only from

the bolometer horizontal camera lines of sight, by combining the spatiotemporal information of the HRTS and the bolometer horizontal camera with the *li* signal, the network is able to detect two different "off-normal" patterns: one characterized by a strong radiation due to an impurity accumulation process (see Figure 7.4) and another one where the radiation leads to a cooling at the edge (see Figure 7.12).

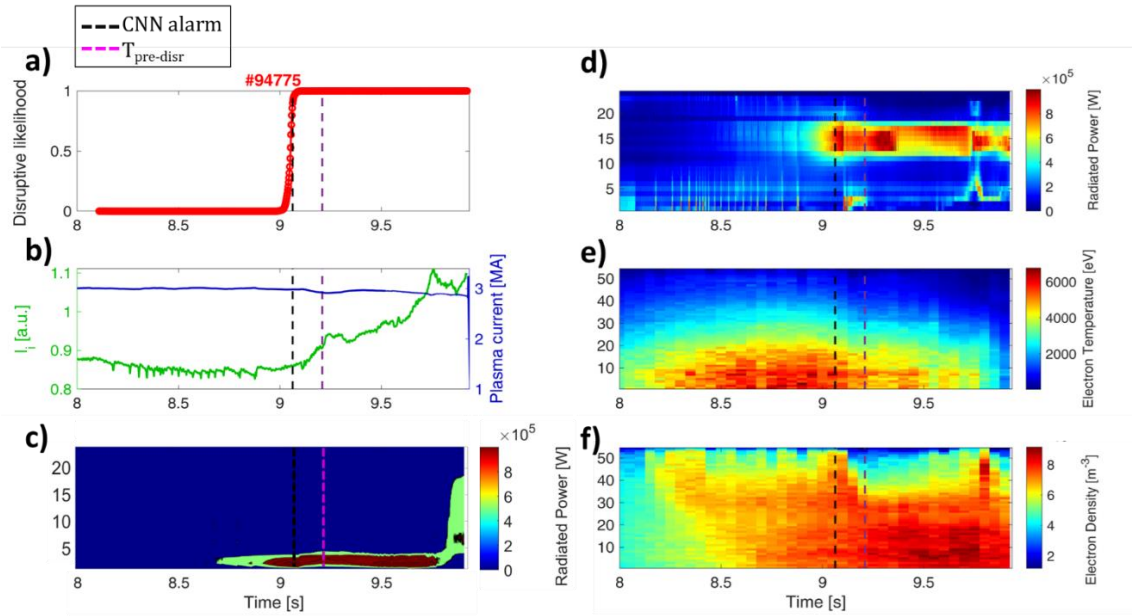

Figure 7.12: JET disrupted discharge #94775. a) CNN disruptive likelihood; b) Internal inductance, in green, and mode lock normalized by the plasma current in blue; c) Radiated power from the bolometer vertical camera; d) Radiated power from the bolometer horizontal camera e) Electron temperature from the HRTS; f) Electron Density from the HRTS. The dashed black line indicates the CNN alarm time, while the dashed magenta line indicates the $t_{pre-disr}$.

7.5 Late fusion architecture and vertical bolometer camera

The input image of the CNN concatenates the 1-D data coming from three diagnostics and the 0-D signals. Note that, in this case, convolutions involving the boundaries are not really senseful and the network discards the corresponding features. On the contrary, the model can learn the associations across the informative regions of the multiple images. Moreover, the idea to train only one CNN, instead of several, allowed a simpler implementation of the model. Following Baltrusaitis et al. [139], the proposed approach can be classified as a joint multimodal representation (also named early fusion as opposed to late fusion). Joint representations combine the unimodal signals into the same representation space. Mathematically, it is expressed as: $x_m = f(x_1, \ldots, x_n)$, where the multimodal representation x_m is computed using the CNN that relies on unimodal representations $x_1, \ldots, x_n$.

Conversely, the late fusion representation processes unimodal signals separately and then the results are merged. Early fusion has the advantage of merging data sources in the beginning of the processing (sometimes after a first

convolution). If the data is properly aligned, cross-correlations between data items may be exploited, thereby providing an opportunity to increase the performance of the system. In [140], the authors argue that those fused low-level features might be irrelevant for the task, thus decreasing the fusion power. When signals from different modalities do not complement each other, i.e., input modalities separately inform the final prediction and do not have any inherent interdependency, then trying an another fusion approach is preferred [141]. Moreover, late fusion retains the ability to make predictions in case of missing or incomplete data, because it employs separate models for separate modalities, and aggregation functions can be applied even when predictions from a modality is missing. The major drawback is the limited potential for the exploitation of cross correlations between the different unimodal data. The optimum fusion strategies for many applications have yet to be determined [142]. In a recent review on fusion techniques for deep learning applications in medicine [141], the authors report that in most applications early fusion is used as the first attempt, a straightforward approach that does not necessarily require training multiple models.

Following to the development of an early fusion disruption predictor, and to the detection of outboard radiation in the most recent non-disrupted deuterium discharges [39], a new late fusion model has been trained to take advantage of the different timescales of disruptive events detected by the 1-D and 0-D data. In fact, the internal inductance and the locked mode signals tend to vary closer to the disruption. Figure 7.13 shows the architecture of the predictor. It consists of two branches, each one being a separate CNN. The top branch, which processes the images of the 1-D profiles, has two convolutional units (CU_1, CU_2) followed by a max pooling layer (P_{max}) and an average pooling layer (Pavg) 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. 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 0-D dimensional data along the horizontal (time) 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.

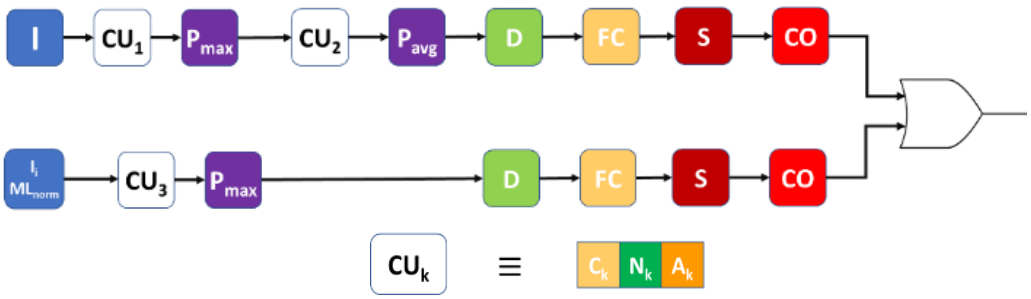

Figure 7.13: CNN architecture, where: I is the image input; CU_k is the kth convolutional unit, composed by the cascade of a convolutional layer (C_k), a batch-normalization layer (N_k) and a nonlinear activation layer with ReLU functions (A_k); P_{max} and P_{avg} are the max-pooling and average-pooling 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 disruptive likelihood of the input segment to belong to a disrupted discharge. As an example, Figure 7.14a 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. Figure 7.14b shows the branch binary outputs, which are obtained by setting to 1 the likelihood 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 7.14b). The logic OR function produces the final disruption trigger.

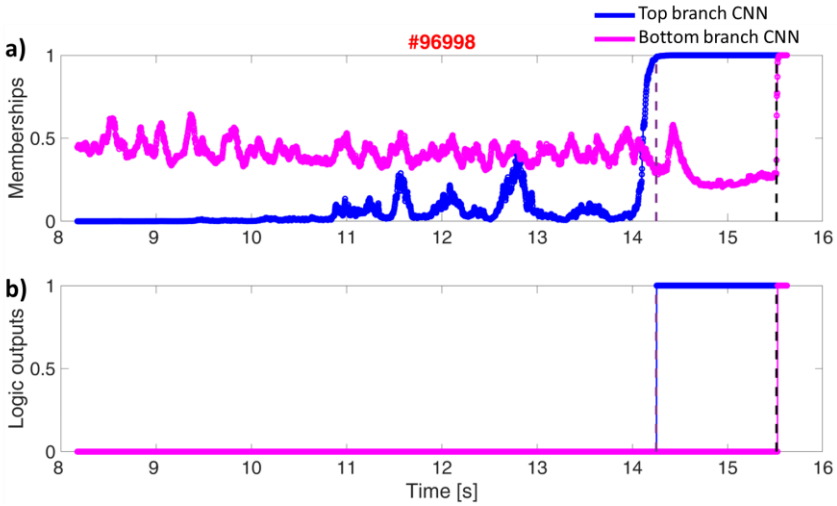

Figure 7.14: Disrupted pulse #96998 a) Disruptive likelihoods for each predictor branch, where the blue line is the top branch one, and the magenta line is the bottom branch membership; b) Logic output for each branch (blue for the top branch, magenta for the bottom one) for the same pulse. The dashed purple line indicates the $t_{pre-disr,AUT}$, the dashed black line indicates the mode-locking time.

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 [137], 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 0-D signals. For the 1-D profile images, the onset of disrupted phase is defined by the $t_{pre-disr}$, whereas, for the 0-D 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, the number of training, validation and test images is indicated in Table 7.1 and is the same as the early fusion model. Instead, for the 0-D 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. 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.

7.6 Performance of the late fusion predictor

The results of the predictor are reported and compared with [143] in Table 7.5. 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 7.5: Predictor performance

Dataset	SP%	MA%	FA%
Train	98.71	0	2.85
Test	98.83	1.87	0.67

Figure 7.15 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 1-D 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 7.16 which refers to the disrupted test pulse #96998 (outside the training range).

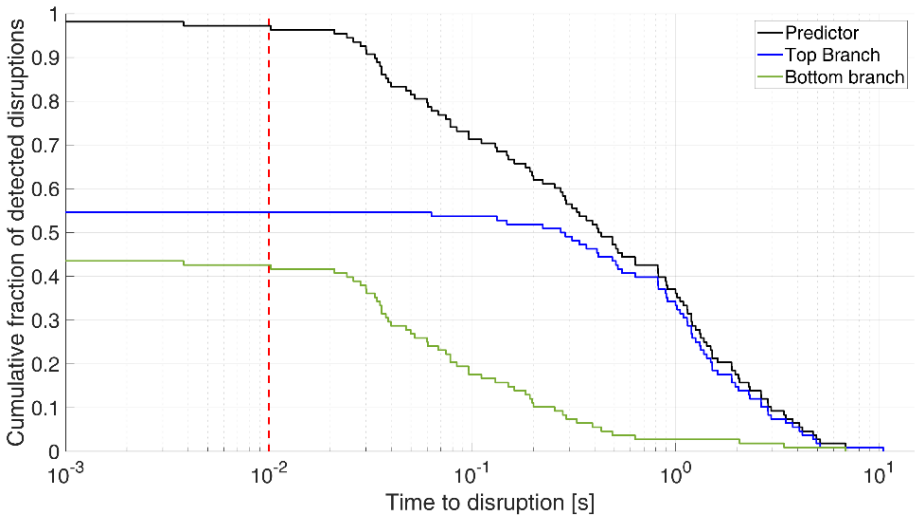

Figure 7.15: CNN model warning time distributions in the test set for the top branch (blue line), the bottom one (green line) and full predictor (black line). Only the first alarm is reported. The vertical red dashed line allows to identify tardive detections.

The top branch of the predictor in Figure 7.16a (blue line) triggers an alarm at around 14.10 s, coherently with the change of the plasma profiles shown in Figure 7.16c-f and in correspondence of the $t_{pre-disr}$, identified by a dashed line. In fact, the electron temperature flattens (Figure 7.16e) and the electron density peaks (Figure 7.16f). This phenomenon is synchronous with strong radiation from the central channels of the horizontal and vertical bolometer (Figure 7.16c-d). On the other hand, the bottom branch of the predictor in Figure 7.16a (magenta line) triggers an alarm at around 15.7s close to the end of the discharge, in correspondence with the rise of l_i and ML_{norm} signals. Hence, the top-branch is trained to detect

destabilizations in the 1-D profiles distributions, while the bottom branch on detecting the onset of a locked-mode and a late disruption pattern. Figure 7.17 shows the regularly terminated pulse #96893, which was detected as disruptive in [143]. 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.

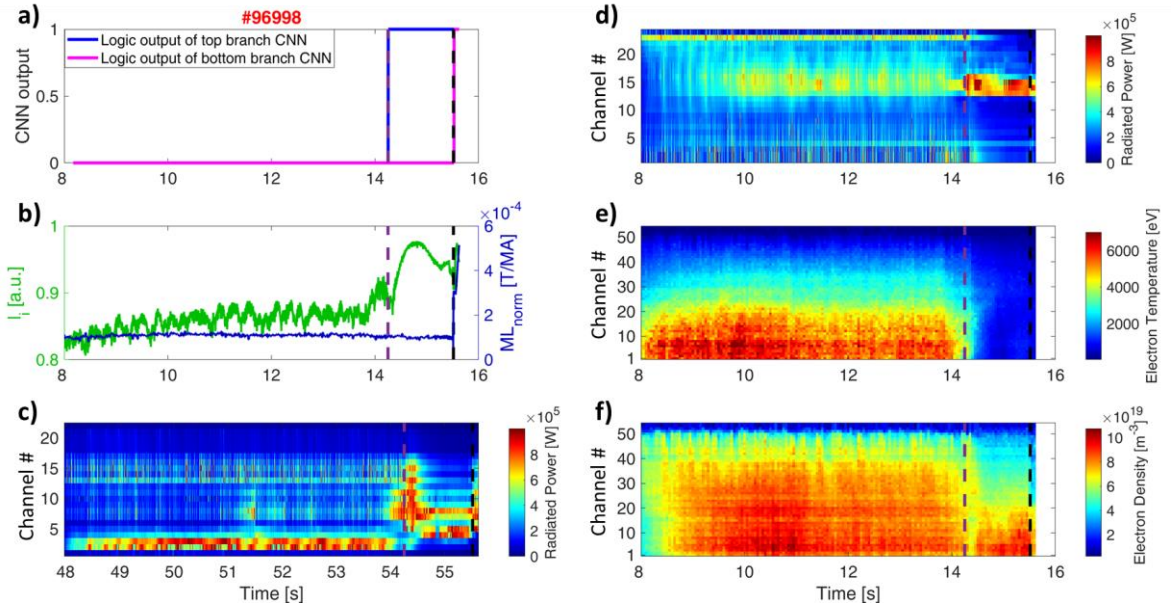

Figure 7.16: JET disrupted discharge #96998. (a) CNN logic output curves, where the blue line is the top branch logic output, and the magenta line is the bottom branch logic output.; (b) internal inductance, in green, and mode lock normalized by the plasma current, in blue; c) radiated power from the 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 $T_{pre-disr}$

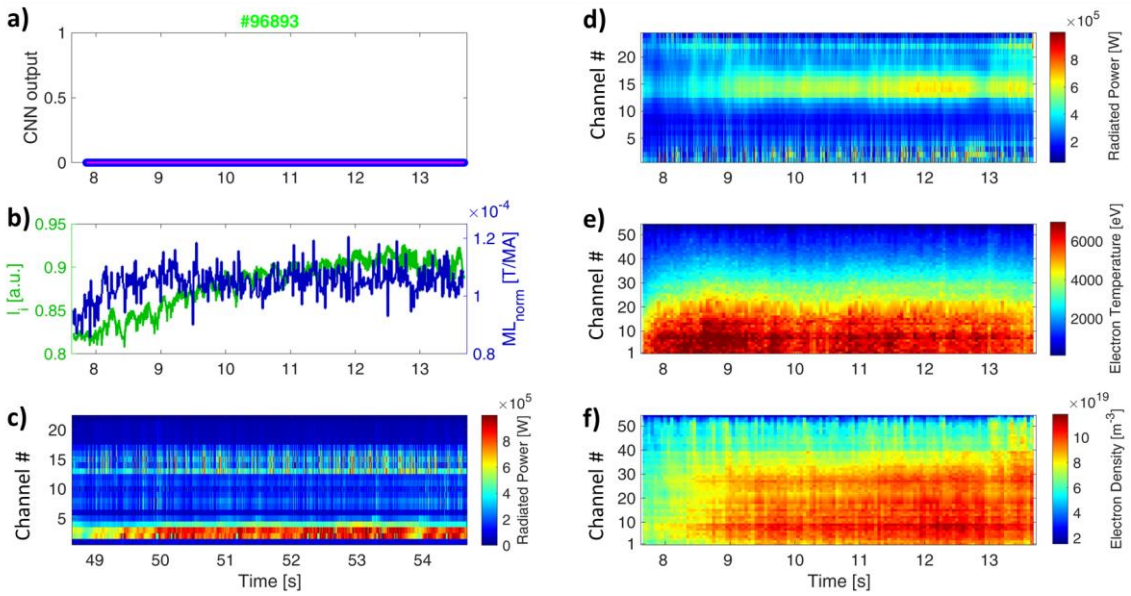

Figure 7.17: JET regularly terminated discharge #96893. (a) CNN logic output curves, where the blue line is the top branch logic output, and the magenta line is the bottom branch logic output.; (b) internal inductance, in green, and mode lock normalized by the plasma current, in blue; c) radiated power 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.

7.7 CNN Architecture with Mirnov signals

To improve the detection of late MHD events, a late fusion CNN architecture has been trained using the information from the Mirnov coil spectrogram. In fact, in [123], the authors have shown that the spectrogram data can be converted in an image and processed with CNNs, providing a disruptive likelihood as an output. For this reason, the spectrogram of the H302 and H305 Mirnov coils at JET were processed to produce input images for the CNN as described in Chapter 4 and starting from the same database. Unfortunately, due to the unavailability of the coil signals for some shots, the database size is changed both on training and test sets. The labelling of the pre-disruptive phase has been made using the automatically computed $t_{pre-disr}$, but in this case, the non-disrupted samples have been retained from both the non-disrupted discharges and the part of the disruptions before $t_{pre-disr}$. This procedure has been carried out after observing that, for the training dataset, the MHD activity in the non-disrupted pulses was usually negligible, while later experiments showed higher level of MHD activity. Finally, for creating the training dataset, one segment every 24 ms has been extracted from the pre-disrupted phase of the disrupted discharges, whereas one segment every 200 ms has been retained from the regularly terminated discharges. During the test instead, the sliding window of 200ms has a stride of 2ms, so that it simulates a real-time implementation of the algorithm. The database adopted for the study is described in Table 7.6.

Table 7.6: number of pulses and time slices in the training, validation and test sets

Set	Disruptions		Regular pulses	
	Pulses	Time slices	Pulses	Time slices
Training	57	3518	50	3690
Validation	18	1147	15	1175
Test	92	226866	131	461595

Then, the CNN architecture in Figure 7.18 has been trained as a disruption predictor. The CNN used in this case is deeper and is based on the use of several squared convolutional blocks, similarly to the one proposed in [123]. The hyperparameters of the training are summarised in Table 7.7. The CNN architecture consists of four convolutional units (CU_1, CU_2, CU_3) followed by a max pooling layer (P_{max}) and a last one (CU_4) followed by an average pooling layer (P_{avg}). The CU_k blocks all have 3x3 filters and they have an increasing number of filters (CU_1 has 4 filters, then CU_2, CU_3, and CU_4 have 8,16 and 32 filters respectively. The P_{max} blocks are 3x3 and reduce the input image size 81 × 101x1 to 1x1x32. The CNN alone adopted as a disruption predictor has the performances shown in Table 7.8. The performance of this model was quite poor, especially in terms of the number of triggered false alarms. Several techniques to process the spectrogram images were tested but the overall results did not improve significantly.

An improvement in the performance of the predictor was achieved by combining the features extracted from this model with the ones extracted by another CNN from 1D profile data contained in the Te, Ne and Horizontal Bolometer camera profiles. For the same training dataset, an architecture with the same structure as the one in Figure 7.1 was trained. Moreover, an alarm scheme with a threshold on the locked mode was adopted, similarly to [15], [115]. This second architecture is described in Figure 7.19, while the alarm scheme is shown in Figure 7.20. The performances of this model are also shown in Table 7.8.

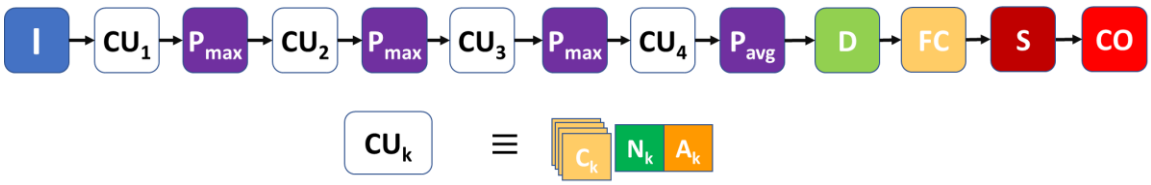

Figure 7.18: CNN architecture, where: I is the image input; CU_k is the kth convolutional unit, composed by the cascade of a convolutional layer (C_k), a batch-normalization layer (N_k) and a nonlinear activation layer with ReLU functions (A_k); P_{max} and P_{avg} are the max-pooling and average-pooling layers, respectively; D is a dropout layer; FC is a fully-connected layer; S and CO are the SoftMax and classification output layers, respectively.

Table 7.7: Training parameters of the CNN model for processing MHD data

Parameters	Values
Optimizer	Stochastic gradient descent with momentum
Initial learning rate	2.5e-4
Learning rate drop factor	0.25
Learning rate drop period (epochs)	20
Momentum	0.9
MiniBatch size	512
Validation frequency (iterations)	50
Validation stop (consecutive evaluations)	100
Weight decay (L2 regularization)	1e-4

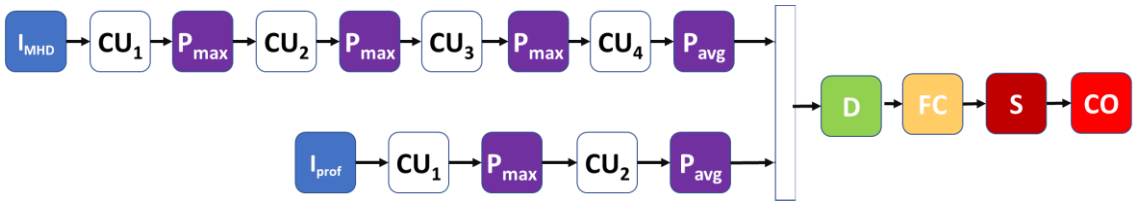

Figure 7.19: CNN architecture, where: I_{MHD} is the image input from the spectrogram of the mirnov coil and I_{prof} is the image input from the 1D plasma profiles; CU_k is the kth convolutional unit, composed by the cascade of a convolutional layer (C_k), a batch-normalization layer (N_k) and a nonlinear 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.

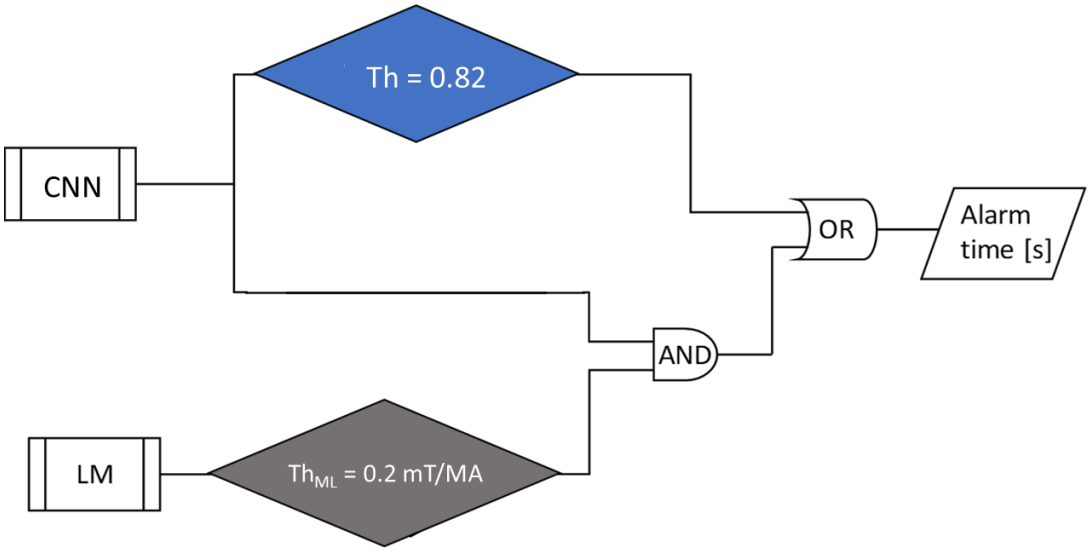

Figure 7.20 Multiple conditions alarm scheme of the CNN disruption predictor

Table 7.8: Test performances of the Mirnov CNN configurations

CNN architecture	TD	MA	FA
$\mathrm{CNN}_{\mathrm{MHD}}$	0%	23.91%	55.73%
$\mathrm{CNN}_{\mathrm{MHD+1Dprofiles}}$ without LM_{norm}	1.09%	34.78%	2.29%
$\mathrm{CNN}_{\mathrm{MHD+1Dprofiles}}$ with LM_{norm}	1.09%	1.09%	2.29%

The investigation on the use of spectrogram data for disruption prediction is still in progress, and more advanced processing schemes could improve the performance of this model.

Chapter 8
Comparison of the models with common metrics

8.1 Common evaluation metrics

As discussed in Chapter 2, a plethora of physics-based and data-driven algorithms have been developed for disruption prediction in tokamaks. However, comparing the different models is not straightforward, due to the lack of common standards:

1) Common sets of input features
2) Common benchmarks and test data sets
3) Common evaluation metrics and definitions

In this thesis, three different machine learning models have been developed for disruption prediction: one based on fully connected neural networks, one based on GTM, and finally a CNN based predictor. These predictors have been developed starting from the set of diagnostics described in Chapter 4, and some specific pre-processing steps are adopted to adapt the input to each model. Hence, the first two conditions for a fair comparison of the approaches are respected. Regarding the metrics, the definitions in the literature differ depending on the research group, and the same term is adopted with different meanings.

The models presented in this thesis all provide a disruptive likelihood as output, such as the one in Figure 8.1. A threshold on the likelihood is applied to binary classify the samples, so that each instance can be assigned to the disrupted or non-disrupted class.

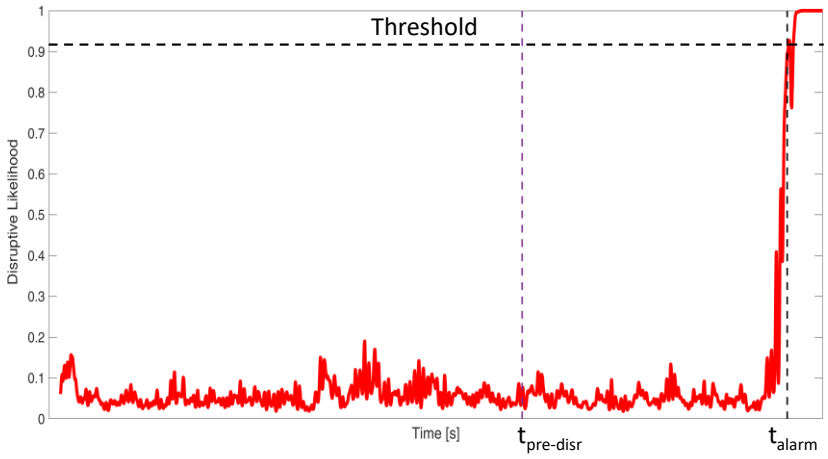

Figure 8.1. Disruptive likelihood evolutions in a disrupted discharge.

The most adopted [15], [72], [76], [79], [81], [82], [85], [92], [93], [99], [144] metrics in the disruption prediction literature are the percentages of successfully predicted discharges (SPs), missed alarms (MAs), tardy detections (TDs), and false alarms (FAs). MAs are the disruptions for which no alarm is triggered before the t_{end}, and the FAs are the non-disrupted pulses where the model raises an alarm. The

definition of TDs is instead machine dependent since it includes the predicted disruptions for which the alarm time is too late to adopt mitigation actions. At JET, the necessary time to activate the mitigation system is 10ms [15]. Successfully predicted discharges are the sum of correctly predicted disruptions and non-disrupted discharges, divided by the total number of pulses. Moreover, since the disruption prediction task can be described as a binary classification one, also the reference metrics from this field are often considered when reporting the results [71], [77], [104], [145], [146]. In this case, disruptive pulses are counted as true positives (TP) if the model raises an alarm before the t_{end}, while it is counted as a false negative (FN) if no alarm or a tardy alarm is triggered (t_{end}-10 ms is still the criterion for defining a tardy alarm). On the other hand, a non-disruptive pulse without an alarm is a true negative (TN) and if the model detects a disruptive behaviour is a false positive (FP). Figure 8.2 summarizes the adopted metrics in this thesis and compares the two definitions. Given the number of TP and TN, the successfully predicted shots can be defined as:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

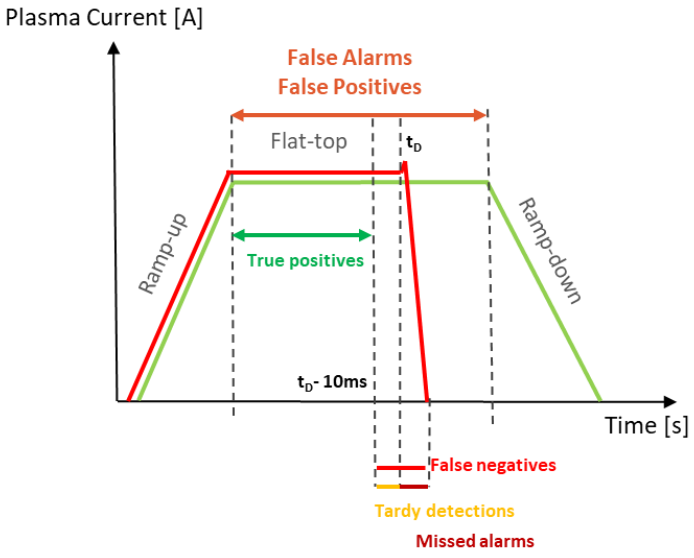

Figure 8.2: A sample non-disrupted discharge (green) and a sample disruption (red) together with the main metrics adopted for evaluating the performances of the disruption predictors: missed alarms, tardy detections and false alarms and compared to true positives, false positives and false negatives.

Moreover, recent disruption prediction systems are being developed especially for avoidance purposes; for a disruption, the goal of an avoidance system is to associate the alarm to the presence of a destabilizing mechanism in the plasma, regardless of the distance of such event to the ending time t_{end}. A well-timed warning time allows the control system to react to the presence of an instability, while with a short warning time the disruption is generally mitigated by the mitigation system. Thus, the premature alarm rate is replaced by the cumulative warning time distribution.

8.2 Common Data Base

The comparison of the three ML DP models has been done referring to the same diagnostic signals in Table 8.1 and the same training, validation and test sets in Table 8.2.

Table 8.1 Diagnostic signals, acronyms and units.

Plasma signal	Acronym	Diagnostics	Dimension
Electron Temperature	T_e	HRTS	1-D
Electron Density	n_e	HRTS	1-D
Radiated Power	P_{rad}	Bolometer	1-D
Total Radiated Power	$P_{rad-TOT}$	Bolometer	0-D
Total Input Power	P_{TOT}	BetaLi	0-D
Internal Inductance	l_i	BetaLi	0-D
Normalized locked mode	LM_{norm}	LMS	0-D

Table 8.2: Training, Validation and Test set discharges

Sets	Disrupted	Non-disrupted	JET campaigns
Training set	63	54	2011-2013
Validation set	22	16	2011-2013
Test set	108	149	2011-2020

8.3 Performance metrics

In binary classification, a true positive (TP) is counted if a positive instance is predicted as positive, whereas it is counted as false negative (FN) if it is predicted as negative. A negative instance predicted as negative is defined as true negative (TN), whereas it is counted as false positive (FP) when predicted as positive. These four values can be summarized in a 2×2 confusion matrix, where each row contains the instances in the actual class whereas each column contains the instances in the predicted class.

Note that, such definitions do not take into account the warning time $\Delta t_{warning}$ provided by the predictor to act on the plasma. However, they can be adapted to the disruption prediction definitions, introduced in 5.5. including tardy detections (TD) and missed alarms (MA) in the counting of FN, and premature detections (PRD) in FP. Thus, a direct correspondence between the two definitions for the performance evaluation can be found, when the instance is the discharge. Note that, TN are evaluated as the difference between negative instances N (number of non-disrupted discharges in the test set) and those counted as FA. The positive instances are indicated as P (number of disrupted discharges in the test set).

Therefore, some performance indices can be used, valid to both the previous definitions:

$$PRECISION = \frac{TP}{TP + FP}$$

$$RECALL = \frac{TP}{TP + FN} = \frac{TP}{P}$$

$$SPECIFICITY = \frac{TN}{TN + FP} = \frac{TN}{N}$$

$$ACCURACY = \frac{TP + TN}{P + N}$$

Using these definitions, the accuracy is equal to the SP metric. In addition, the F-score indicators, that encompasses the information of $PRECISION$ and $RECALL$, can be defined:

$$F_\alpha = (1 + \alpha^2) \cdot \frac{PRECISION \cdot RECALL}{\alpha^2 \cdot PRECISION + RECALL}$$

F_1 score is the harmonic mean between $PRECISION$ and $RECALL$, whereas F_2 assigns a higher cost to the disrupted misclassifications.

For a binary classifier parametrized by a threshold, as our case, the relative trade-off between benefits and costs can be displayed by the receiver operating characteristics (ROC), which draws the true positive rate $TPR = {TP}/{P}$ as a function of the false negative rate $FNR = {FN}/{P}$ by varying the threshold. Moreover, the area under the ROC curve (AUC) can also be used to assess the ability of the model to distinguish between the two classes.

However, as previously cited, in the disruption prediction literature, a most informative figure of merit is defined by the cumulative fraction of detected disruptions as a function of $\Delta t_{warning}$. It allows to read, in a unique graph, besides the successful prediction and the tardy detections, also a general overview of the premature detections and the alarm anticipation times.

All these metrics will be presented in the following to compare the performance of the three DP models.

8.4 FC-NN Results

The training and the tuning of the FC-NN parameters (threshold, Assertion Time) were discussed in Chapter 6 and reported in Table 6.2. As an example, the vertical dashed line in Figure 8.3b) identifies the alarm time t_{alarm}, resulting in a warning time $\Delta t_{warning}$= 408 ms.

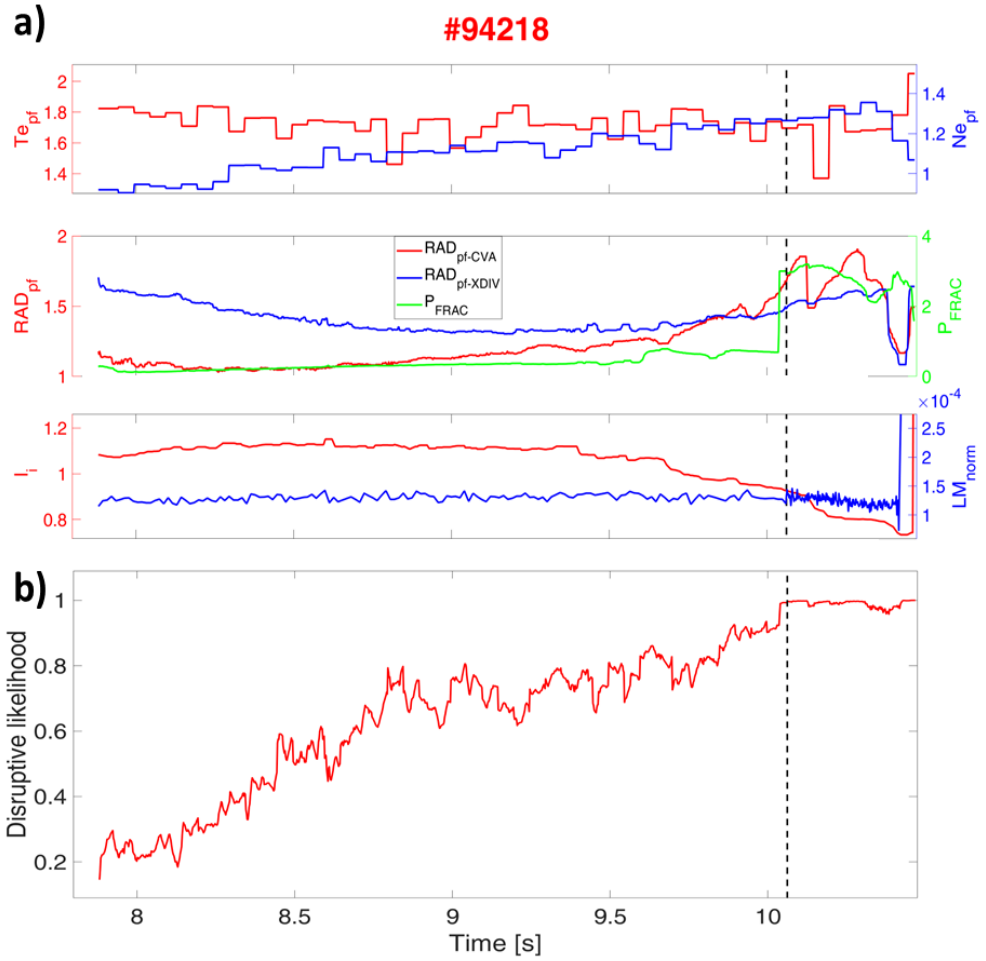

Figure 8.3: JET disrupted discharge #94218: a)Time evolution of the seven plasma dimensionless parameters: temperature (Te_{pf}), plasma density (ne_{pf}), and radiated power (Rad_{pf-CVA}), and $Rad_{pf-XDIV}$ peaking factors, internal inductance l_i, fraction of radiated power P_{frac}, normalized Locked Mode amplitude LM_{norm} signal; b) Disruptive likelihood of the disrupted discharge #94218 supplied by MLP. The dashed black line identifies the alarm time.

In Table 8.3, the confusion matrix of the FC-NN DP model is reported together with the prediction performance indices. All the indices have very good values with an excellent balance between correct predictions of the disrupted pulses and a very limited number of false alarms in regularly terminated pulses. All these numbers overcome the results in literature, e.g., [92], where a MLP was trained with only 0-D signals without the introduction of information, even if synthesized, from plasma profiles. The use of this information really introduces a big benefit on the predictor performance.

Despite these very high-performance index values, and despite the extreme simplicity of the model architecture, the FC-NNs suffer to be 'black boxes' models, which provide a good prediction but are very difficult to interpret. For this reason, other ML predictor architectures have been nominated in recent years to be those selected for future fusion devices.

Table 8.3 Confusion matrix and performance indices of the FC-NN prediction model evaluated on the test set.

		Predicted	
	P+N=257	P=108	N=149
Actual	P=108	TP=103	FN=5
	N=149	FP=5	TN=144
	PRECISION=0.954	RECALL=0.954	SPECIFICITY=0.966
	ACCURACY=0.961	F_1=0.954	F_2=0.954
Train	SP%=99.15%	MA%=0%	FA%=1.85%
Test	SP%=96.11%	MA%=2.78%	FA%=3.36%

8.5 GTM disruption prediction model results

All the synthesized features except LM_{norm} used to train the FC-NN model have been used also to train the GTM model. The free parameters of the GTM model, reported in Table 8.4, have been optimized with a Tabu Search procedure [147]. In Table 8.4, also the resulting GTM map composition is reported. The obtained GTM map of the JET operational space is reported in Figure 8.4a), where same disrupted pulse #94218 reported in the previous section is tracked. The trajectory of the discharge firstly evolves within the green "safe" region and then enter in the red disruptive region. The lighter points of the trajectory correspond to the beginning of the discharge, whereas the darker one corresponds to the end, at the disruption time t_D. The corresponding disruptive likelihood is reported in Figure 8.4b). The vertical dashed line identifies the alarm time t_{alarm}.

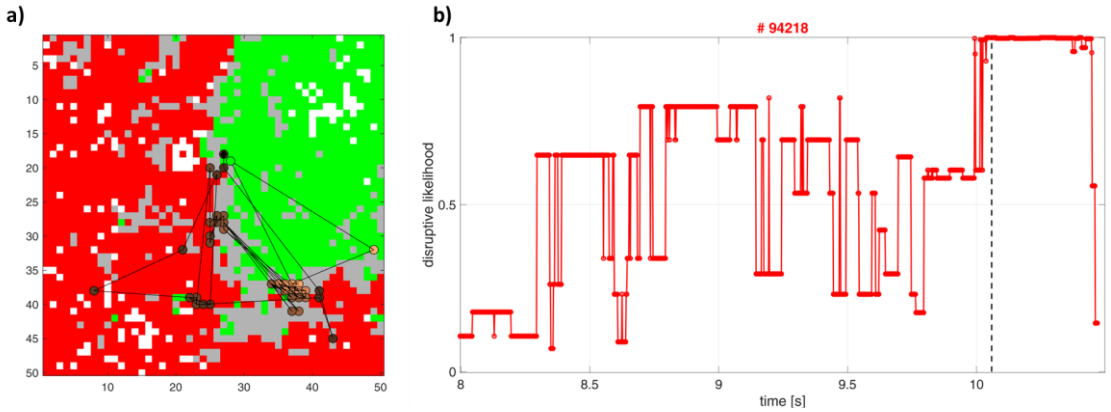

Figure 8.4: a) GTM map of JET operational space with trajectory of the disrupted discharge #94218; b) Disruptive likelihood of the disrupted discharge #94218. The dashed black line identifies the alarm time.

The disruptive likelihood has usually a discontinuous trend with numerous peaks that could trigger incorrect alarms if an adequate threshold and assertion time were not optimized. Moreover, the normalized locked mode signal, not used to train the GTM model, is used in the multiple condition alarm scheme shown in Figure 8.5, as proposed in [15]. Note that, the number d dynamically varies during the discharge. As the sampling time is assumed equal to 2 ms, a mean assertion time of

10 samples can be easily derived from d. For the disrupted discharge #94218 in Figure 8.4, the GTM correctly predicts the disruption with a resulting warning time $\Delta t_{warning}$= 410ms.

Table 8.4 GTM training parameters

Parameters	Value
Optimizer	Expectation Maximization
Map dimension	50x50 grid
Type of RBF	Radially symmetric
Number of RBF	400
Width σ of the RBF	0.8
Alarm threshold	100%
Log Likelihood	9.85×10^5
Disrupted units	50.48%
Non-disrupted units	28.16%
Mixed units	18.52%
Empty clusters	5.48%

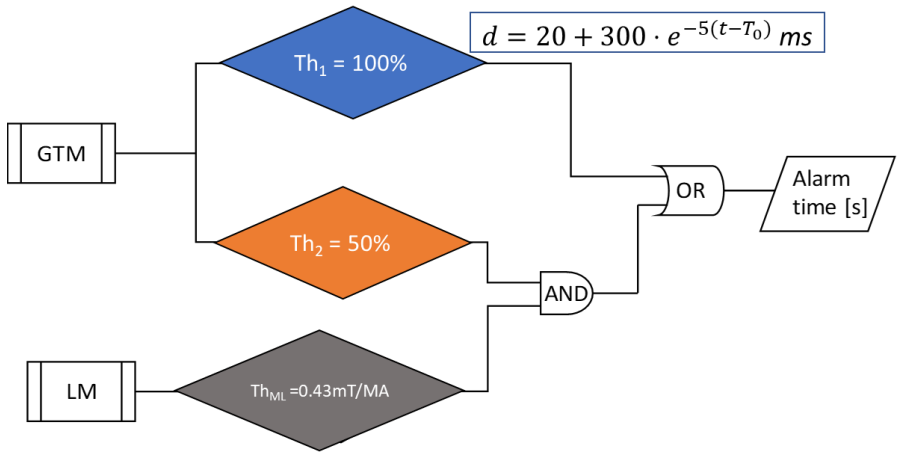

Figure 8.5: Multiple conditions alarm scheme of the GTM disruption predictor (T_0 is the starting time of the flat-top).

Table 8.5 reports the confusion matrix and the values of the same prediction performance indices reported in Table 8.4. The Recall is very high, which means a very high percentage of successful disruption predictions (97.22% in the test), but the specificity degrades compared to MLP due to the greater number of false alarms.

Despite this lower performance, the GTM model has had a considerable appreciation for its remarkable capabilities of visualizing the plasma operational space and the trajectories of the discharges on the map. This allows one to perform disruption prevention actions by monitoring the proximity of the discharge to the safe operational boundary.

Table 8.5: Confusion matrix and performance indices of the GTM prediction model evaluated on the test set.

	Predicted

	P+N=257	P=133	N=124
Actual	P=108	TP=105	FN=3
	N=149	FP=28	TN=121
	PRECISION=0.789	RECALL=0.972	SPECIFICITY=0.812
	ACCURACY=0.879	F_1=0.871	F_2=0.929
Train	SP%=100%	MA%=0%	FA%=0%
Test	SP%=87.9%	MA%=1.85%	FA%=18.79%

8.6 CNN disruption prediction model

The architecture of the CNN disruption predictor selected for the comparison is reported in Figure 8.6. Due to the ability of the CNN to process images, the plasma profiles, which are 1-D signals, have been treated as a single Image, as previously described. The other 0-D signals are fed in the CNN downstream of the first filter block and after this block has been trained and frozen. The optimized free parameters of the training process are reported inTable 8.6. Figure 8.7a) reports the Image of the plasma profiles of the disrupted discharge #94218. By feeding the CNN with a sliding window of 200 ms on the test discharge, the corresponding disruptive likelihood outcomes, as reported in Figure 8.7b). The vertical dashed line identifies the alarm time t_{alarm}. Also, the CNN is able to correctly predict the disruption with a warning time $\Delta t_{warning}$= 372ms.

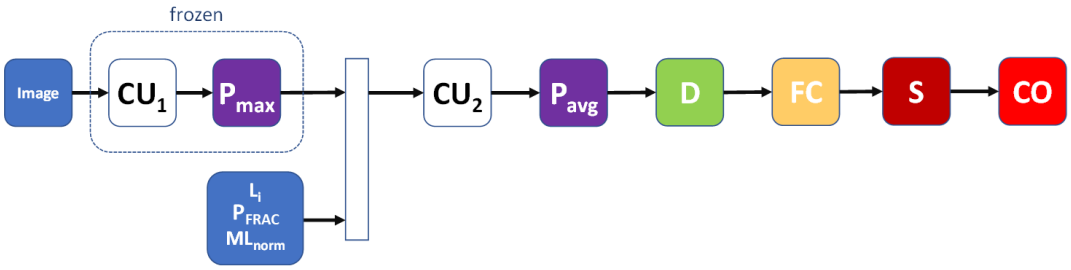

Figure 8.6: Architecture of the CNN disruption predictor.

Table 8.6: CNN training parameters

Parameters	Value
Optimizer	Stochastic gradient descent with momentum
Initial learning rate	$1x10^{-4}$
Learning rate drop factor	0.1
Learning rate drop period (epochs)	20
Momentum	0.9
MiniBatch size	16
Validation frequency (iterations)	50
Validation stop (consecutive evaluations)	100
Weight decay (L2 regularization)	$1x10^{-4}$
Assertion time (samples)	0

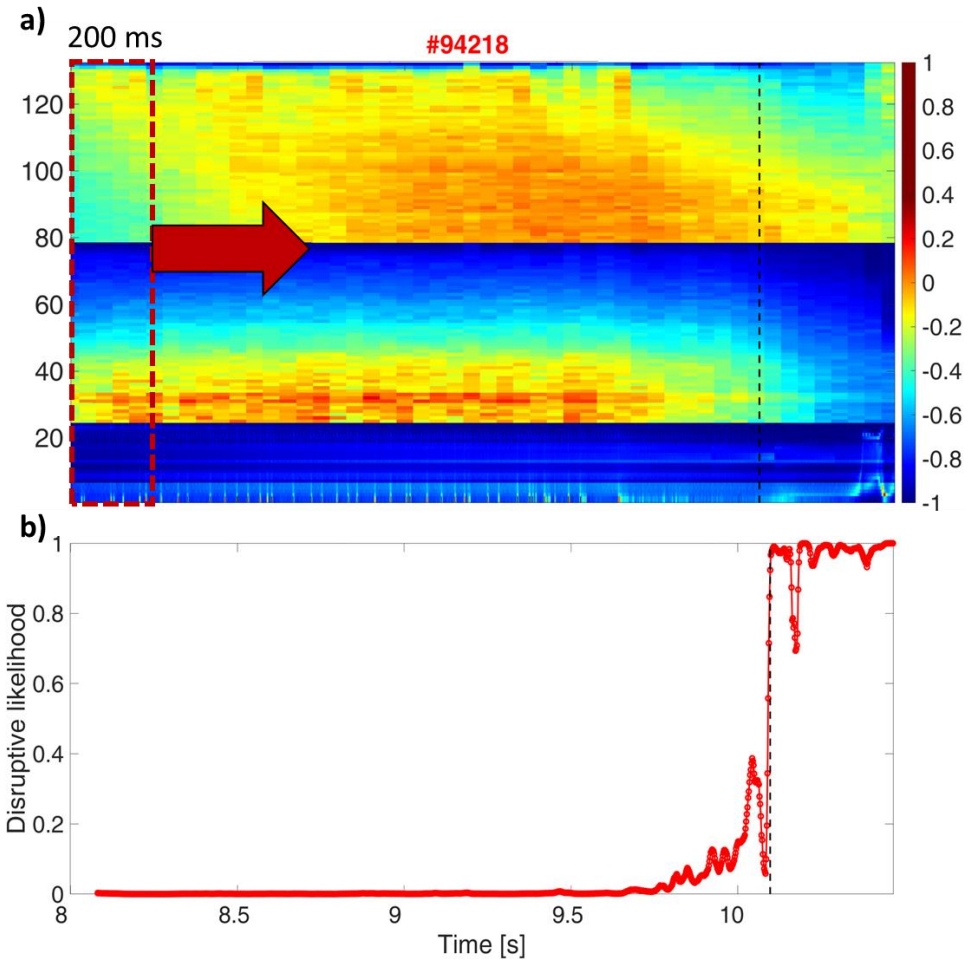

Figure 8.7: a) Input image of the disrupted discharge #94218; b) Disruptive likelihood of the disrupted discharge #94218. The dashed black line identifies the alarm time.

Table 8.7 reports the confusion matrix and the values of the prediction performance indices of the CNN model. As in the case of the MLP model, all the indices have high values getting a tradeoff between successful predictions and false alarms. Note that, the plasma profiles have been directly used to feed the CNN model without the feature engineering processing implemented for the MLP and GTM.

Table 8.7: Confusion matrix and performance indices of the CNN prediction model evaluated on the test set.

		Predicted	
	P+N=257	P=110	N=147
Actual	P=108	TP=102	FN=6
	N=149	FP=8	TN=141
	PRECISION=0.927	RECALL=0.944	SPECIFICITY=0.946
	ACCURACY=0.946	F_1=0.936	F_2=0.941
Train	SP%=99.15%	MA%=0%	FA%=1.852%
Test	SP%=94.6%	MA%=2.778	FA%=5.369%

8.7 Discussion and Conclusions

An indicator of performance of more immediate reading in the prediction of disruptions is the accumulated fraction of detected disruption as a function of the warning time $\Delta t_{warning}$. It provides, for each value of the desired warning time (in x-axis), the percentage (or per-unit) of predicted disruptions, and allows to read, in a unique graph, besides the predicted disruptions and the tardy detections, also a general overview of the premature detections and the alarm anticipation times. This is also a powerful means for comparing different models. Figure 8.8 reports such comparison for the three proposed ML DP models. It is possible to see that the GTM (red line) has the earliest warning times, and its cumulative distribution of alarms is often to the right of the $t_{pre-disr,AUT}$ one. These early alarms can be associated to the high number of false alarms of the GTM, which has a less smooth disruptive likelihood and needs an assertion time to trigger the alarm. Then, the CNN and MLP have similar cumulative distributions, with the MLP which triggers one alarm more just before the red dashed vertical line, and the CNN which triggers some tardy alarms after it. The CNN follows the $t_{pre-disr,AUT}$ distribution until 300 before the disruption, while the MLP tends to be stay to the left of the $t_{pre-disr,AUT}$ curve.

Despite the better results in terms of performances, the MLP and the CNN are mostly employed as black box algorithms and do not allow to extract significant information on the disruption type and possible recovery strategies, while the GTM allows to track the position of the discharge and to associate the instability mechanism with the position of the point in the map. Among the MLP and the CNN, the latter provides an overall higher number of alarms and a generally higher warning time keeping a low number of false alarms.

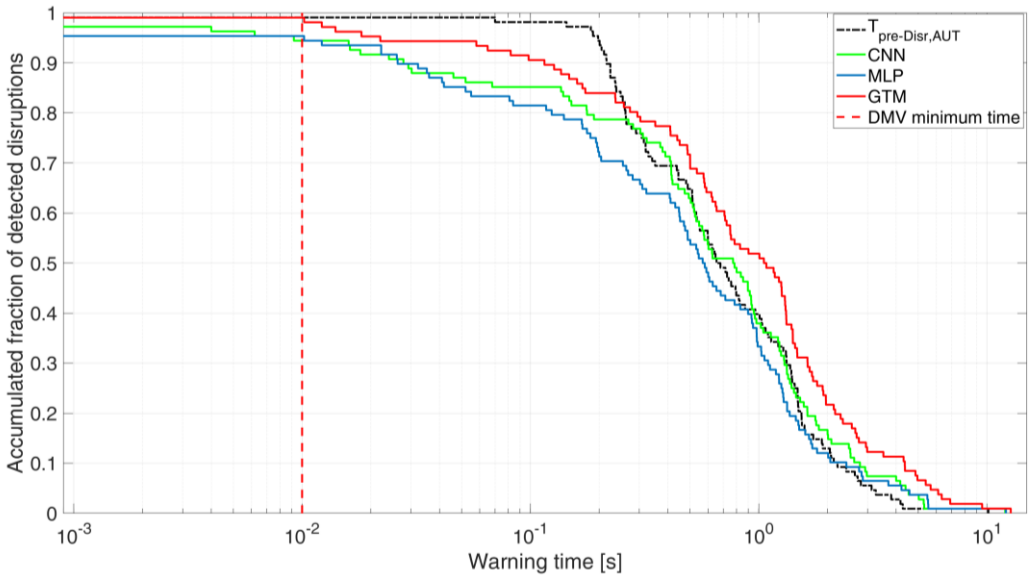

Figure 8.8. Accumulated fraction of detected disruptions by the MLP-NN (blue line), the GTM (red line) and the CNN (green line) models versus the warning time in the test set. The vertical red dashed line allows us to identify tardy detections.

Figure 8.9 reports, for the three predictors, the ROC curve. It is possible to see how the CNN and the MLP have the best compromise between detection of disruptions and number of false positives (false alarms), as also visible in the Area Under the Curve (AUC) reported in Table 8.8. Moreover, looking at the points of the ROC, it is possible to verify that the CNN performance on the test is slightly more robust than the MLP one. In fact, both models have an optimal threshold above 0.9, but the CNN has an overall accuracy of 89% even with lower thresholds, up to 0.7. It is possible to confirm this remark also by comparing the three disruptive likelihoods in Figure 8.3b, Figure 8.4b and Figure 8.7b. The CNN has a lower disruptive likelihood in the stable phase of the disruption, and then rises abruptly in the last part of the discharge, where it is possible to see also a clear variation in the images of the profiles.

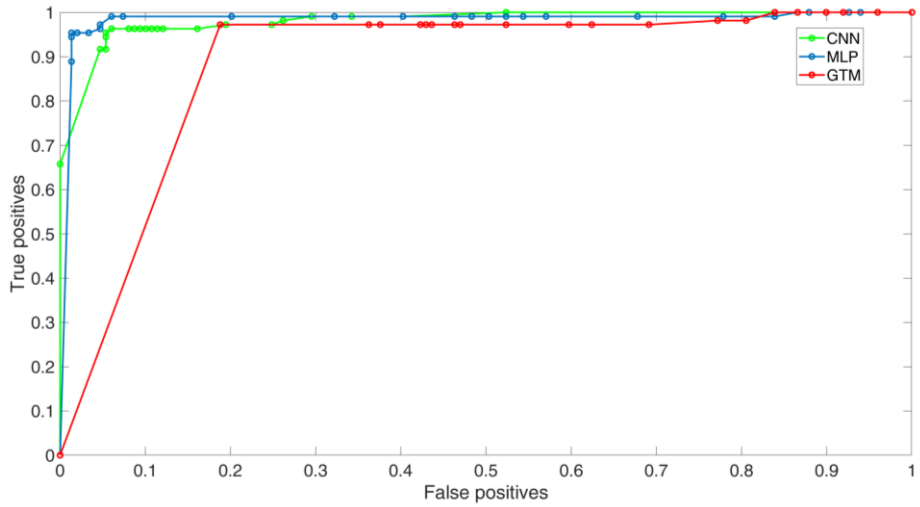

Figure 8.9: Receiver Operating Curve (ROC) of the MLP-NN (blue line), the GTM (red line) and the CNN (green line) models.

Table 8.8 AUC and Assertion Time for MLP-NN, GTM, and CNN models.

DP model	AUC	Assertion Time (# of samples)
MLP-NN	0.98	0
GTM	0.89	10
CNN	0.98	0

In the recent years, a plethora of different machine learning models, metrics and features has been proposed for the development of disruption predictors. This section aims to provide a systematic comparison of some of the most adopted models and to select common metrics for the results evaluation. Using the same training and test set, an MLP, a GTM and a CNN have been trained as disruption prediction starting from the same set of diagnostics: the electron temperature, density and radiation profiles, the locked mode signal, the radiated fraction and the internal inductance. The GTM and the MLP have been trained using a set of processed signals developed from the plasma profiles, the peaking factors, while the CNN is able to directly process the spatiotemporal images of the diagnostics. All the evaluated methods demonstrated the capability of producing early warning times and, in the case of the MLP and of the CNN, with a reduced number of false alarms. Despite the GTM performances being a bit below the other two, its advantage is the interpretability of the model and the possibility to quantify the distance of the tracked discharge from the non-disrupted area of the map. On the other hand, the CNN has the advantage of being able to process the input images without the need of manually extracting physics-based feature from them, due its capability to process image data. The lower interpretability of neural network models could be addressed by exploiting analysis algorithms such as Class Activation Mapping [123] and by developing predictors which identify specific events.

Nevertheless, the use of appropriate diagnostic signals, of a physics-based feature extraction and of automatic training times specific for each disruption allowed to train the models on a reduced number of discharges, to enable the detection of destabilization with larger warning times and to maintain the performance on more recent discharges up to the 2020 campaign.

Several metrics were adopted in evaluating the predictors, from the confusion matrix to the typical metrics adopted in the machine learning community, such as recall, precision. etc. However, among all the proposed methods, the accumulated fraction of detected disruptions against the warning time, together with the respective false alarms rate allowed to provide a clear and immediate overview of the performances of the model, confirming the use of these metrics in the evaluation of the predictors.

Part 2 Conclusions

In this Part 2, disruption prediction algorithms based on the GTM (Generative Topographic Mapping), FC-NN (Fully Connected Neural Network) and CNN (Convolutional Neural Network) have been introduced, as well as an algorithm for automatically identifying the pre-disruptive phase of tokamak discharges. This work is framed in the complex and broad field of disruption prediction and classification; the field addresses the issues related to the integrity preservation of the tokamaks and to the better understanding of the physical mechanisms which destabilize the plasma. Researchers are interested not only in the classification task, but also in the properties of the parameter space where relevant disruption physics occurs, in the visualization and interpretation of this data. The use of automatic pre-disruptive times as inputs to the three models, rather than manually detected times, resulted in encouraging results.

A performance indicator for predicting disruptions is the accumulated fraction of detected disruptions as a function of the warning time ($\Delta t_{warning}$). This provides, for each value of the desired warning time (on the x-axis), the percentage of successful predictions and allows for a general overview of successful predictions, tardy detections, premature detections, and alarm anticipation times. Figure 8.8 compares the performance of the three proposed ML DP models using this indicator. The GTM (red line) has the earliest warning times, but also has a high number of false alarms, as its disruptive likelihood is not smooth and requires an assertion time to trigger the alarm. The CNN and MLP have similar cumulative distributions, with the MLP triggering one more alarm just before the red dashed vertical line and the CNN triggering some tardy alarms after it. The CNN follows the $t_{pre-disr,AUT}$ distribution until 300 before the disruption, while the MLP tends to stay to the left of the $t_{pre-disr,AUT}$ curve. While the MLP and CNN have better performance, they are mostly used as black box algorithms and do not provide significant information on disruption type or potential recovery strategies. In contrast, the GTM allows for tracking the position of the discharge and for associating instability mechanisms with the position on the map. Of the MLP and CNN, the CNN has a higher overall number of alarms and generally longer warning times, while maintaining a low number of false alarms.

Part 3: Heat-flux computation at W7-X

Chapter 9
First wall monitoring and state of the art

9.1 Overview of the wall protection activities

The monitoring of the first wall components is a very important task in view of ITER and next generation power plants, since ITER will have up to 850 MW of power transferred to the cooling system through the wall [148]. Infrared cameras are used in stellarator and tokamaks to monitor the power loads in the first wall [149]–[152]. Infrared thermography is based on the conversion of the light emitted by the observed surface into temperature, according to Planck's law. The light is typically in the wavelength ranges of 3-5 μm or 8-14 μm. In the nuclear fusion devices, the temperature measurements are used to spot and classify thermal events with manual and automatic approaches [150], [151], [153]–[155]. Nevertheless, during steady state operation, the temperature of the wall will progressively increase, making the detection of hot spots and events from the temperature alone complex. In this regard, the heat fluxes provide an estimate of the available time before overloading the component, and the material limits are expressed in terms of maximum tolerable steady state heat fluxes. For this reason, the real-time estimation of heat fluxes is a pivotal activity in view of the future Wendelstein 7-X experiments, which aim to demonstrate the feasibility of a steady state operation of the device.

In the coming experimental campaigns, W7-X will sustain the plasmas for up to 30 minutes [156]. However, the first wall of the machine is exposed to elevated temperature and heat loads. In particular, the divertor tiles are subject to the risk of erosion and melting if localised heat loads overcome the material limits. For this reason, to prevent damages to the wall tiles while keeping high performances during the discharges, a real-time monitoring and control of the heat loads in the first wall is necessary.

9.2 First wall of Wendelstein 7-X

In W7-X, the different parts which constitute the first wall are jointly called Plasma Facing Components (PFCs) [157]. Each of them has a specific function:

- Divertor Targets: this part must sustain the highest heat loads since it is where the last closed flux surface of the plasma is formed. There are 10 island divertors, each one with a Horizontal target and a Vertical target.
  The Horizontal target is constituted by the Low, Middle and the High Iota part. The Low Iota part is made by four CFC modules, TM1H to TM4H, the Middle part has two low load modules in fine graphite, TM5H and TM6H, and the High Iota part includes CFC modules TM7H-TM9H. The vertical part is instead made by three modules, TM1V-TM3V [21]. Figure 9.1 shows the different modules and the OP2 divertor geometry.

- Inner Wall Shield parts of the main wall. They are made of CuCrZr cooling structures, to which graphite tiles are clamped [158].
- Outer Wall Panels: The wall panels are made of two stainless steel sheets welded together to enable the heat transfer [158].
- Baffles: the baffles drive the neutral flux particles into the pumping gap. They are made of a similar technology to the wall shield, with CFC tiles fixed to a copper cooling structure.
- Pumping gap: this part removes impurities in the plasma from the device. It is connected to the Cryo-Vacuum Pump (CVP) which causes a pressure gradient and captures the exhaust particles. It is made in steel.
- Closures: the closures in the poloidal and toroidal directions enclose the divertor island, to achieve an increased pressure in this region and improve the CVP efficiency. It has a similar composition as the wall shield and the baffles.

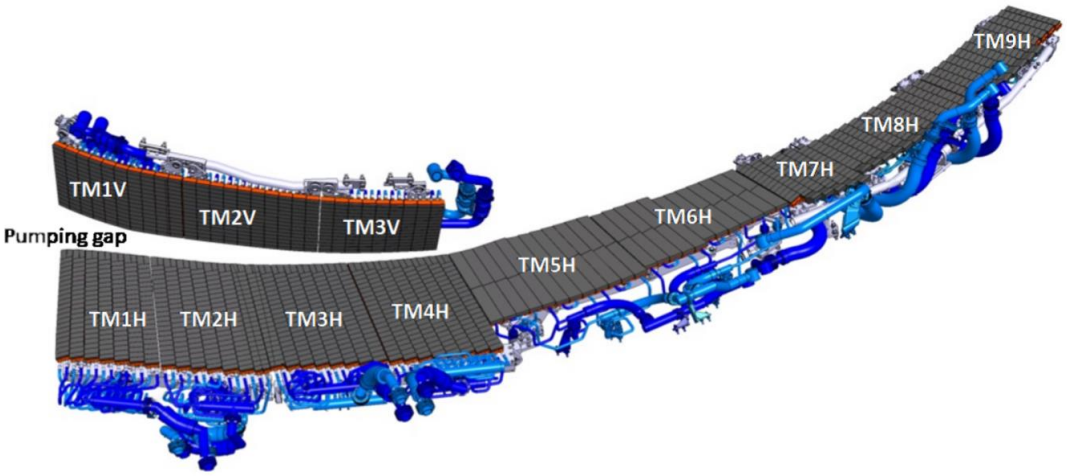

Figure 9.1: The divertor units are 5 m long and 1 m wide with several target modules: vertical targets (TM1V-TM3V), low-iota targets (TM1H-TM4H), low-load target modules (TM5H-TM6H) and high-iota targets (TM7H-TM9H). The divertor water-cooled tiles are made of CFC (Carbon Fibre Composite) except the two low-load central target modules, which are made of fine-grain graphite. [21]

Figure 9.2 shows an example of an image where the measurement of an infrared camera is overlaid to the CAD model of the first wall and the different parts are highlighted. It is also possible to see how the divertor (horizontal and vertical targets in particular) has the highest temperature, due to the close interaction with the plasma.

The materials of these parts are different, depending on the temperatures that they must withstand. For instance, the divertor tiles are made of a Carbon Fibre Composite (CFC) layer joined to a CuCrZr heat sink structure. The Cu interlayer, which should not exceed a sustained temperature of 475 °C, limits the maximum temperature of the tiles. This temperature is reached at 10 MW/m^2 when the surface temperature is 1200 °C [158], [159]. Figure 9.3 shows the maximum tolerable

temperature for each PFC. The different materials require an accurate reconstruction of the position of each thermal event.

In OP 1.2, nine immersion tubes and 1 endoscope were used to monitor the island divertors [152]. Both the endoscope and the immersion tube are equipped with a visible detector to analyse the plasma behaviour and identify events in the visible spectrum, and an infrared detector to measure the temperature of the first wall components.

At the beginning of OP2, the diagnostic comprises 8 water-cooled immersion tubes and 2 steady-state endoscopes. Moreover, two additional high-resolution cameras monitor the two vertical targets observed by the endoscopes, due to the reduced field of view of the latter. The immersion tubes, however, are not suitable for the steady-state operation and the endoscopes equipped for steady-state operation, will replace them in the later phase of OP2, when the plasma energy will progressively increase up to 18 GJ.[21]

A scene model has been built to provide the mapping between the field of view of the cameras and the Computer Aided Design (CAD) geometry of the PFCs [152], [157]. The mapping can be built automatically after an automatic spatial calibration procedure, providing a camera model which takes into account the lens distortion parameters, and provides, for each camera, pixelwise information regarding: the observed PFC; the distance of the target material from the camera eye and the angle of the line of sight with respect to the surface normal; the 3-D coordinates of the observed target.

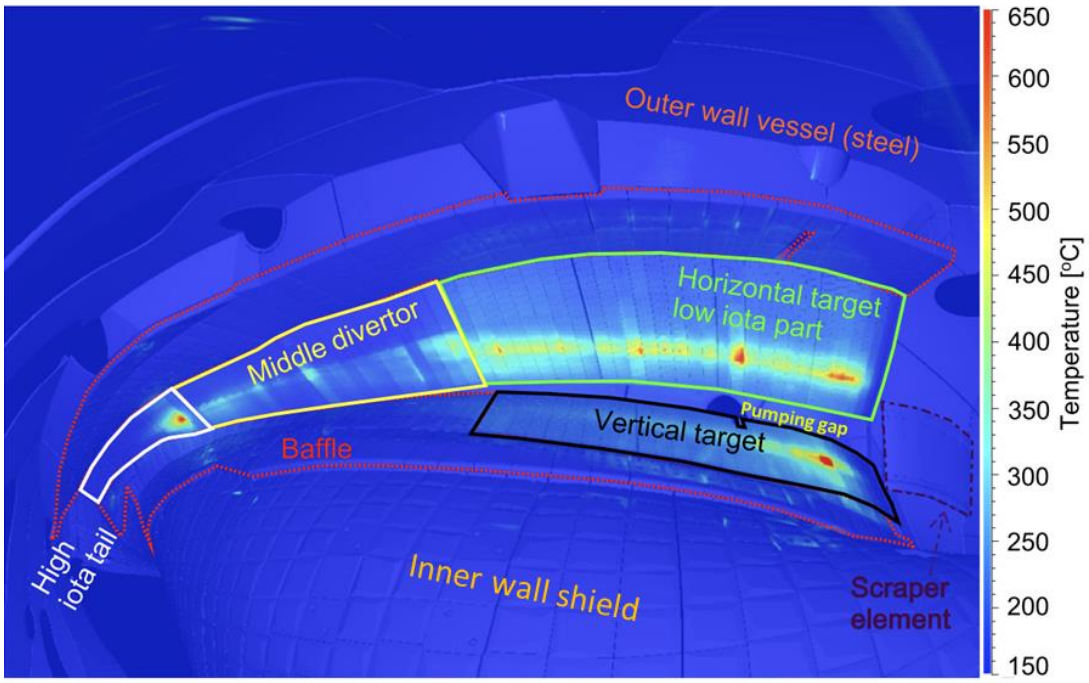

Figure 9.2 Temperature measurements from the IR camera overlaid to the CAD model of the island divertor. Image adapted from [23]

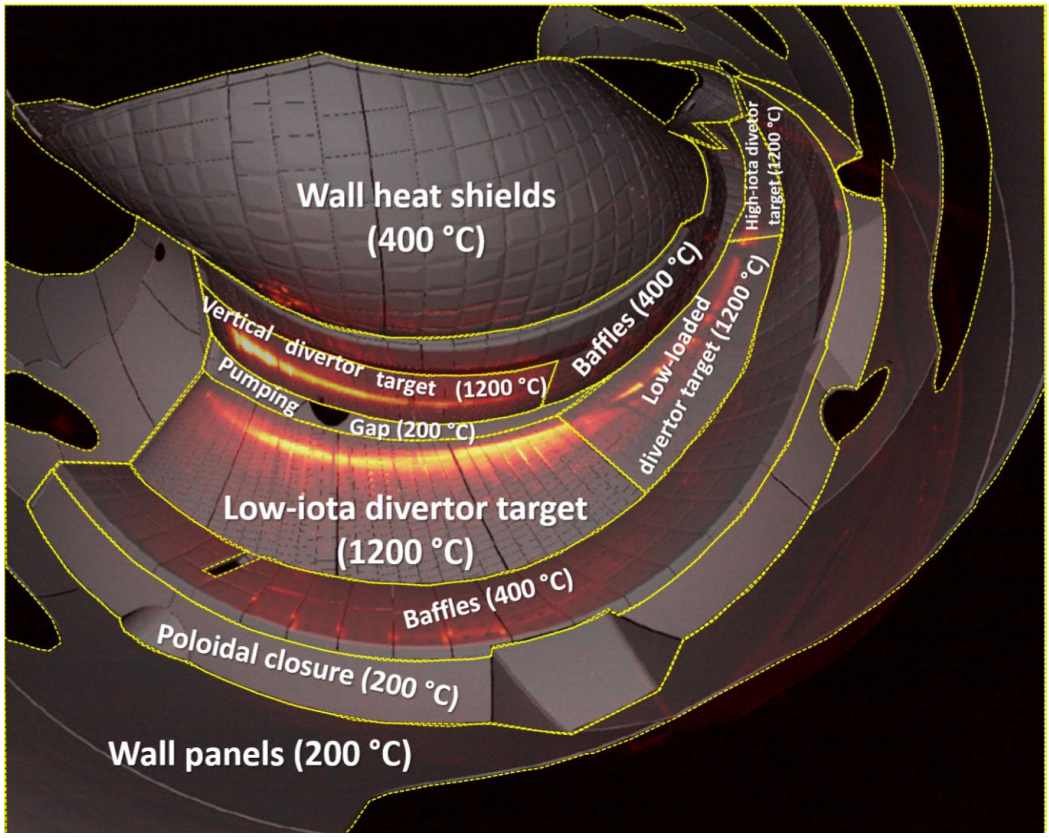

Figure 9.3: Infrared image overlaid on the CAD of the different plasma-facing components with their maximum operational temperatures [21]

For the early limiter configuration of W7-X, the IR cameras were calibrated and their images were projected in a 2D grid where the curvilinear distances were preserved despite the shape of the W7-X vessel [160]. Regarding the divertor configuration, in [23] a 2D projection of the divertor surface from the 3D CAD model is produced. Each divertor component is made of consecutive thermally insulated target elements or the so called divertor fingers. The mapping of the divertor in a 2D space eases the adoption of the THEODOR code, which uses finite differences along 2 dimensions, for the postprocessing of the IR camera images and the analysis of the heat fluxes.

9.3 Thermal protection activities at W7-X and WEST

Using the temperature data, researchers at W7-X developed an overload detection algorithm [21], which analyses the calibrated images and evaluates the risk of overload for each PFC by estimating the heat flux with a transient 1D heat diffusion assumption [161]. This model is valid for uncooled as well as cooled components before reaching the steady-state temperature or during fast transient heat loads when the heat propagates into the material down to the actively cooled heat sink. Close to the steady state, however, this model overestimates the increase in the surface temperature, resulting in a conservative assessment. Furthermore, since the estimation involves the approximation of the temperature time derivative

with the temperature difference between consecutive IR images, the heat flux measurement is very noisy, resulting in a prediction with high uncertainty [21].

When the PFCs surface temperature becomes too high, the overload risk rises, and the interlock system stops the experiment.

Unfortunately, this approach can protect the reactor only by prematurely terminating the experiment when the integrity of the wall tiles is at risk. For this reason, the research activity for the protection of nuclear fusion reactors is focusing on the development of automatic routines for thermal events classification and characterization, with the final goal of implementing control schemes to allow the continuous operation of the device avoiding overloads.

9.3.1 Classification of thermal events

The main events detectable at W7-X are hot-spots (both overload and shine-through ones), surface layers, strike lines, leading edges and reflections. These events are present in Figure 9.4, where labelled thermal events are overlayed on the CAD of W7-X divertor. From the same Figure 9.4, it is possible to observe that these events have different shapes, and they are present in distinct parts of the wall:

- Hot-spots: The hot-spots are localized areas where the temperature is higher than the surroundings. They can be due to overloads or to shine-through effects of the plasma heating system. They have generally a rounded shape and it is important to identify them before their temperature overcomes the critical threshold for compromising the wall integrity. Hence, the same hot-spot may cause an alarm or not also depending on the divertor part where it is identified.
- Surface layers: Because of the tile erosion, thin surface layers can develop on the PFCs when the material is re-deposited. The surface layers have low thermal capacity and heat transfer properties. For this reason, surface layers appear in certain specific parts of the wall where the plasma-surface interaction caused the redeposition of the material. They may be misclassified as hot-spots and trigger false alarms [161], [162]. In [162] a method for automatically detecting surface layers has been developed to avoid the triggering of false alarms by the hot spot detection system. To identify surface layers, dedicated discharges with special modulated heat are periodically run and analysed.
- Strike lines: Strike lines are elevated temperature areas which are determined by the interaction of the plasma with the divertor. Their shape is usually long and narrow, and they are oriented along the divertor targets. They must be localized on the divertor to avoid damages to the first wall, and they must be tracked to avoid erosion and melting. Strike line characterization, modelling and tracking algorithms are under development for the real-time operation of W7-X.
- Leading edges: The leading-edge patterns are due to the misalignments and gaps between the divertor tiles. As well as surface layers, since the leading edges are due to misalignments in the tile positioning, they can be detected by

their position on the divertor. Moreover, they are oriented along the separation between tiles [163].

- Reflections: Due to the disposition of the divertor islands, a thermal event present in one divertor could be reflected and captured in the orthogonally placed one. Reflections are not consistent in time, and they are mirroring a phenomenon seen on the corresponding tile with lower intensity.

This allows the possibility to manually classify them by analysing the IR images and cross checking with other input data (such as the input power from heating systems). Unfortunately, the IR diagnostics measure the first wall temperature with a frame rate of 100 Hz, making it difficult for the operator to monitor the videos and act on the system in real-time. For this reason, a fully automatic routine is necessary to detect specific thermal events and their properties, enabling the implementation of feedback control schemes. In [164], [165] authors discuss the real-time system for the development of the thermal event classification at W7-X and in [166] their implementation on the GPU for the real-time system is discussed.

At WEST instead, researchers implemented an R-CNN approach for the automatic classification of the events from the temperature data of the IR cameras [155], after a thorough manual labelling of several IR camera videos. The events detected include hot-spots, strike points and reflections among the seven classes [155].

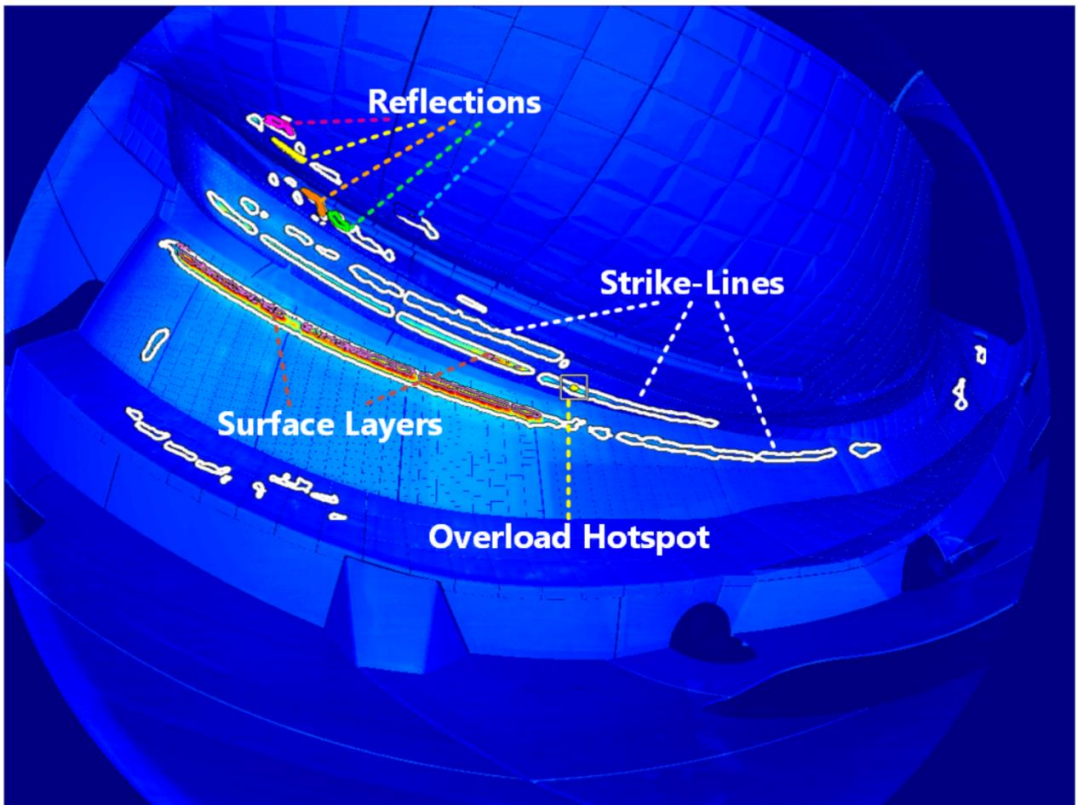

Figure 9.4: Thermal events overlayed on the CAD of the W7-X divertor [166]

9.3.2 Control of thermal events and real-time heat flux estimation

Presently, the termination of the experiment is the only possible action in the existing concept of the monitoring system at W7-X. Instead, researchers are investigating the possibility to implement active control schemes to move strike-lines away from a specific undesired position, i.e. a leading edge or the edge of the divertor target plates, by changing the currents of the control coils. For this purpose, the relationship between the control coil currents and the strike-line patterns has been studied during OP 1.2b [167]. Moreover, the electron cyclotron current drive (ECCD) was also tested in the first divertor campaign for an active control of the divertor power distribution, for which heat flux computation is essential. Similarly, [168], [169] evaluated the relationship between the magnetic configuration parameters, due to the planar and non-planar coils currents, and the heat distribution patterns with simulations and measured data for OP 1.1.

Chapter 10
Heat Flux computation at Wendelstein 7-X

10.1 Introduction

Wendelstein 7-X, the world largest superconducting advanced stellarator, aims to demonstrate high-performance steady-state experiments lasting up to 30 minutes. To this purpose, high heat flux (HHF) divertors capable of withstanding steady-state heat fluxes up to 10 MW/m^2 have been installed on the machine, in preparation for the next experimental campaign (OP2.1). The real-time heat flux estimation is pivotal for monitoring the divertor heat loads during the experiments. To measure the heat fluxes, an estimation of the temperature distribution in the bulk of the tile is necessary, which is provided by modelling the diffusion process inside the tile. For this goal, the temperature data has been processed with the THEODOR code [23], [151], [170] which estimates the heat flux on the surface of the tiles. In fact, strike-line patterns are identified starting from the heat flux images. Hence, the computation of heat fluxes is a necessary step for implementing a feedback control to ensure the safety of the first wall. In fact, for the implementation of real-time heat load control strategies, it is necessary to be able to estimate the divertor heat loads in real-time. Currently, THEODOR cannot be run in the real-time GPU at Wendelstein 7-X. Hence, part of the work of this thesis focused on the refactoring of the THEODOR code and on the development of alternative approaches for the real-time estimation of heat fluxes at W7-X.

This Chapter describes the study of the code, its optimization and parallelization in order to reduce the computation time.

10.2 Heat Flux calculation and THEODOR

The infrared cameras installed at W7-X monitor the surface temperature of each divertor. Then, since the THEODOR code is a Finite Difference Method (FDM) algorithm, it processes the temperature data in a 2D grid, which makes the projection of the measurements from 3D to 2D necessary [23]. The projection is made for each separate finger, which are separate thermally insulated divertor parts, to preserve the geometry of the divertor. In fact, the fingers are designed and manufactured to be flat. Moreover, the mapping also eases the visualization of the temperature or heat flux patterns on the different divertor modules.

At the beginning of the experiment the tile is at thermal equilibrium and there are not significant variations between the surface and the bottom temperature. However, as the experiment starts, the increase of the surface temperature determines a diffusion process in the bulk. Hence, the code should estimate the evolution of the temperature distribution in the target. To obtain the temperature distribution, the heat diffusion equation is solved consecutively:

$$\rho c_p = \frac{\partial T}{\partial t} = \nabla(k(T)\nabla T) \tag{10-1}$$

where, ρ is the density of the target material and c_p is the specific heat capacity of the target material. The equation is a nonlinear partial differential equation (PDE) and solving it directly is computationally demanding. The THEODOR code is based on the definition of the heat potential u:

$$u = \int_0^T k(t)dt \tag{10-2}$$

Using this definition, the derivatives of u can be rewritten as:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial T}\frac{\partial T}{\partial t} = k(T)\frac{\partial T}{\partial t} \tag{10-3}$$

$$\nabla \mathrm{u} = \frac{\partial u}{\partial T}\nabla T = k(T)\nabla T \tag{10-4}$$

$$\nabla^2 u = \nabla(k(T)\nabla T) \tag{10-5}$$

Now, the heat diffusion equation can be represented as a quasilinear partial differential equation:

$$\frac{\partial u}{\partial t} = D(u)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{10-6}$$

Here, $D(u)$ is the heat diffusion coefficient, x is the direction along the depth of the tile, y is the poloidal direction and z is the toroidal direction, as depicted in the sketch of the tile of Figure 10.1. Finally, u is the heat-potential, defined as:

$$u(T) = \int_0^T k(T)dT \tag{10-7}$$

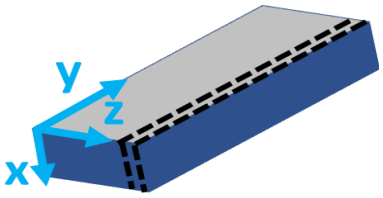

Figure 10.1: sketch of the divertor tile. x is the direction along the depth of the tile, y is the direction along the length of the tile (poloidal direction) and z is the toroidal direction. Dashed lines segment a profile, the computation domain of the PDE

In this formulation, the toroidal heat diffusion is neglected due to the presence of a homogeneous distribution of the strike line in finite toroidal range, as also reported in [23]. To solve a PDE (i.e., to make the solution unique), initial and boundary conditions must be specified. The surface temperature is sampled from the infrared cameras and acts as a boundary condition at the surface of the profile, while

the lateral edges are considered adiabatic. The mapping from the camera pixels to the real coordinates of the divertor points is provided by the spatial calibration techniques presented in [160].

When estimating the evolution of the temperature in the tile, the initial condition can be either assumed as a uniform constant temperature (at the beginning of the experiment) or reconstructed from the previous frames (during the experiment). The same conditions are assumed in the THEODOR code.

However, the code initially present only allowed for "offline" use, that is, for data processing at the end of the experiment, whereas real-time monitoring would obviously require real-time computation. In real-time use, the code should be used as follows:

1) The first temperature measurement made using the thermal imaging camera allows us to assume the temperature distribution in the tile. Since the first measurement is taken before any plasma is present, and the experiments are interspersed, the tile has a substantially uniform temperature.

2) Every 10 ms, the temperature of the divertor is measured using a thermal camera, and this becomes the boundary condition of the numerical problem to be solved. The initial condition is given by the assumption made in the previous iteration or step 1 (it is necessary to know the initial temperature throughout the tile).

3) Diffusion is simulated for the required time and the heat fluxes are estimated, i.e.:

$$q = -\frac{\partial u}{\partial x} \tag{10-8}$$

Surface heat fluxes are the ones of interest for the application. The x is the depth of the tile (vertical dimension) and y its length (horizontal).

10.3 Single step version of the code and optimization

To implement a real-time version, it was necessary to allow the diffusion calculation for only one time instant at a time, as shown in Figure 10.2. Figure 10.3, on the other hand, shows a comparison of the heat flux calculation performed with the two codes.

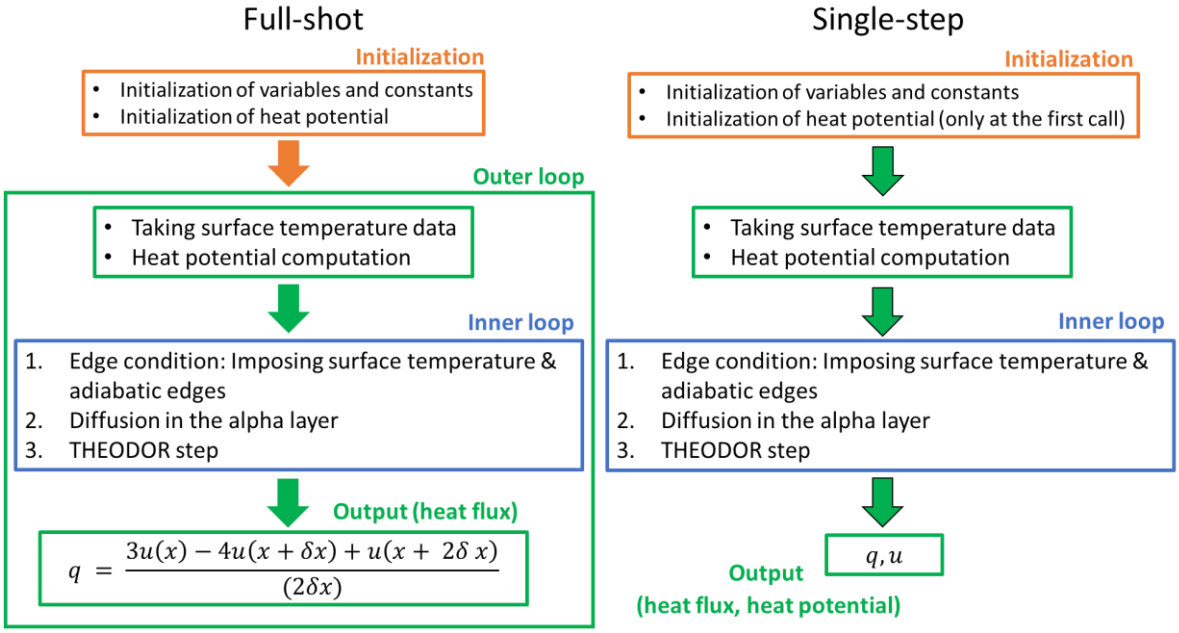

Figure 10.2: Left, the code in its offline version, used to process the data at the end of the experiment; right, the code that can be used for real-time use, one time step at a time

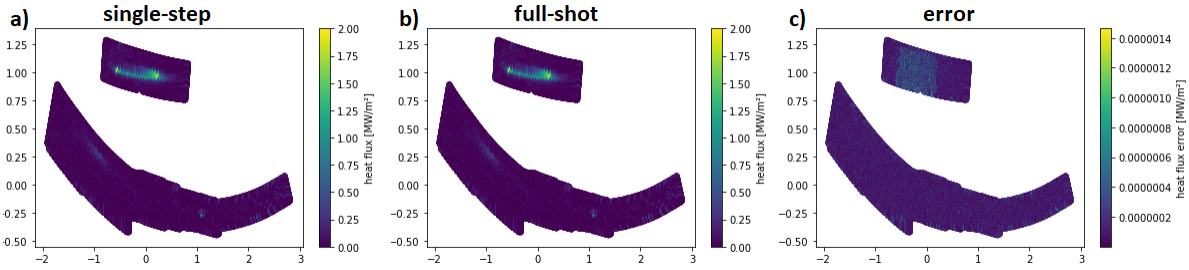

Figure 10.3: a) calculation of heat fluxes on the divertor using the version of the code for real-time use; b) calculation of heat fluxes on the divertor performed using the offline code. c) difference between the two results

Then, the performance of the code was measured: it was found that compared with the timing required for real-time use, which would require a maximum computation time of about 80 ms, the performance was insufficient (about 600 ms). Therefore, the code was optimized, and parallelization attempts were made using multi-processor computation. To do this, the code was loaded on a computer with 50 processors available so that the feasibility of parallelization of the code could be studied. The result of the parallelization is shown in Figure 10.4. In summary, it was shown that although the code numerically solves the equation on many mutually independent components (as can be seen from the almost 1/n reduction for the first values of the computation time), the scalability of the computation is possible only up to about 6 processes. This fact is due both to the creation of new processes necessary for parallelization and to the fact that the most onerous part of the code (the numerical resolution of the equation) is actually implemented efficiently via a

C++ executable. However, the time required to calculate thermal fluxes with 6 processes was substantially reduced to 1/6 the value with one processor.

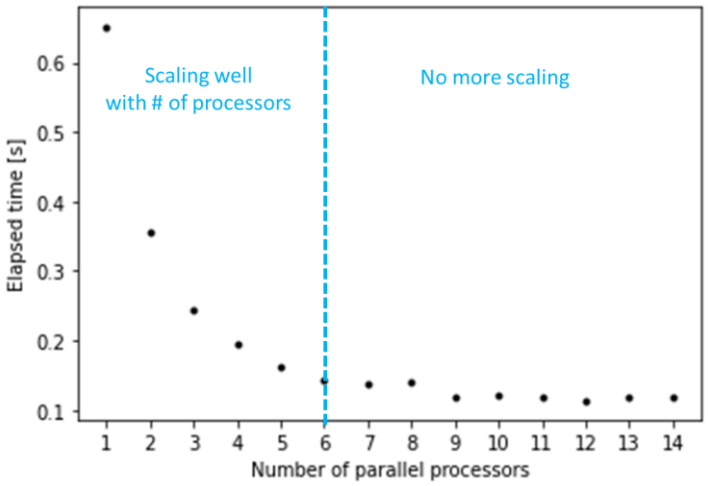

Figure 10.4: parallelization of heat flow calculation. In abscissa is represented the number of independent processors used, in the y-axis the computation time

In addition, further optimization was performed using the Numba library [171], for transforming python code into pseudo-compiled (or "just in time" compiled) code. This allowed, on the same machine used for parallelization, to achieve computation times around 80 ms. This time is ideally compatible with that of a real-time implementation; however, to confirm the results obtained, the code should be tested on the machine used for real-time control of the experiments. It was not possible to implement parallelization using GPUs with the version of the code available, as it would be necessary to rewrite the code entirely to allow data processing using GPU kernels.

Chapter 11
Physics Informed Neural Networks for heat flux estimation

11.1 Introduction

For porting the THEODOR code in the real-time system at W7-X, a GPU parallel implementation of the code should be developed. Hence, a Physics Informed Neural Network (PINN) model is proposed to speed up the heat-flux computation towards the real-time implementation. PINNs have several advantages with respect to the other numerical PDE solvers: they can be used to regress nonlinear PDE operators; they are mesh-free and can handle irregular domains; they are able to exploit the parallel computing capabilities of Graphical Processing Units (GPUs) [42], [43], [54]. Physics informed models can be trained to numerically solve partial differential equations by including physics-based criteria in the NN loss function. The "physics informed" models exploit the possibility of calculating the gradient of the output with respect to the input. In this case, the model is trained using as input the spatial position and time instant at which the solution is to be calculated, while the function to be minimized is based on the heat diffusion PDE described in 10.

11.2 Example with $\alpha = 1$

A first physics-informed neural network model was developed to solve heat diffusion in an arbitrary two-dimensional tile with 1m x 1m sides, time domain up to 0.1 seconds and constant diffusion coefficient $D = 1\frac{m^2}{s}$. A sketch of the proposed model is reported in Figure 11.1. The architecture of the neural network is that of a FC-NN with two hidden layers. Specifically, the developed neural network has an input layer with 3 scalar inputs, which indicate where the function should be calculated, 2 hidden layers with 32 neurons each and a hyperbolic tangent activation function, and a linear output layer where the function u is reconstructed. The model was developed and trained using the DeepXDE library for PINN [45].

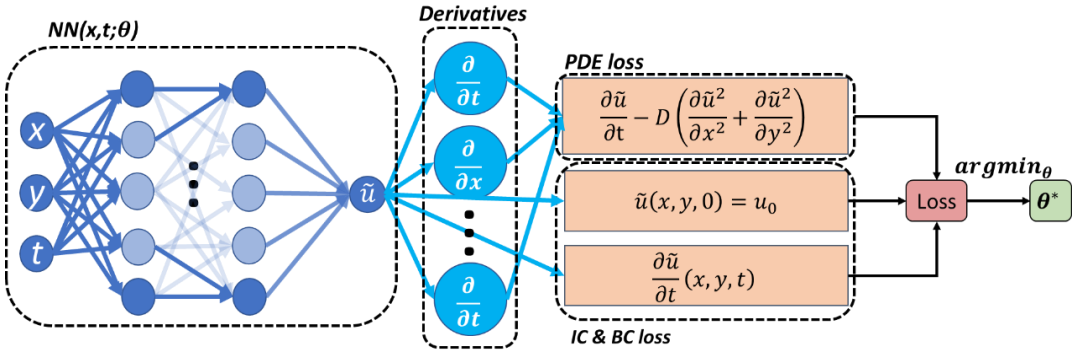

Figure 11.1: Scheme of a Physics Informed NN: the inputs are the scalar values of the time and spatial position where solution of the PDE should be computed. The network output can be automatically derived with respect to the inputs using automatic differentiation, enabling the satisfaction of the PDE. The other components of the loss are the boundary and initial conditions of the PDE. Adapted from [45]

Since α is constant and equal to 1, the value of u and T coincide. The output of the network is normalized between 0 and 1. In this first example, the initial condition of the problem is a uniform value equal to 0, while the boundary condition is a Gaussian on the upper surface ($x = 0$), with mean equal to 0.5 and standard deviation equal to 0.1. The boundary condition on the side walls ($y = 0$ and $y = 1$) is an adiabatic condition, so the normal derivative of u must be 0. There is no boundary condition on the bottom of the tile (the $x = 1$ wall) in the THEODOR code, so it has not been implemented in this model either.

Figure 11.2 shows the training of the network for 500000 iterations, with a learning rate of $3 \cdot 10^{-4}$. For training, 10000 points are randomly sampled from the domain of the function, i.e. the parallelepiped of sides $(x, y, t) = (1,1,0.1)$, while 100 points are sampled on each boundary surface. In order to maintain a proportion between the point density along surfaces t=0 (initial condition, space x space dimensions) and surfaces $x = 0$, $y = 0$, $y = 1$ (boundary conditions, space x time dimensions), a link must be established between the linear and volumetric point density. To maintain the same linear point density, the average sampling steps along x and y (dx, dy) will have to be equal, while the average sampling step along t (dt) will be calculated by scaling the spatial dimension through α: $dt = \frac{dx^2}{\alpha}$. This relationship comes from the normalization operation of the PDE.

At this point, the proportionality between the points on the surface and those in the volume can be maintained if the number of points on the volume will be $N_V = N_x N_y N_t$, while each surface will have $N_{Sx} = N_y N_t$, $N_{Sy} = N_x N_t$, $N_{St} = N_x N_y$ points, where N_x, N_y, N_t are the number of points along x, y, t. The training points are re-sampled every 10000 iterations. A sum of several contributions is used as the error function: the error on the PDE approximation, the error with respect to the value of the initial condition, the error with respect to the value of each boundary condition. In this case, the network is said to weakly satisfy the boundary conditions and the

initial condition, in contrast to the case in which the output of the network is multiplied by a distance function with respect to the edge of the domain and summed by a function that constrains the output to the values of the boundary conditions (in this case we speak of strong imposition of the boundary conditions). The error function can therefore be written as:

$$L(u,x,y,t) = \left\langle \left(\frac{\partial u}{\partial t} - D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right)^2 \right\rangle + \langle (u(x,y,0) - u_0)^2 \rangle + \left\langle \left(\frac{\partial u}{\partial y}(x,y_b,t) - g_E \right)^2 \right\rangle + \langle (u(0,y,t) - u_b)^2 \rangle \quad (11\text{-}1)$$

Where u_0 is the initial condition of the PDE, u_b is the boundary condition on the surface of the tile, expressed as the value of the function at the edge of the tile (or Dirichlet condition) and g_E is the boundary condition on the sides of the tile, which imposes a condition on the value of the derivative (Neumann condition). Finally, $N_D, N_0, N_{b,y}$. and $N_{b,x}$ are the number of points of in the domain, initial condition y and x boundary conditions respectively. The error function used is the mean square error (MSE). Together with the training error, the error on test points, which the network does not use for training, is evaluated. For the test, 10000 points were chosen randomly. Figure 11.2 shows that the network achieves convergence for an error of approximately $2 \cdot 10^{-3}$ and has no overfitting effects. In general, overfitting is the phenomenon whereby the data-driven model loses the ability to generalize to new data because it has approximated the training set data too specifically (hence the name overfitting). In general, it is not possible to provide the data-driven model with enough data to cover the entire variation space of the input variables. In the case of physics-informed networks, however, the boundaries of the domain are delimited and the points within it are randomly sampled, allowing the entire domain of the function to be explored.

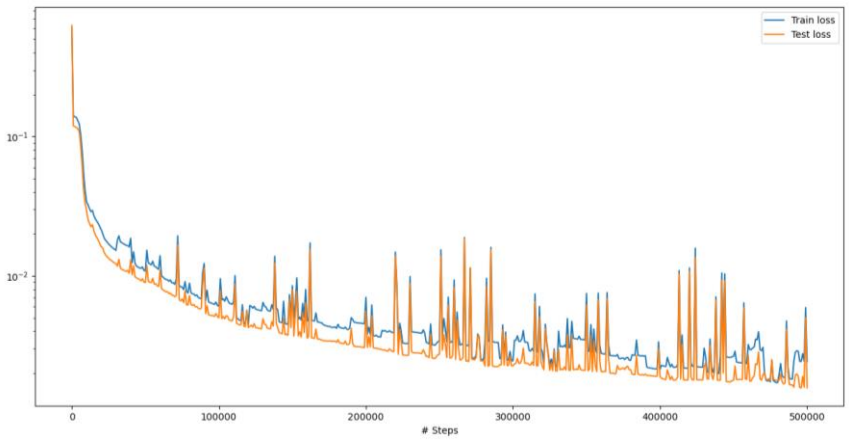

Figure 11.2: Network training loss: in blue training loss, in orange test loss

Figures 11.3-11.4 show the results of the reconstruction using the neural network for two different simulation time instants, compared with the reconstruction performed by the THEODOR code. Note that the temperature output from the network is dimensionless as the range can simply be rescaled to represent

the scale of variation of the real case. In Figure 11.3c, the error at the top of the tile is small and its values tend to concentrate towards the middle of the tile. Furthermore, Figure 11.4c shows the shift of the error towards the bottom of the tile, which is however the surface for which no boundary condition has been defined. In contrast, the approximation of the function at the top of the tile tends to improve over time, as can be seen in the comparison of Figure 11.3c and Figure 11.4c.

The network can then calculate heat fluxes by determining the gradient of the output with respect to the input. In particular, if we are interested in the heat fluxes at the surface, we have to calculate the derivative with respect to the normal: $q = -\frac{\partial u}{\partial x}$ i.e. the derivative in the direction normal to the surface. Figure 11.5-11.6 show the thermal fluxes reconstructed in this way, compared with those of THEODOR. Again, the error on the surface decreases over time, as can be seen by comparing Figures 11.5c-11-6c. In general, the method of 'physics-informed' neural networks seems promising in view of its application for the calculation of heat fluxes in the real problem.

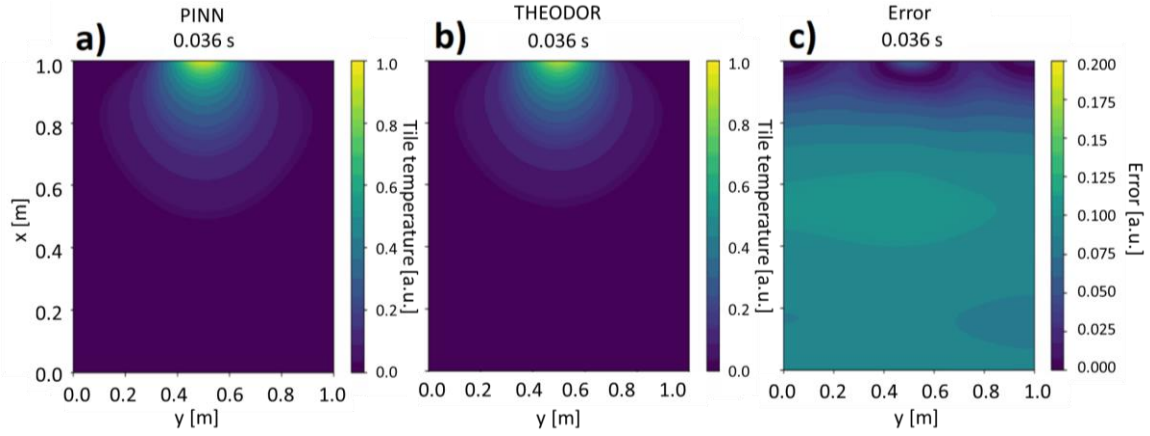

Figure 11.3: Temperature reconstruction at time $t = 0.036s$ a) Temperature reconstructed using the PINN; b) Temperature reconstructed using THEODOR; c) Absolute error

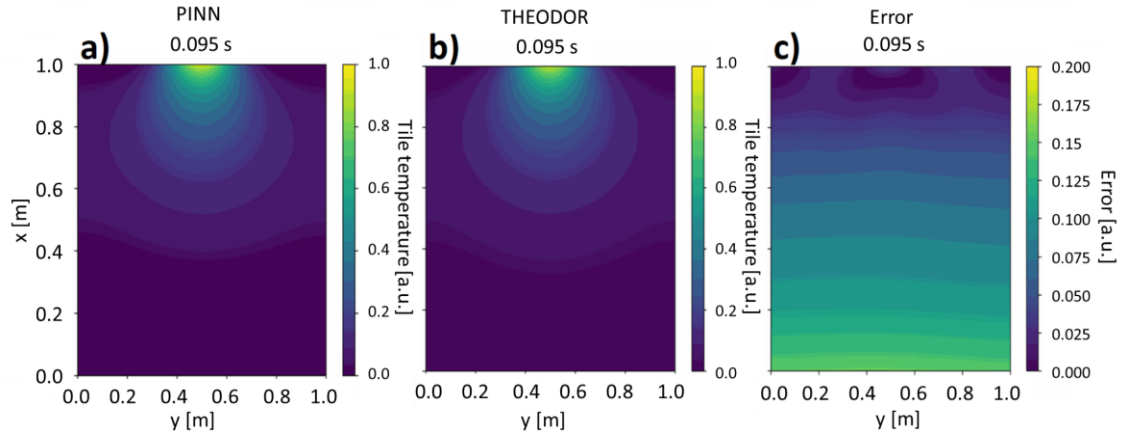

Figure 11.4: Temperature reconstruction at time $t = 0.095s$ a) Temperature reconstructed using the PINN; b) Temperature reconstructed using THEODOR; c) Absolute error

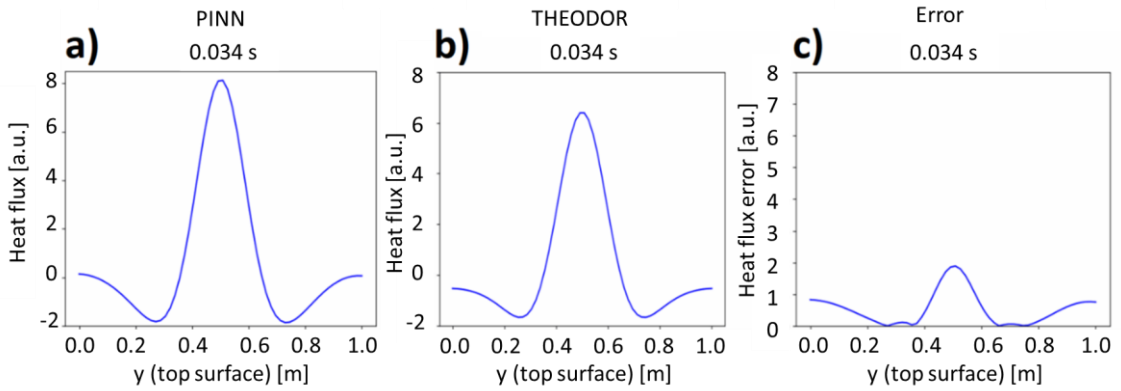

Figure 11.5: Heat flux reconstruction at time $t = 0.034s$ a) Heat flux reconstructed using the PINN; b) Heat flux reconstructed using THEODOR; c) Absolute error

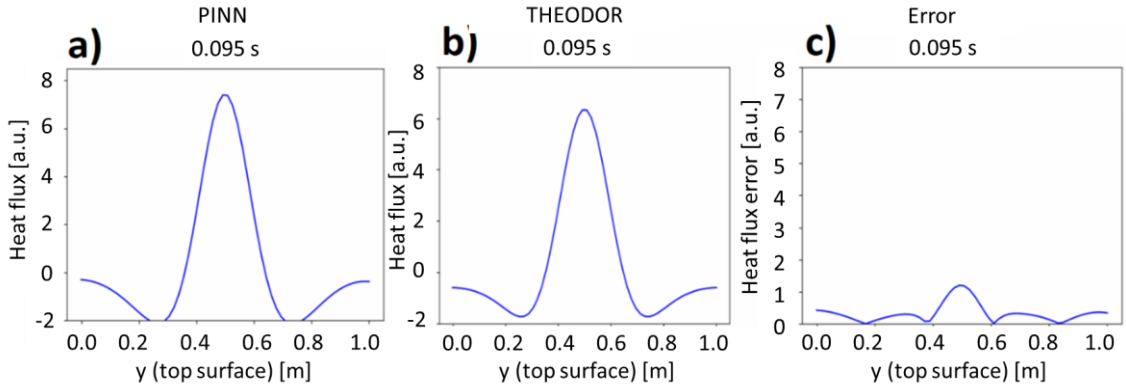

Figure 11.6: Heat flux reconstruction at time $t = 0.095s$ a) Heat flux reconstructed using the PINN; b) Heat flux reconstructed using THEODOR; c) Absolute error

11.3 Application in a real size domain

In the following sub-sections, the results of two simulations of the heat diffusion in the divertor tile are discussed. For both examples, the computational domain is consistent with the typical tile profile size, with (x_{max}, y_{max}) = (28 mm, 560mm) sides, and PDE evolution time up to $t_{end} = 0.1$s. The network spatial inputs are rescaled by dividing them with respect to y_{max}, and the time is divided by t_{end}. Finally, the diffusion coefficient D is then rescaled accordingly. The computational domain for the NN becomes:

$$\begin{cases} x_N \in [0,0.05] \\ y_N \in [0,1] \\ t_N \in [0,1] \end{cases}, D_N = D \cdot \frac{t_{end}}{{y_{max}}^2} \tag{11-2}$$

where D_N is the dimensionless diffusion coefficient for the normalized equation and (x_N, y_N, t_N) the normalized inputs to the PINN. In the first example, the PINN learns the equation with a constant $D = 70 \cdot 10^{-6}\text{mm}^2/\text{s}$, while in the second one material properties are introduced and $D(T)$ depends on the temperature (hence on the heat potential u). In both examples the initial and boundary conditions are fixed.

11.3.1 Architecture optimization

Following to the application of the model in the toy model, the parameters of the problem were The number of layers, the number of neurons per layer and the learning rate were optimized with a Bayesian optimization scheme, an automatic optimization scheme where the network performance is modelled as a sample from a Gaussian Process [172]. The optimized network is a Fully Connected Neural Network (FC-NN) with 10 hidden layers, each of them with 97 neurons, and with a hyperbolic tangent (tanh) activation function.

11.3.2 Diffusion with constant D

The initial condition of this example is a profile with a uniform heat potential of 0, while the top boundary condition (in the normalized domain) is a gaussian heat potential on the upper surface (x=0), with amplitude 1, centered in 0.5 and with a standard deviation of 0.1. The amplitude of the gaussian is normalized between 0 and 1, but a simple rescaling of the output would allow to adapt the range to the real case. Figure 11.7a compares the heat flux estimated with the PINN model to the ones computed with a 2D version of THEODOR, with a relative error below 2% on the heat flux values.

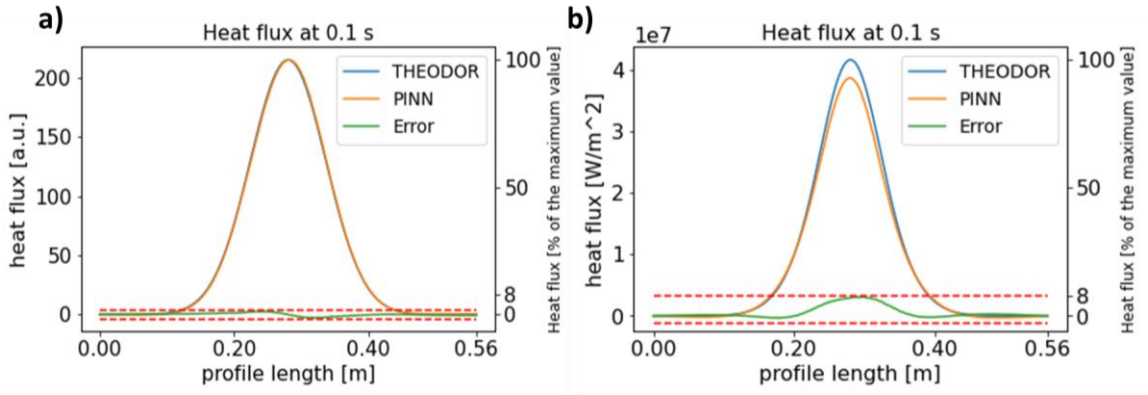

Figure 11.7: a) - constant D case: Heat flux on the surface of a tile by THEODOR (blue line), PINN (orange line) and Error (green line) in the absolute and percentage scale, respectively at the left and right side of the plot. Red dashed lines delimit the error range in [-2,2]% (right y-axis). b) – $D(T)$ case: Heat flux on the surface of a tile with THEODOR (blue line), PINN (orange line) and Error (green line) in the absolute and percentage scale, respectively at the left and right side of the plot. Red dashed lines delimit the error range in [-3,8]% (right y-axis)

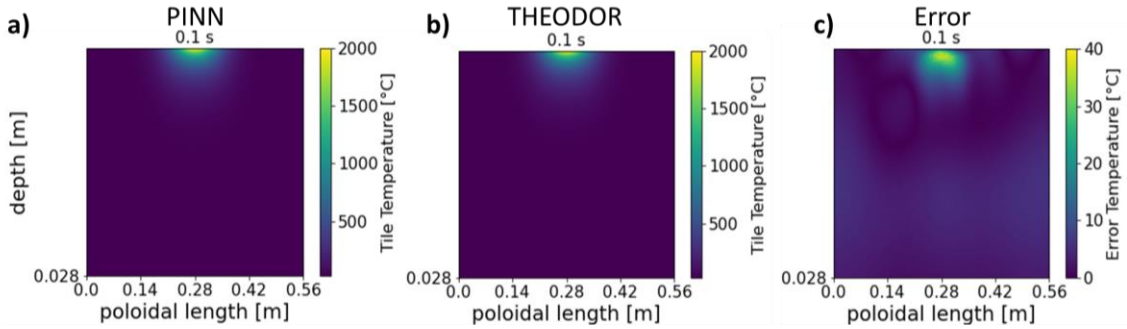

Figure 11.8: a): PINN reconstruction of the temperature distribution in the profile at t=0.1s. b): THEODOR reconstruction of the temperature distribution in the profile at t=0.1s c): Error computed as the absolute difference value between the two reconstructions

11.3.3 Diffusion with material properties

In this second example, the temperature dependency has been implemented by using the following nonlinear interpolation for D:

$$D(T) = a_{d0} + b_{d0} / \left(1 + \frac{T}{T_{d0}}\right)^2 \tag{11-3}$$

which is the same applied in THEODOR [23], [151]. This formula has been implemented by modifying the PDE loss: the T is computed from the heat potential u by inverting equation (2) and then $D(u)$ is computed and used in the L_{PDE}. Since there is a nonlinear dependency between D and T, the boundary condition was not normalized as in the previous case, but it was considered as a gaussian of temperature between 25°C and 2000°C. In this case, at the end of the simulation it was possible to achieve an error smaller than 8% on the temperature and heat-flux

reconstructions. The results of the heat flux reconstruction are reported in Figure 11.7b, while the temperature reconstruction for $t = 0.1s$ is shown in Figure 11.8.

11.4 Comparison of the computation time

To verify that the PINN can easily be ported on a GPU and benefit from the parallel computation of the values in the domain, the computation times of THEODOR and of the PINN model were compared on a tile with an increasing number of points. The PINN model was run on a GPU while THEODOR was run on a CPU on the same laptop. The results are shown in Figure 11.9. It is possible to see that, while the THEODOR computation time (red line) increases with the number of points that must be evaluated, the PINN computation time (blue line) is generally lower and stay constant despite the increasing number of points.

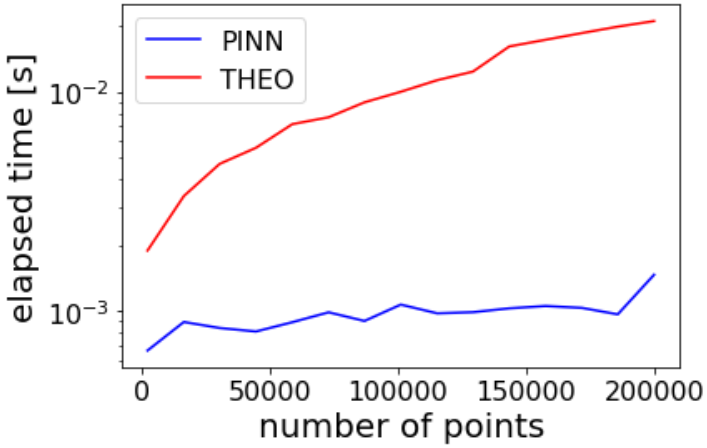

Figure 11.9: Elapsed computation time of the heat equation PDE for the PINN running on a GPU in blue and the THEODOR code running on a CPU in red

11.5 Further developments

In this Chapter, the development of a PINN model for the solution of the heat equation PDE has been addressed. However, the conditions of this problem still differ from the real application. In a real-time framework, the initial conditions will change, and the model must be able to solve the PDE starting from any initial condition. Hence, an improvement of the model is under development, exploiting the possibility to enter the parameters of the PDE, such as the boundary and initial conditions, as input to the network. This is a current problem under study in the physics informed machine learning community. This is currently a problem under study in the physics informed machine learning community [42], [43], [54], [56].

11.5.1 Generalization of the initial condition and next steps

A first step towards the development of the model has been the extraction or generation of a set of initial conditions, which should be representative of the variety of the possible experimental conditions, to train the physics informed model. For this purpose, the heat-flux profiles from a set of W7-X experiments were analyzed. The

set includes six experiments in standard configuration performed during OP1.2b campaign for strike line control studies [167]. For each profile in the divertor, the profile length was normalized, and the profile shape was compared to the other ones, in order to select a reduced set of relevant profiles. A metric based on the Root Normalised Mean Squared Error (RNMSE) was introduced to compare two different profiles y_1 and y_2:

$$RNMSE = \sqrt{\frac{\sum_i (y_1 - y_2)^2}{\sum_i (y_1)^2}} \tag{11-4}$$

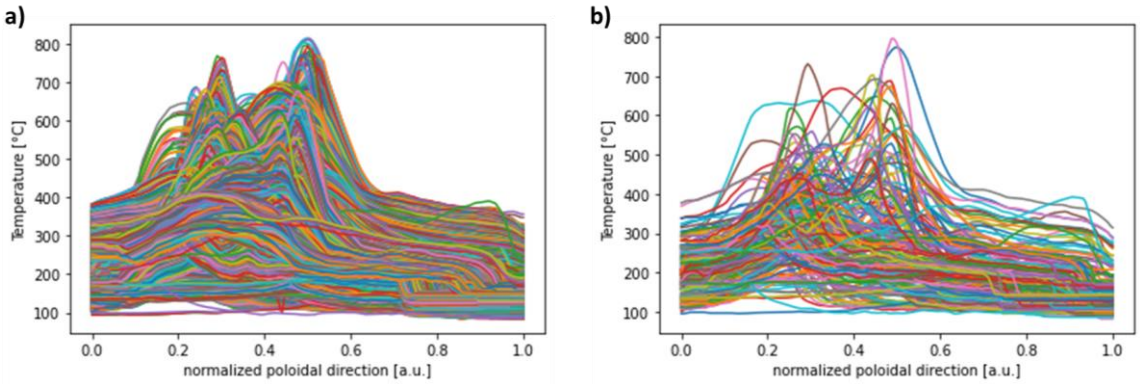

Figure 11.10: a) Plot of all the profiles from the selected experiments. b) Plot of the 185 extracted profiles

A profile was selected from the list of profiles, if the RNMSE was higher than a threshold for all the previously extracted profiles. With a threshold of 0.15, a total of 185 profiles was extracted from the 80000 starting ones. Figure 11.10a shows all the profiles, while Figure 11.10b shows the extracted profiles. It is possible to see that the main different shapes of profiles are preserved. The goal is the extraction of a sufficient number of example profiles to develop a generative approach. In particular, by parametrizing the example profiles it is possible to model the variation of their parameters to enable the generation of synthetic profiles for the training of the PINN.

The profiles have been then parametrized with a gaussian fitting. Three gaussians and a bias was used to fit every profile, for a total of 10 fitting parameters. Since there is a correlation among the different parameters, to generate new profiles the parameter matrix has been decomposed in principal components by means of Principal Component Analysis [167]. Finally, a new set of gaussian parameters can be generated by shuffling each principal component [173], so that the same statistical distributions of and correlation among principal components is preserved, and reconstructing the parameters by PCA inversion.

The Pearson Correlation Coefficient of the original matrix is compared with the one reconstructed with the shuffling in Figure 11.11a-c, while the distribution of the parameters of one gaussian is shown in Figure 11.12. Among all the parameters,

the standard deviation parameter was slightly more impacted from the reconstruction than the other ones.

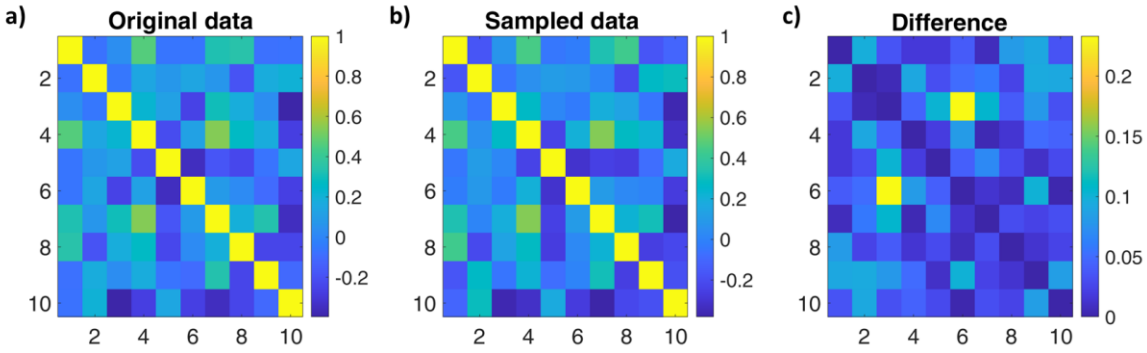

Figure 11.11: a) Pearson Correlation Coefficients of the original fitting parameters data. b) Pearson Correlation Coefficients of the fitting parameters data reconstructed with the shuffled PC. c) Difference between the two matrices (absolute value)

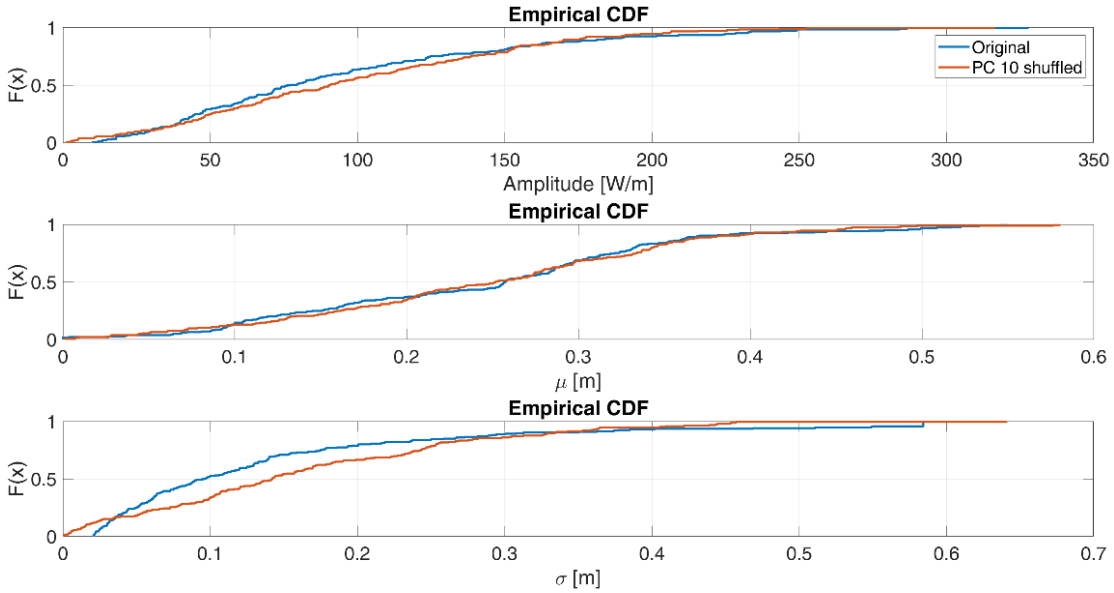

Figure 11.12: Cumulative distributions of the parameters of one of the gaussians, in the original matrix and in the reconstructed one. From top to bottom, cumulative distribution of the amplitude, of the mean and of the standard deviation of the gaussian.

In Figure 11.13, an example of an extracted profile, together with the fitting and the generated profile is shown. It is possible to notice the good quality of the fit and that the generated profile retains the properties of the extracted one.

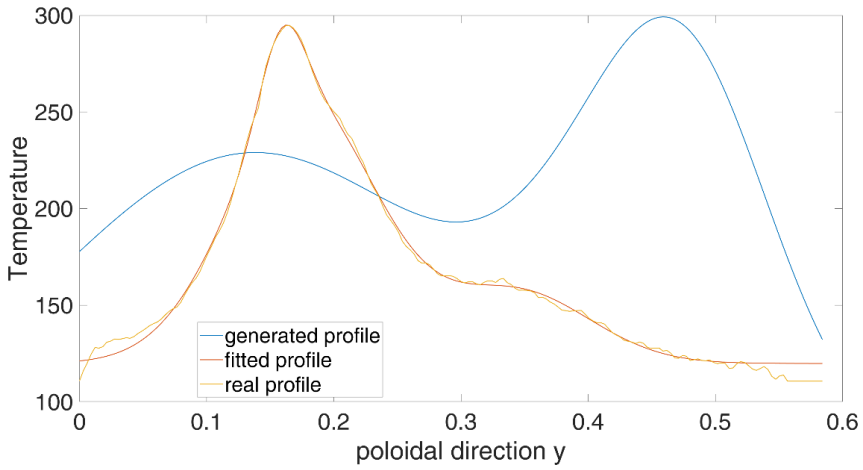

Figure 11.13: The yellow line shows the original profile extracted from the experimental data, the red one the gaussian fitting and the blue one is a profile generated with the shuffling of the PC loadings.

11.5.2 Training of a parametrized model

The technique presented in this report can be used to generate an arbitrary number of profiles by simply fitting the distribution of the PC loadings and by sampling randomly from their distribution to create new profiles. Then a parametrized model, such as the ones in [43], [54], [55], will be trained by sampling the initial condition in a fixed set of points and a training procedure will be performed.

During the new training procedure, the model will learn the solution of a certain PDE (i.e. for a specific initial condition) for a given number of epochs, and then the initial condition value will be randomly resampled so that the model will need to learn the solution operator of the PDE to reduce the loss.

Since the new divertor will have a boundary condition on the bottom of the tile, i.e., the temperature of the cooling water, another boundary condition of the problem will be introduced to take into account the effect of active cooling.

Part 3 Conclusions

In view of ITER and of the development of nuclear fusion power plants, the handling of the high heat fluxes due to the plasma during the continuous operation of the reactor is a critical task. At the moment, infrared cameras are used in stellarator and tokamaks to monitor the temperature and the power loads transferred to the first wall, but usually the power loads are analysed in post processing after each pulse, as it is currently done at W7-X.

However, in the coming experimental campaigns, W7-X will sustain the plasmas for up to 30 minutes, and the first wall of the machine will be exposed to hot temperature and heat loads for long times. In particular, the divertor tiles are subject to the risk of erosion and melting if localised heat loads overcome the material limits. For this reason, to prevent damages to the wall tiles while keeping high performances during the discharges, researchers are investigating approaches for the real-time monitoring and control of the heat loads in the first wall.

A system for the protection of the first wall is currently under development at W7-X, and it monitors the IR camera data to find overloads in the different wall parts. Moreover, it estimates heat fluxes with a 1D transient model to preventively stop the operation or the heating system before the maximum operational surface temperature of a component is reached.

On the other hand, the real-time control of heat loads requires a better measurement of the heat flux patterns. The relationship between the magnetic configuration parameters and the heat load patterns non-linear and complex, researchers adopted machine learning approaches for modelling it in specific OP 1.2b experiments, with promising results. In these studies, heat fluxes were reconstructed using THEODOR, which is a FDM code for the offline post-processing of IR measurement data. For the application of a feedback control scheme, the heat fluxes should be available in real-time ideally after each frame (the IR camera samples at 100 Hz), which requires the speed up of the code.

This Part described the activity in the refactoring of the THEODOR code exploiting the parallel computation of the heat fluxes of each finger across different CPUs and a compilation with Numba to achieve computation times around 80 ms. This time is ideally compatible with that of a real-time implementation; however, the real-time control system could only run the code on a GPU, since the CPU is used for handling other critical codes. It was not possible to implement parallelization using GPUs with the version of the code available, as it would be necessary to rewrite the code entirely to allow data processing using GPU kernels.

For this reason, a Physics-Informed Neural Network approach is adopted to port the computation of the heat partial differential equation on a GPU. The PINN method is based on recent advances in the automatic differentiation and machine learning and provides a very flexible model to solve a PDE in a mesh-free domain.

Moreover, the PINN can be straightforwardly run on a GPU, hence allowing the real-time use of the method. In the simulations shown in this work, the PINN allows to compute the PDE solution and the heat flux computation error with respect to THEODOR is lower than 8%. Future work on the PINN model will focus on providing initial and boundary conditions as inputs, hence solving a parametrized PDE and enabling the real-time evaluation of the heat flux.

Part 4: Conclusions and future work

Conclusions and future work

Problems addressed

Nuclear fusion research has the ambitious goal of providing an almost unlimited amount of energy with a very low impact for the environment. However, the research faces great technological challenges for the development of a fusion power plant. The next generation devices, such as the International Thermonuclear Experimental Reactor (ITER), the Smallest Private-Funded Affordable Robust Compact (SPARC) are being built with the aim of demonstrating the feasibility of the magnetic confinement fusion approach for energy production, since they should reach a net energy gain from the fusion reaction ($Q>1$). Moreover, stellarator reactor design is being investigated.

In view of the next generation devices, one of the main challenges regards the protection of the reactor components from the interaction with the plasma. The first challenge is posed by disruptions in Tokamaks; the huge electromechanical stresses may compromise the integrity of the magnetic coils and of the first wall. The second challenge is the handling of high heat fluxes during the operation of nuclear fusion devices, especially in view of continuous operation of a nuclear fusion power plant. Both these challenges undermine the economic viability of the power plants. In fact, for the plant to be viable, a very high capacity factor must be guaranteed, which means that the machine should be in operation most of the time and that the maintenance time should be minimized. At the moment, devices are operated in a pulsed manner and the longest discharge lasted around 17 minutes, and it was achieved at the Experimental Advanced Superconducting Tokamak (EAST).

Disruption prediction with data-driven methods

In the case of tokamak devices, the field of disruption prediction aims to understand the physical mechanisms which destabilize the plasma and to their automatic detection. Unfortunately, despite the presence of very comprehensive simulation codes, the real-time adoption of simulators for the safe operation of the device is not yet possible, and most of disruption predictors are data-driven models developed using the data from previous experiments.

This thesis is framed in this context and contributed with the development of an algorithm for automatically identifying the pre-disruptive phase of tokamak discharges, with the aim of automating the retraining of data-driven models, which are typically subject to the problem of ageing. Moreover, disruption prediction models based on the GTM, FC-NN and CNN Machine and Deep Learning methods have been introduced, exploiting early and late fusion techniques for the extraction of the relevant information. Finally, the models have been compared on the same database of discharges, with the same set of diagnostics and using common metrics. The purpose of this work is the determination of criteria to systematically compare different predictors, in view of the adoption by EUROfusion of common databases for

the testing of data-driven disruption predictors. It was observed that the accumulated fraction of detected disruption as a function of the warning time $\Delta t_{warning}$, together with the false alarm rate, provide a compact yet comprehensive overview of the predictor performance.

The results are promising, and the future work will focus on the porting of these models to different devices and to the investigation of cross-machine approaches.

Fast heat flux computation

Since the operation time of the future nuclear fusion devices will be in the order of hours, the handling of the high heat fluxes coming from plasma-wall interaction during the continuous operation of the reactor is a critical task. Infrared cameras in Stellarator and Tokamaks measure the surface temperature and their data is processed to compute the power loads transferred to the first wall. In most cases, the power loads are only reconstructed after each pulse, as currently done at W7-X.

In view of the coming experimental campaigns at W7-X, which will test the possibility to sustain the operation at high performance for up to 30 minutes, the monitoring of the first wall of the device will be pivotal. The divertor tiles may undergo local erosion and melting due to high heat loads during the experiment and researchers are investigating approaches for the real-time monitoring and control of heat loads.

A real-time system for the protection of the wall is under development at W7-X; it is based on the analysis of the IR camera data to detect overloads in the different components of the wall. This system also estimates heat loads using a 1D model to enable the preventive interruption of the operation or of the auxiliary heating before reaching a critical temperature. However, for the real-time control of heat loads during the experiment, the impact of the different magnetic configuration parameters of the device on the heat loads patterns should be understood. A preliminary study developed a model to learn the non-linear relationship between the magnetic configuration parameters and the heat load patterns with machine learning approaches in specific OP 1.2b experiments, with promising results. In these studies, heat fluxes were reconstructed offline using THEODOR, which is a FDM code for the post-processing of IR measurement data. For the development of a feedback control scheme which exploits this model, the heat fluxes should be available in real-time, and a speed up of the heat flux reconstruction becomes mandatory.

The work of the thesis focused first on the refactoring of the THEODOR code exploiting the parallel computing and the development of compiled code with the Numba library. However, since the real-time control system could only run the code on a GPU, and the development of a GPU version of the code would have required a significant refactoring, a Physics-Informed Neural Network (PINN) approach has been investigated to natively port the computation on a GPU.

The PINN method is based on recent advances in the automatic differentiation and machine learning and provides a very flexible model to solve a PDE in a mesh-free domain. The PINN can be straightforwardly run on a GPU, hence allowing the real-time use of the method. In the simulations shown in this work, the PINN allows to compute the PDE solution and the heat flux computation error with respect to THEODOR is lower than 8%. Future work on the PINN model will focus on providing initial and boundary conditions as inputs, hence solving a parametrized PDE and enabling the real-time evaluation of the heat flux. The development of a parametrized model is pivotal for the real-time application, since the tile condition will vary during the discharge.

References

[1] M. Girtan, A. Wittenberg, M. L. Grilli, D. P. S. de Oliveira, C. Giosuè, and M. L. Ruello, 'The Critical Raw Materials Issue between Scarcity, Supply Risk, and Unique Properties', *Materials*, vol. 14, no. 8, Art. no. 8, Jan. 2021, doi: 10.3390/ma14081826.

[2] G. Federici, W. Biel, M. R. Gilbert, R. Kemp, N. Taylor, and R. Wenninger, 'European DEMO design strategy and consequences for materials', *Nucl. Fusion*, vol. 57, no. 9, p. 092002, Jun. 2017, doi: 10.1088/1741-4326/57/9/092002.

[3] H. Han *et al.*, 'A sustained high-temperature fusion plasma regime facilitated by fast ions', *Nature*, vol. 609, no. 7926, Art. no. 7926, Sep. 2022, doi: 10.1038/s41586-022-05008-1.

[4] M. Schirber, 'Gaining Ground in Nuclear Fusion', *Physics*, vol. 15, p. 195, Dec. 2022.

[5] G. McCracken and P. Stott, 'Chapter 5 - Magnetic Confinement', in *Fusion*, G. McCracken and P. Stott, Eds., Burlington: Academic Press, 2005, pp. 47–60. doi: 10.1016/B978-012481851-4/50007-X.

[6] S. Li, H. Jiang, Z. Ren, and C. Xu, 'Optimal Tracking for a Divergent-Type Parabolic PDE System in Current Profile Control', *Abstr. Appl. Anal.*, vol. 2014, p. e940965, Jun. 2014, doi: 10.1155/2014/940965.

[7] K. Ikeda, 'ITER on the road to fusion energy', *Nucl. Fusion*, vol. 50, no. 1, p. 014002, Dec. 2009, doi: 10.1088/0029-5515/50/1/014002.

[8] J. Wesson and D. J. Campbell, *Tokamaks*. OUP Oxford, 2011.

[9] Aledda, Raffaele, 'Manifold Learning Techniques and Statistical Approaches Applied to the Disruption Prediction in Tokamaks', PhD Thesis, University of Cagliari, 2014.

[10] P. C. de Vries *et al.*, 'Survey of disruption causes at JET', *Nucl. Fusion*, vol. 51, no. 5, p. 053018, Apr. 2011, doi: 10.1088/0029-5515/51/5/053018.

[11] A. H. Boozer, 'Theory of tokamak disruptions', *Phys. Plasmas*, vol. 19, no. 5, p. 058101, May 2012, doi: 10.1063/1.3703327.

[12] P. C. de Vries *et al.*, 'The influence of an ITER-like wall on disruptions at JET', *Phys. Plasmas*, vol. 21, no. 5, p. 056101, May 2014, doi: 10.1063/1.4872017.

[13] B. Esposito, G. Granucci, P. Smeulders, J. R. Martín-Solís, and L. Gabellieri, 'Disruption Avoidance in the Frascati Tokamak Upgrade by Means of Magnetohydrodynamic Mode Stabilization Using Electron-Cyclotron-Resonance Heating', *Phys. Rev. Lett.*, vol. 100, no. 4, p. 045006, Feb. 2008, doi: 10.1103/PhysRevLett.100.045006.

[14] J. L. Barr *et al.*, 'Development and experimental qualification of novel disruption prevention techniques on DIII-D', *Nucl. Fusion*, vol. 61, no. 12, p. 126019, Oct. 2021, doi: 10.1088/1741-4326/ac2d56.

[15] A. Pau *et al.*, 'A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET', *Nucl. Fusion*, vol. 59, no. 10, p. 106017, Aug. 2019, doi: 10.1088/1741-4326/ab2ea9.

[16] H. Fountain, 'A Dream of Clean Energy at a Very High Price', *The New York Times*, Mar. 27, 2017. Accessed: Sep. 27, 2022. [Online]. Available: https://www.nytimes.com/2017/03/27/science/fusion-power-plant-iter-france.html

[17] Y. Xu, 'A general comparison between tokamak and stellarator plasmas', *Matter Radiat. Extrem.*, vol. 1, no. 4, pp. 192–200, Jul. 2016, doi: 10.1016/j.mre.2016.07.001.

[18] T. Andreeva *et al.*, 'Magnetic configuration scans during divertor operation of Wendelstein 7-X', *Nucl. Fusion*, vol. 62, no. 2, p. 026032, Jan. 2022, doi: 10.1088/1741-4326/ac3f1b.

[19] T. Rummel *et al.*, 'The Wendelstein 7-X Trim Coil System', *IEEE Trans. Appl. Supercond.*, vol. 24, no. 3, pp. 1–4, Jun. 2014, doi: 10.1109/TASC.2013.2284671.

[20] E. Jauregi, D. Ganuza, I. García, J. M. Del Río, and T. Rummel, 'Power supply of the control coils of Wendelstein 7-X experiment', *Fusion Eng. Des.*, vol. 58–59, pp. 79–86, Nov. 2001, doi: 10.1016/S0920-3796(01)00357-X.

[21] A. Puig Sitjes *et al.*, 'Real-Time Detection of Overloads on the Plasma-Facing Components of Wendelstein 7-X', *Appl. Sci.*, vol. 11, no. 24, Art. no. 24, Jan. 2021, doi: 10.3390/app112411969.

[22] M. W. Jakubowski *et al.*, 'Infrared imaging systems for wall protection in the W7-X stellarator (invited).', *Rev. Sci. Instrum.*, 2018, doi: 10.1063/1.5038634.

[23] Y. Gao *et al.*, 'Methods for quantitative study of divertor heat loads on W7-X', *Nucl. Fusion*, vol. 59, no. 6, p. 066007, Jun. 2019, doi: 10.1088/1741-4326/ab0f49.

[24] M. Endler *et al.*, 'Managing leading edges during assembly of the Wendelstein 7-X divertor', *Plasma Phys. Control. Fusion*, vol. 61, no. 2, p. 025004, Nov. 2018, doi: 10.1088/1361-6587/aaef52.

[25] S. A. Lazerson *et al.*, 'Error fields in the Wendelstein 7-X stellarator', *Plasma Phys. Control. Fusion*, vol. 60, no. 12, p. 124002, Nov. 2018, doi: 10.1088/1361-6587/aae96b.

[26] K. C. Hammond *et al.*, 'Drift effects on W7-X divertor heat and particle fluxes', *Plasma Phys. Control. Fusion*, vol. 61, no. 12, p. 125001, Oct. 2019, doi: 10.1088/1361-6587/ab4825.

[27] C. M. Bishop, M. Svensén, and C. K. I. Williams, 'GTM: The Generative Topographic Mapping', *Neural Comput.*, vol. 10, no. 1, pp. 215–234, Jan. 1998, doi: 10.1162/089976698300017953.

[28] S. Haykin, S. S. Haykin, and S. A. HAYKIN, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[29] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, 'A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects', *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2021, doi: 10.1109/TNNLS.2021.3084827.

[30] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis, 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations', *J. Comput. Phys.*, 2019, doi: 10.1016/j.jcp.2018.10.045.

[31] T. Kohonen, 'The self-organizing map', *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990, doi: 10.1109/5.58325.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, 'Maximum Likelihood from Incomplete Data via the EM Algorithm', *J. R. Stat. Soc. Ser. B Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[33] C. M. Bishop and P. of N. C. C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[34] Y. Bengio, 'Learning Deep Architectures for AI', *Found. Trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009, doi: 10.1561/2200000006.

[35] M. Z. Alom *et al.*, 'A State-of-the-Art Survey on Deep Learning Theory and Architectures', *Electronics*, vol. 8, no. 3, Art. no. 3, Mar. 2019, doi: 10.3390/electronics8030292.

[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, 'Learning representations by back-propagating errors', *Nature*, vol. 323, no. 6088, Art. no. 6088, Oct. 1986, doi: 10.1038/323533a0.

[37] S. Ioffe and C. Szegedy, 'Batch normalization: accelerating deep network training by reducing internal covariate shift', in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, in ICML'15. Lille, France: JMLR.org, Jul. 2015, pp. 448–456.

[38] P. Goyal *et al.*, 'Self-supervised Pretraining of Visual Features in the Wild', *ArXiv210301988 Cs*, Mar. 2021, Accessed: Feb. 08, 2022. [Online]. Available: http://arxiv.org/abs/2103.01988

[39] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, 'Deep Clustering for Unsupervised Learning of Visual Features', in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11218. Cham: Springer International Publishing, 2018, pp. 139–156. doi: 10.1007/978-3-030-01264-9_9.

[40] A. C. R. Marques, M. M. Raimundo, E. M. B. Cavalheiro, L. F. P. Salles, C. Lyra, and F. J. V. Zuben, 'Ant genera identification using an ensemble of convolutional neural networks', *PLOS ONE*, vol. 13, no. 1, p. e0192011, Jan. 2018, doi: 10.1371/journal.pone.0192011.

[41] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, 'Discovering Physical Concepts with Neural Networks', *Phys. Rev. Lett.*, vol. 124, no. 1, p. 010508, Jan. 2020, doi: 10.1103/PhysRevLett.124.010508.

[42] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, 'Physics-informed machine learning', *Nat. Rev. Phys.*, vol. 3, no. 6, Art. no. 6, Jun. 2021, doi: 10.1038/s42254-021-00314-5.

[43] S. Wang, H. Wang, and P. Perdikaris, 'Learning the solution operator of parametric partial differential equations with physics-informed DeepONets', *Sci. Adv.*, vol. 7, no. 40, p. eabi8605, doi: 10.1126/sciadv.abi8605.

[44] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Radul, Alexey Andreyevich Radul, and Jeffrey Mark Siskind, 'Automatic differentiation in machine learning: a survey', *J. Mach. Learn. Res.*, 2017.

[45] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, 'DeepXDE: A Deep Learning Library for Solving Differential Equations', *SIAM Rev.*, vol. 63, no. 1, pp. 208–228, Jan. 2021, doi: 10.1137/19M1274067.

[46] G. Pang, L. Lu, and G. E. Karniadakis, 'fPINNs: Fractional Physics-Informed Neural Networks', *SIAM J. Sci. Comput.*, vol. 41, no. 4, pp. A2603–A2626, Jan. 2019, doi: 10.1137/18M1229845.

[47] Liu Yang, Dongkun Zhang, and George Em Karniadakis, 'Physics-Informed Generative Adversarial Networks for Stochastic Differential Equations | SIAM Journal on Scientific Computing', vol. 42, no. 1, p. 10.1137/18M1225409, 2020.

[48] D. Zhang, L. Guo, and G. E. Karniadakis, 'Learning in Modal Space: Solving Time-Dependent Stochastic PDEs Using Physics-Informed Neural Networks', *SIAM J. Sci. Comput.*, vol. 42, no. 2, pp. A639–A665, Jan. 2020, doi: 10.1137/19M1260141.

[49] S. Wang, X. Yu, and P. Perdikaris, 'When and why PINNs fail to train: A neural tangent kernel perspective', *J. Comput. Phys.*, vol. 449, p. 110768, Jan. 2022, doi: 10.1016/j.jcp.2021.110768.

[50] S. Wang, Y. Teng, and P. Perdikaris, 'Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks', *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3055–A3081, Jan. 2021, doi: 10.1137/20M1318043.

[51] A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne, 'Mitigating Propagation Failures in PINNs using Evolutionary Sampling'. arXiv, Oct. 03, 2022. doi: 10.48550/arXiv.2207.02338.

[52] M. A. Nabian, R. J. Gladstone, and H. Meidani, 'Efficient training of physics-informed neural networks via importance sampling', *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 36, no. 8, pp. 962–977, Aug. 2021, doi: 10.1111/mice.12685.

[53] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, 'A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks', *Comput. Methods Appl. Mech. Eng.*, vol. 403, p. 115671, Jan. 2023, doi: 10.1016/j.cma.2022.115671.

[54] L. Lu, P. Jin, and G. E. Karniadakis, 'DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators', *Nat. Mach. Intell.*, vol. 3, no. 3, pp. 218–229, Mar. 2021, doi: 10.1038/s42256-021-00302-5.

[55] S. Wang and P. Perdikaris, 'Long-time integration of parametric evolution equations with physics-informed DeepONets', *ArXiv210605384 Phys.*, Jun. 2021, Accessed: Jan. 12, 2022. [Online]. Available: http://arxiv.org/abs/2106.05384

[56] Y. Nakamura, S. Shiratori, H. Nagano, and K. Shimano, 'Physics-Informed Neural Network with Variable Initial Conditions', presented at the 7th World Congress on Mechanical, Chemical, and Material Engineering, Aug. 2021. doi: 10.11159/htff21.113.

[57] ITER Physics Expert Group on Disruptions Plasma Control, MHD, and I. P. B. Editors, 'Chapter 3: MHD stability, operational limits and disruptions', *Nucl. Fusion*, vol. 39, no. 12, p. 2251, Dec. 1999, doi: 10.1088/0029-5515/39/12/303.

[58] K. Ikeda, 'Progress in the ITER Physics Basis', *Nucl. Fusion*, vol. 47, no. 6, p. E01, Jun. 2007, doi: 10.1088/0029-5515/47/6/E01.

[59] M. Hoelzl *et al.*, 'The JOREK non-linear extended MHD code and applications to large-scale instabilities and their control in magnetically confined fusion plasmas', *Nucl. Fusion*, vol. 61, no. 6, p. 065001, May 2021, doi: 10.1088/1741-4326/abf99f.

[60] C. R. Sovinec *et al.*, 'Nonlinear magnetohydrodynamics simulation using high-order finite elements', *J. Comput. Phys.*, vol. 195, no. 1, pp. 355–386, Mar. 2004, doi: 10.1016/j.jcp.2003.10.004.

[61] S. C. Jardin, J. Breslau, and N. Ferraro, 'A high-order implicit finite element method for integrating the two-fluid magnetohydrodynamic equations in two dimensions', *J. Comput. Phys.*, vol. 226, no. 2, pp. 2146–2174, Oct. 2007, doi: 10.1016/j.jcp.2007.07.003.

[62] P. C. de Vries, M. F. Johnson, I. Segui, and JET EFDA Contributors, 'Statistical analysis of disruptions in JET', *Nucl. Fusion*, vol. 49, no. 5, p. 055011, May 2009, doi: 10.1088/0029-5515/49/5/055011.

[63] E. J. Strait *et al.*, 'Progress in disruption prevention for ITER', *Nucl. Fusion*, vol. 59, no. 11, p. 112012, Jun. 2019, doi: 10.1088/1741-4326/ab15de.

[64] A. H. Boozer, 'Theory of tokamak disruptions', *Phys. Plasmas*, vol. 19, no. 5, p. 058101, May 2012, doi: 10.1063/1.3703327.

[65] S. A. Sabbagh *et al.*, 'Disruption event characterization EX/P6-26 and forecasting in tokamaks', in *IAEA Fusion Energy Conference*, 2018.

[66] S. A. Sabbagh *et al.*, 'Progress on disruption event characterization and forecasting in tokamaks and supporting physics analysis', in *46th EPS Conference on Plasma Physics, EPS 2019*, 2019.

[67] S. A. Sabbagh, J. W. Berkery, and et al, 'Tokamak Disruption Event Characterization and Forecasting Research and Expansion to Real-Time Application', in *28th IAEA Fusion Energy Conference (FEC 2020)*, 2021.

[68] F. G. Rimini *et al.*, 'The development of safe high current operation in JET-ILW', *Fusion Eng. Des.*, vol. 96–97, pp. 165–170, Oct. 2015, doi: 10.1016/j.fusengdes.2015.01.014.

[69] U. A. Sheikh *et al.*, 'Disruption avoidance through the prevention of NTM destabilization in TCV', *Nucl. Fusion*, vol. 58, no. 10, p. 106026, Aug. 2018, doi: 10.1088/1741-4326/aad924.

[70] M. Maraschek *et al.*, 'Path-oriented early reaction to approaching disruptions in ASDEX Upgrade and TCV in view of the future needs for ITER and DEMO', *Plasma Phys. Control. Fusion*, vol. 60, no. 1, p. 014047, Nov. 2017, doi: 10.1088/1361-6587/aa8d05.

[71] C. Rea *et al.*, 'Disruption prediction investigations using Machine Learning tools on DIII-D and Alcator C-Mod', *Plasma Phys. Control. Fusion*, vol. 60, no. 8, p. 084004, Jun. 2018, doi: 10.1088/1361-6587/aac7fe.

[72] A. Pau *et al.*, 'A First Analysis of JET Plasma Profile-Based Indicators for Disruption Prediction and Avoidance', *IEEE Trans. Plasma Sci.*, vol. 46, no. 7, pp. 2691–2698, Jul. 2018, doi: 10.1109/TPS.2018.2841394.

[73] C. Rea, K. J. Montes, A. Pau, R. S. Granetz, and O. Sauter, 'Progress Toward Interpretable Machine Learning–Based Disruption Predictors Across Tokamaks', *Fusion Sci. Technol.*, vol. 76, no. 8, pp. 912–924, Nov. 2020, doi: 10.1080/15361055.2020.1798589.

[74] E. Aymerich *et al.*, 'A statistical approach for the automatic identification of the start of the chain of events leading to the disruptions at JET', *Nucl. Fusion*, vol. 61, no. 3, p. 036013, Feb. 2021, doi: 10.1088/1741-4326/abcb28.

[75] R. Aledda, B. Cannas, A. Fanni, A. Pau, and G. Sias, 'Improvements in disruption prediction at ASDEX Upgrade', *Fusion Eng. Des.*, vol. 96–97, pp. 698–702, Oct. 2015, doi: 10.1016/j.fusengdes.2015.03.045.

[76] G. Sias, B. Cannas, A. Fanni, A. Murari, and A. Pau, 'A locked mode indicator for disruption prediction on JET and ASDEX upgrade', *Fusion Eng. Des.*, vol. 138, pp. 254–266, Jan. 2019, doi: 10.1016/j.fusengdes.2018.11.021.

[77] J. Zhu *et al.*, 'Scenario adaptive disruption prediction study for next generation burning-plasma tokamaks', *Nucl. Fusion*, vol. 61, no. 11, p. 114005, Oct. 2021, doi: 10.1088/1741-4326/ac28ae.

[78] R. A. Tinguely, K. J. Montes, C. Rea, R. Sweeney, and R. S. Granetz, 'An application of survival analysis to disruption prediction via Random Forests', *Plasma Phys. Control. Fusion*, vol. 61, no. 9, p. 095009, Aug. 2019, doi: 10.1088/1361-6587/ab32fc.

[79] Y. Zhang, G. Pautasso, O. Kardaun, G. Tardini, X. D. Zhang, and the A. U. Team, 'Prediction of disruptions on ASDEX Upgrade using discriminant analysis', *Nucl. Fusion*, vol. 51, no. 6, p. 063039, Jun. 2011, doi: 10.1088/0029-5515/51/6/063039.

[80] P. A. Lachenbruch, *Discriminant Analysis*. Hafner Press, 1975.

[81] R. Aledda, B. Cannas, A. Fanni, G. Sias, and G. Pautasso, 'Multivariate statistical models for disruption prediction at ASDEX Upgrade', *Fusion Eng. Des.*, vol. 88, no. 6, pp. 1297–1301, Oct. 2013, doi: 10.1016/j.fusengdes.2013.01.103.

[82] B. Cannas, R. S. Delogu, A. Fanni, P. Sonato, and M. K. Zedda, 'Support vector machines for disruption prediction and novelty detection at JET', *Fusion Eng. Des.*, vol. 82, no. 5, pp. 1124–1130, Oct. 2007, doi: 10.1016/j.fusengdes.2007.07.004.

[83] G. A. Rattá *et al.*, 'An advanced disruption predictor for JET tested in a simulated real-time environment', *Nucl. Fusion*, vol. 50, no. 2, p. 025005, Jan. 2010, doi: 10.1088/0029-5515/50/2/025005.

[84] G. A. Rattá, J. Vega, and A. Murari, 'Improved feature selection based on genetic algorithms for real time disruption prediction on JET', *Fusion Eng.*

Des., vol. 87, no. 9, pp. 1670–1678, Sep. 2012, doi: 10.1016/j.fusengdes.2012.07.002.

[85] J. M. López *et al.*, 'Implementation of the Disruption Predictor APODIS in JET's Real-Time Network Using the MARTe Framework', *IEEE Trans. Nucl. Sci.*, vol. 61, no. 2, pp. 741–744, Apr. 2014, doi: 10.1109/TNS.2014.2309254.

[86] G. A. Rattá, J. Vega, A. Murari, S. Dormido-Canto, and R. Moreno, 'Global optimization driven by genetic algorithms for disruption predictors based on APODIS architecture', *Fusion Eng. Des.*, vol. 112, pp. 1014–1018, Nov. 2016, doi: 10.1016/j.fusengdes.2016.02.049.

[87] J. M. López *et al.*, 'Implementation of the Disruption Predictor APODIS in JET's Real-Time Network Using the MARTe Framework', *IEEE Trans. Nucl. Sci.*, vol. 61, no. 2, pp. 741–744, Apr. 2014, doi: 10.1109/TNS.2014.2309254.

[88] T. Yokoyama *et al.*, 'Prediction of high-beta disruptions in JT-60U based on sparse modeling using exhaustive search', *Fusion Eng. Des.*, vol. 140, pp. 67–80, Mar. 2019, doi: 10.1016/j.fusengdes.2019.01.128.

[89] T. Yokoyama *et al.*, 'Likelihood Identification of High-Beta Disruption in JT-60U', *Plasma Fusion Res.*, vol. 16, pp. 1402073–1402073, 2021, doi: 10.1585/pfr.16.1402073.

[90] G. Cybenko, 'Approximation by superpositions of a sigmoidal function', *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989, doi: 10.1007/BF02551274.

[91] R. Yoshino, 'Neural-net disruption predictor in JT-60U', *Nucl. Fusion*, vol. 43, no. 12, pp. 1771–1786, Dec. 2003, doi: 10.1088/0029-5515/43/12/021.

[92] B. Cannas, A. Fanni, P. Sonato, and M. K. Z. and, 'A prediction tool for real-time application in the disruption protection system at JET', *Nucl. Fusion*, vol. 47, no. 11, pp. 1559–1569, Oct. 2007, doi: 10.1088/0029-5515/47/11/018.

[93] B. Cannas, A. Fanni, G. Pautasso, G. Sias, and P. Sonato, 'An adaptive real-time disruption predictor for ASDEX Upgrade', *Nucl. Fusion*, vol. 50, no. 7, p. 075004, Jun. 2010, doi: 10.1088/0029-5515/50/7/075004.

[94] A. Sengupta and P. Ranjan, 'Forecasting disruptions in the ADITYA tokamak using neural networks', *Nucl. Fusion*, vol. 40, no. 12, p. 1993, Dec. 2000, doi: 10.1088/0029-5515/40/12/304.

[95] D. Wroblewski, G. L. Jahns, and J. A. Leuer, 'Tokamak disruption alarm based on a neural network model of the high- beta limit', *Nucl. Fusion*, vol. 37, no. 6, p. 725, Jun. 1997, doi: 10.1088/0029-5515/37/6/I02.

[96] S. Y. Wang *et al.*, 'Prediction of density limit disruptions on the J-TEXT tokamak', *Plasma Phys. Control. Fusion*, vol. 58, no. 5, p. 055014, Apr. 2016, doi: 10.1088/0741-3335/58/5/055014.

[97] A. Piccione, J. W. Berkery, S. A. Sabbagh, and Y. Andreopoulos, 'Physics-guided machine learning approaches to predict the ideal stability properties of fusion plasmas', *Nucl. Fusion*, vol. 60, no. 4, p. 046033, Mar. 2020, doi: 10.1088/1741-4326/ab7597.

[98] B. Cannas, A. Fanni, E. Marongiu, and P. Sonato, 'Disruption forecasting at JET using neural networks', *Nucl. Fusion*, vol. 44, no. 1, pp. 68–76, Dec. 2003, doi: 10.1088/0029-5515/44/1/008.

[99] B. Cannas, A. Fanni, G. Pautasso, and G. Sias, 'Disruption prediction with adaptive neural networks for ASDEX Upgrade', *Fusion Eng. Des.*, vol. 86, no. 6, pp. 1039–1044, Oct. 2011, doi: 10.1016/j.fusengdes.2011.01.069.

[100] W. Zheng *et al.*, 'Hybrid neural network for density limit disruption prediction and avoidance on J-TEXT tokamak', *Nucl. Fusion*, vol. 58, no. 5, p. 056016, May 2018, doi: 10.1088/1741-4326/aaad17.

[101] Breiman, L., *Classification and Regression Trees*. Routledge, 2017. doi: 10.1201/9781315139470.

[102] R. Aledda, B. Cannas, A. Fanni, G. Sias, and G. Pautasso, 'Mapping of the ASDEX Upgrade Operational Space for Disruption Prediction', *IEEE Trans. Plasma Sci.*, vol. 40, no. 3, pp. 570–576, Mar. 2012, doi: 10.1109/TPS.2011.2174385.

[103] B. Cannas *et al.*, 'Automatic disruption classification in JET with the ITER-like wall', *Plasma Phys. Control. Fusion*, vol. 57, no. 12, p. 125003, Dec. 2015, doi: 10.1088/0741-3335/57/12/125003.

[104] B. H. Guo *et al.*, 'Disruption prediction using a full convolutional neural network on EAST', *Plasma Phys. Control. Fusion*, vol. 63, no. 2, p. 025008, Dec. 2020, doi: 10.1088/1361-6587/abcbab.

[105] R. M. Churchill, B. Tobias, and Y. Zhu, 'Deep convolutional neural networks for multi-scale time-series classification and application to tokamak disruption prediction using raw, high temporal resolution diagnostic data', *Phys. Plasmas*, vol. 27, no. 6, p. 062510, Jun. 2020, doi: 10.1063/1.5144458.

[106] J. Kates-Harbeck, A. Svyatkovskiy, and W. Tang, 'Predicting disruptive instabilities in controlled fusion plasmas through deep learning', *Nature*, vol. 568, no. 7753, Art. no. 7753, Apr. 2019, doi: 10.1038/s41586-019-1116-4.

[107] D. R. Ferreira, P. J. Carvalho, and H. Fernandes, 'Deep Learning for Plasma Tomography and Disruption Prediction From Bolometer Data', *IEEE Trans. Plasma Sci.*, vol. 48, no. 1, pp. 36–45, Jan. 2020, doi: 10.1109/TPS.2019.2947304.

[108] Zong Yu Y. et al., 'An interpretable, transferable and real-time disruption predictor in HL-2A based on deep learning', Mánes, Mánes: 48th EPS Conference on Plasma Physics, 2022.

[109] Z. Yang, F. Xia, X. Song, Z. Gao, S. Wang, and Y. Dong, 'In-depth research on the interpretable disruption predictor in HL-2A', *Nucl. Fusion*, vol. 61, no. 12, p. 126042, Nov. 2021, doi: 10.1088/1741-4326/ac31d8.

[110] J. Vega, A. Murari, S. Dormido-Canto, G. A. Rattá, and M. Gelfusa, 'Disruption prediction with artificial intelligence techniques in tokamak plasmas', *Nat. Phys.*, vol. 18, no. 7, Art. no. 7, Jul. 2022, doi: 10.1038/s41567-022-01602-2.

[111] J. Vega *et al.*, 'Adaptive high learning rate probabilistic disruption predictors from scratch for the next generation of tokamaks', *Nucl. Fusion*, vol. 54, no. 12, p. 123001, Oct. 2014, doi: 10.1088/0029-5515/54/12/123001.

[112] S. Esquembri *et al.*, 'Real-Time Implementation in JET of the SPAD Disruption Predictor Using MARTe', *IEEE Trans. Nucl. Sci.*, vol. 65, no. 2, pp. 836–842, Feb. 2018, doi: 10.1109/TNS.2018.2791719.

[113] Pau, Alessandro, 'Techniques for prediction of disruptions on TOKAMAKS', PhD Dissertation, Centro Interdipartimentale 'Centro Ricerche Fusione', Padova, 2014. Accessed: Nov. 02, 2022. [Online]. Available: https://www.research.unipd.it/handle/11577/3423696

[114] J. M. López *et al.*, 'Implementation of the disruption predictor APODIS in JET real time network using MARTe framework', p. 1.

[115] E. Aymerich *et al.*, 'A statistical approach for the automatic identification of the start of the chain of events leading to the disruptions at JET', *Nucl. Fusion*, vol. 61, no. 3, p. 036013, Feb. 2021, doi: 10.1088/1741-4326/abcb28.

[116] M. J. Leyland *et al.*, 'Edge profile analysis of Joint European Torus (JET) Thomson scattering data: Quantifying the systematic error due to edge localised mode synchronisation', *Rev. Sci. Instrum.*, vol. 87, no. 1, p. 013507, Jan. 2016, doi: 10.1063/1.4939855.

[117] S. N. Gerasimov and et al., 'Locked mode and disruption in JET-ILW', in *46th European Physical Society Conference on Plasma Physics (EPS), Milan, 8-12 July 2019*, Accessed: Jan. 18, 2023. [Online]. Available: https://scientific-publications.ukaea.uk/papers/locked-mode-and-disruption-in-jet-ilw/

[118] J. S. Kim, D. H. Edgell, J. M. Greene, E. J. Strait, and M. S. Chance, 'MHD mode identification of tokamak plasmas from Mirnov signals', *Plasma Phys. Control. Fusion*, vol. 41, no. 11, p. 1399, Nov. 1999, doi: 10.1088/0741-3335/41/11/307.

[119] C. Sozzi *et al.*, 'Early Identification of Disruption Paths for Prevention and Avoidance', presented at the 27th IAEA Fusion Energy Conference (FEC 2018), 2018. Accessed: Oct. 12, 2022. [Online]. Available: https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_3007351

[120] B. Cannas *et al.*, 'Wavelet analysis of Mirnov coils signals for disruption prediction at JET', Mánes, Mánes: 45th EPS Conference on Plasma Physics, 2018.

[121] F. M. Poli, S. E. Sharapov, S. C. Chapman, and J.-E. Contributors, 'Study of the spectral properties of ELM precursors by means of wavelets', *Plasma Phys. Control. Fusion*, vol. 50, no. 9, p. 095009, Jul. 2008, doi: 10.1088/0741-3335/50/9/095009.

[122] D. Alves, R. Coelho, and the J. E. contributors, 'Kalman filter methods for real-time frequency and mode number estimation of MHD activity in tokamak plasmas', *Plasma Phys. Control. Fusion*, vol. 55, no. 10, p. 105003, Aug. 2013, doi: 10.1088/0741-3335/55/10/105003.

[123] D. R. Ferreira, T. A. Martins, P. Rodrigues, and J. E. T. Contributors, 'Explainable deep learning for the analysis of MHD spectrograms in nuclear fusion', *Mach. Learn. Sci. Technol.*, vol. 3, no. 1, p. 015015, Dec. 2021, doi: 10.1088/2632-2153/ac44aa.

[124] A. Bustos, E. Ascasíbar, A. Cappa, and R. Mayo-García, 'Automatic identification of MHD modes in magnetic fluctuation spectrograms using deep learning techniques', *Plasma Phys. Control. Fusion*, vol. 63, no. 9, p. 095001, Jul. 2021, doi: 10.1088/1361-6587/ac08f7.

[125] A. Ultsch and H. P. Siemon, 'Kohonen's self-organizing feature maps for exploratory data analysis', in *Proc. INNC'90, Int. Neural Network Conf. (Paris, France, 9 -13 July 1990)*, Boston, MA: Dordrecht ; Kluwer Academic, 1990, pp. 305–8. [Online]. Available: https://archive.org/details/innc90parisinter0001inte/page/305/mode/2up

[126] S. Dormido-Canto *et al.*, 'Development of an efficient real-time disruption predictor from scratch on JET and implications for ITER', *Nucl. Fusion*, vol. 53, no. 11, p. 113001, Sep. 2013, doi: 10.1088/0029-5515/53/11/113001.

[127] B. Cannas, A. Fanni, A. Murari, A. Pau, G. Sias, and JET EFDA Contributors, 'Automatic disruption classification based on manifold learning for real-time applications on JET', *Nucl. Fusion*, vol. 53, no. 9, p. 093023, Sep. 2013, doi: 10.1088/0029-5515/53/9/093023.

[128] B. Cannas *et al.*, 'Automatic disruption classification in JET with the ITER-like wall', *Plasma Phys. Control. Fusion*, vol. 57, no. 12, p. 125003, Dec. 2015, doi: 10.1088/0741-3335/57/12/125003.

[129] C. I. Stuart *et al.*, 'PETRA: A generalised real-time event detection platform at JET for disruption prediction, avoidance and mitigation', *Fusion Eng. Des.*, vol. 168, p. 112412, Jul. 2021, doi: 10.1016/j.fusengdes.2021.112412.

[130] A. Pau *et al.*, 'A tool for the automatic construction of reliable disruption databases', *Fusion Eng. Des.*, vol. 125, pp. 139–153, 2017.

[131] B. Cannas, A. Fanni, A. Murari, A. Pau, G. Sias, and JET EFDA Contributors, 'Overview of manifold learning techniques for the investigation of disruptions on JET', *Plasma Phys. Control. Fusion*, vol. 56, no. 11, p. 114005, Nov. 2014, doi: 10.1088/0741-3335/56/11/114005.

[132] B. Cannas, A. Fanni, A. Murari, A. Pau, G. Sias, and the JET EFDA Contributors, 'Manifold learning to interpret JET high-dimensional operational space', *Plasma Phys. Control. Fusion*, vol. 55, no. 4, p. 045006, Apr. 2013, doi: 10.1088/0741-3335/55/4/045006.

[133] P. C. de Vries *et al.*, 'The impact of the ITER-like wall at JET on disruptions', *Plasma Phys. Control. Fusion*, vol. 54, no. 12, p. 124032, Nov. 2012, doi: 10.1088/0741-3335/54/12/124032.

[134] M. F. Møller, 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Netw.*, vol. 6, no. 4, pp. 525–533, Jan. 1993, doi: 10.1016/S0893-6080(05)80056-5.

[135] S. Pouyanfar *et al.*, 'A Survey on Deep Learning: Algorithms, Techniques, and Applications', *ACM Comput. Surv.*, vol. 51, no. 5, p. 92:1-92:36, Sep. 2018, doi: 10.1145/3234150.

[136] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, 'FreezeOut: Accelerate Training by Progressively Freezing Layers', *ArXiv170604983 Cs Stat*, Jun. 2017, Accessed: Dec. 13, 2021. [Online]. Available: http://arxiv.org/abs/1706.04983

[137] G. Pucella *et al.*, 'Onset of tearing modes in plasma termination on JET: the role of temperature hollowing and edge cooling', *Nucl. Fusion*, vol. 61, no. 4, p. 046020, Mar. 2021, doi: 10.1088/1741-4326/abe3c7.

[138] J. Garcia *et al.*, 'Integrated Scenario Development at JET for DT Operation and ITER Risk Mitigation', 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

[139] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, 'Multimodal Machine Learning: A Survey and Taxonomy', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019, doi: 10.1109/TPAMI.2018.2798607.

[140] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, 'Early vs Late Fusion in Multimodal Convolutional Neural Networks', in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, Jul. 2020, pp. 1–6. doi: 10.23919/FUSION45008.2020.9190246.

[141] S.-C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, and M. P. Lungren, 'Fusion of medical imaging and electronic health records using deep learning: a

systematic review and implementation guidelines', *Npj Digit. Med.*, vol. 3, no. 1, pp. 1–9, Oct. 2020, doi: 10.1038/s41746-020-00341-z.

[142] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, 'Early versus late fusion in semantic video analysis', in *Proceedings of the 13th annual ACM international conference on Multimedia*, in MULTIMEDIA '05. New York, NY, USA: Association for Computing Machinery, Nov. 2005, pp. 399–402. doi: 10.1145/1101149.1101236.

[143] E. Aymerich *et al.*, 'Disruption prediction at JET through Deep Convolutional Neural Networks using spatiotemporal information from plasma profiles', *Nucl. Fusion*, vol. 62, p. 066005, 2022, doi: 10.1088/1741-4326/ac525e.

[144] S. Dormido-Canto *et al.*, 'Development of an efficient real-time disruption predictor from scratch on JET and implications for ITER', *Nucl. Fusion*, vol. 53, no. 11, p. 113001, Sep. 2013, doi: 10.1088/0029-5515/53/11/113001.

[145] K. J. Montes *et al.*, 'Machine learning for disruption warnings on Alcator C-Mod, DIII-D, and EAST', *Nucl. Fusion*, vol. 59, no. 9, p. 096015, Jul. 2019, doi: 10.1088/1741-4326/ab1df4.

[146] D. R. Ferreira, P. J. Carvalho, C. Sozzi, P. J. Lomas, and J. E. T. Contributors, 'Deep Learning for the Analysis of Disruption Precursors based on Plasma Tomography', *ArXiv200902708 Phys.*, Sep. 2020, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/2009.02708

[147] S. Carcangiu, A. Fanni, and A. Montisci, 'Multiobjective Tabu Search Algorithms for Optimal Design of Electromagnetic Devices', *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 970–973, Jun. 2008, doi: 10.1109/TMAG.2007.916336.

[148] M. Merola *et al.*, 'ITER plasma-facing components', *Fusion Eng. Des.*, vol. 85, no. 10, pp. 2312–2322, Dec. 2010, doi: 10.1016/j.fusengdes.2010.09.013.

[149] D. F. Valcárcel *et al.*, 'The JET real-time plasma-wall load monitoring system', *Fusion Eng. Des.*, vol. 89, no. 3, pp. 243–258, Mar. 2014, doi: 10.1016/j.fusengdes.2013.10.010.

[150] A. Huber *et al.*, 'Real-time protection of the JET ITER-like wall based on near infrared imaging diagnostic systems', *Nucl. Fusion*, vol. 58, no. 10, p. 106021, Aug. 2018, doi: 10.1088/1741-4326/aad481.

[151] B. Sieglin *et al.*, 'Real time capable infrared thermography for ASDEX Upgrade', *Rev Sci Instrum*, p. 7, 2015.

[152] M. Jakubowski *et al.*, 'Infrared imaging systems for wall protection in the W7-X stellarator (invited)', *Rev. Sci. Instrum.*, vol. 89, no. 10, p. 10E116, Oct. 2018, doi: 10.1063/1.5038634.

[153] A. Huber *et al.*, 'The near infrared imaging system for the real-time protection of the JET ITER-like wall', *Phys. Scr.*, vol. 2017, no. T170, p. 014027, Oct. 2017, doi: 10.1088/1402-4896/aa8a14.

[154] A. Herrmann, R. Drube, T. Lunt, and P. de Marné, 'Real-time protection of in-vessel components in ASDEX Upgrade', *Fusion Eng. Des.*, vol. 86, no. 6, pp. 530–534, Oct. 2011, doi: 10.1016/j.fusengdes.2011.02.037.

[155] E. Grelier, R. Mitteau, and V. Moncada, 'Deep learning and image processing for the automated analysis of thermal events on the first wall and divertor of fusion reactors', *Plasma Phys. Control. Fusion*, vol. 64, no. 10, p. 104010, Sep. 2022, doi: 10.1088/1361-6587/ac9015.

[156] R. C. Wolf *et al.*, 'Performance of Wendelstein 7-X stellarator plasmas during the first divertor operation phase', *Phys. Plasmas*, vol. 26, no. 8, p. 082504, Aug. 2019, doi: 10.1063/1.5098761.

[157] A. Puig Sitjes *et al.*, 'Wendelstein 7-X Near Real-Time Image Diagnostic System for Plasma-Facing Components Protection', *Fusion Sci. Technol.*, vol. 74, no. 1–2, pp. 116–124, Aug. 2018, doi: 10.1080/15361055.2017.1396860.

[158] R. Stadler *et al.*, 'The in-vessel components of the experiment WENDELSTEIN 7-X', *Fusion Eng. Des.*, vol. 84, no. 2, pp. 305–308, Jun. 2009, doi: 10.1016/j.fusengdes.2008.11.067.

[159] J. Boscary *et al.*, 'Actively Water-Cooled Plasma Facing Components of the Wendelstein 7-X Stellarator', *Fusion Sci. Technol.*, vol. 64, no. 2, pp. 263–268, Aug. 2013, doi: 10.13182/FST12-499.

[160] F. Pisano *et al.*, 'Towards a new image processing system at Wendelstein 7-X: From spatial calibration to characterization of thermal events', *Rev. Sci. Instrum.*, vol. 89, no. 12, p. 123503, Dec. 2018, doi: 10.1063/1.5045560.

[161] A. Herrmann, B. Sieglin, M. Faitsch, and A. U. Team, 'Surface Temperature Measurement of In-Vessel Components and Its Real-Time Validation', *Fusion Sci. Technol.*, vol. 69, no. 3, pp. 569–579, May 2016, doi: 10.13182/FST15-187.

[162] A. Ali *et al.*, 'Initial results from the hotspot detection scheme for protection of plasma facing components in Wendelstein 7-X', *Nucl. Mater. Energy*, vol. 19, pp. 335–339, May 2019, doi: 10.1016/j.nme.2019.03.006.

[163] M. Endler *et al.*, 'Managing leading edges during assembly of the Wendelstein 7-X divertor', *Plasma Phys. Control. Fusion*, vol. 61, no. 2, p. 025004, Nov. 2018, doi: 10.1088/1361-6587/aaef52.

[164] Clemente Bonjour, Rocco, 'Detection and Classification of Thermal Events in the Wendelstein 7-X', Master Thesis, Universitat Politecnica de Catalunya, 2020.

[165] A. Puig Sitjes *et al.*, 'Strategy for the real-time detection of thermal events on the plasma facing components of Wendelstein 7-X', presented at the 31st Symposium on Fusion Technology (SOFT 2020), 2020. Accessed: May 15, 2022. [Online]. Available: https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_3267208

[166] B. Jabłoński *et al.*, 'Evaluation of NVIDIA Xavier NX Platform for Real-Time Image Processing for Plasma Diagnostics', *Energies*, vol. 15, no. 6, Art. no. 6, Jan. 2022, doi: 10.3390/en15062088.

[167] F. Pisano *et al.*, 'Learning control coil currents from heat-flux images using convolutional neural networks at Wendelstein 7-X', *Plasma Phys. Control. Fusion*, Nov. 2020, doi: 10.1088/1361-6587/abce19.

[168] M. Blatzheim, D. Böckenhoff, and the W. 7-X. Team, 'Neural network regression approaches to reconstruct properties of magnetic configuration from Wendelstein 7-X modeled heat load patterns', *Nucl. Fusion*, vol. 59, no. 12, p. 126029, Oct. 2019, doi: 10.1088/1741-4326/ab4123.

[169] D. Böckenhoff *et al.*, 'Reconstruction of magnetic configurations in W7-X using artificial neural networks', *Nucl. Fusion*, vol. 58, no. 5, p. 056009, Mar. 2018, doi: 10.1088/1741-4326/aab22d.

[170] A. Herrmann *et al.*, 'Energy flux to the ASDEX-Upgrade diverter plates determined by thermography and calorimetry', *Plasma Phys. Control. Fusion*, vol. 37, no. 1, pp. 17–29, Jan. 1995, doi: 10.1088/0741-3335/37/1/002.

[171] Lam, S, Pitrou, A, and Seibert, S, 'Numba: A llvm-based python jit compiler', in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 1–6.

[172] J. Snoek, H. Larochelle, and R. P. Adams, 'Practical Bayesian Optimization of Machine Learning Algorithms', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: May 16, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html

[173] M. E. Tipping and C. M. Bishop, 'Probabilistic Principal Component Analysis', *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999, doi: 10.1111/1467-9868.00196.

Acknowledgements

The three years of the PhD were filled with events, mostly positive but with some difficult ones. Nevertheless, I am extremely grateful for having completed this experience, thanks to the help of several people who contributed to my personal and professional growth.

First and foremost, I would like to thank **Prof. Alessandra Fanni**, who supervised me and motivated me to complete this challenging PhD work. She guided me through the complex field of academic research and helped me overcome difficult situations and deadlines. I am also grateful to **Dr. Fabio Pisano**, who pushed me beyond my comfort zone and was always available to answer my questions. This thesis would not have been possible without their support and invaluable personal and professional advice.

Similarly, I would like to express my gratitude to **Prof. Giuliana Sias**, with whom I worked side by side during most of the PhD. She advised and encouraged me to complete the work, and supported me during difficult times, even outside of work-related issues. I am thankful to **Prof. Barbara Cannas** for her collaboration at the end of the PhD and for her invaluable support during my time in Greifswald. I would also like to thank **Prof. Augusto Montisci** for his support in completing the MHD coursework and **Sara Carcangiu** and **Manuela Pasella** for making the office time lighter with their company and conversations. Finally, I would like to thank **Massimiliano Lacquaniti**, with whom I shared the PhD experience, despite being physically apart most of the time. I found myself aligned with the whole group's approach to work, which was both passionate and humble.

I thank **Dr. Marcin Jakubowski** and **Dr. Daniel Böckenhoff** for supporting me during my visit in Greifswald and thanks to the **DAAD association** for making this international experience possible.

I am also grateful for my special friends. First of all, **Alice**, **Ludovica**, and **Emma** were there for me when nothing seemed to work anymore and talked me out of a bottomless pit of grief. I am grateful for their kind words and advice at that time and even now. Moreover, I would like to thank **Filippo** and **Nicoletta**, who were here in Cagliari when I felt alone, and **Yari** and **Davide**, who had to tolerate me when I needed to vent. You really are my closest friends, and I also owe my success in submitting this work to you.

Finally, I would like to extend my heartfelt appreciation to my parents **Sandra** and **Andrea,** and my brother **Marco** for their support and love at all times.

I am humbled by the unwavering support and encouragement from my friends, family, and colleagues during my PhD journey. It would not have been the same without your help.