Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

Laboratorio Nacional de Fusión

Universidad Complutense de Madrid

Facultad de Informática

Universidad Nacional de Educación a Distancia

Escuela Técnica Superior de Ingeniería Informática

Automatic Recognition and Analysis of Collapsing Discharges in NBI Plasmas in the TJ-II Stellarator

Luis López Sánchez

Directora: Belén López-Miranda
Co-director: Jesús Antonio Vega Sánchez
Tutor: Sebastián Dormido Canto

Trabajo de Fin de Máster

Máster Universitario
en Ingeniería de Sistemas y Control

Curso 2024-2025 / Convocatoria de Junio

Agradecimientos

En primer lugar me gustaría dedicar estos agradecimientos a mis tutores Belén, Jesús y Sebastián por todo el esfuerzo que han puesto en mí y en mi amigo David. Hemos aprendido mucho con vosotros, pero lo más importante que nos llevamos es haber conocido a personas tan maravillosas que lo han dado todo por nosotros.
Agradecimientos a mi amigo David, por supuesto, que es mi compañero de batallas, y sobre todo a mi familia y en especial a mi novia Arantxa, que ha sido la persona que me ha apoyado en el día a día durante todo el transcurso del máster y, si no fuera por su apoyo, no podría haber finalizado el máster en un año mientras lo compaginaba con el trabajo.
¡Gracias a todos, sois increíbles!

Resumen

El objetivo de este trabajo final de máster es procesar las señales de calentamiento, energía y radiación de descargas de plasmas de fusión del stellarator TJ-II mediante técnicas de procesamiento de señales y de Machine Learning, con el fin de determinar si se ha producido un colapso en el TJ-II. En este tipo de colapsos, radiativos, entran impurezas en el plasma, que empiezan a radiar, se pierde energía, se enfría el plasma y se produce el colapso. Para lograr el objetivo de este trabajo, se ha desarrollado una aplicación software en MATLAB llamada PCAI Tool (Plasma Collapse Analysis Identifier Tool), que permite a los usuarios procesar de forma automática miles de descargas de manera intuitiva.

Palabras clave: TJ-II, colapso, descarga, densidad, energía, Butterworth, transformada Haar, SVM.

Abstract

The purpose of this final master's thesis is to process the heating, energy, and radiation signals from fusion plasma discharges in the TJ-II stellarator by means of state-of-the-art signal processing and Machine Learning techniques in order to determine whether a collapse has occurred in the TJ-II. In this type of collapses, radiative ones, impurities enter the plasma, start to radiate, energy is lost, the plasma cools, and the collapse occurs. To accomplish the objective of the thesis, a MATLAB software application called PCAI Tool (Plasma Collapse Analysis Identifier Tool) has been developed, which enables users to process automatically thousands of shots intuitively.

Keywords: TJ-II, collapse, shot, density, energy, Butterworth filter, wavelet transform, SVM.

Glossary

DWT Discrete Wavelet Transform.

ECRH Electron Cyclotron Resonance Heating.

GUI Graphical User Interface.

NBI Neutral Beam Injection.

PCAI Tool Plasma Collapse Analysis Identifier Tool.

SVM Support Vector Machine.

Contents

Glossary **ix**

1 Introduction **1**
1.1 Context . . . 1
1.2 Motivation . . . 6
1.3 Proposal and Objectives . . . 6
1.4 Document Structure . . . 6

2 Foundations and Background **9**
2.1 Fusion Devices . . . 9
2.1.1 Tokamak . . . 9
2.1.2 Stellarator . . . 10
2.2 Machine Learning Applied to Nuclear Fusion . . . 12
2.3 Scope . . . 13

3 Algorithms and Software Architecture **15**
3.1 Signal Processing . . . 15
3.1.1 Determination of the Collapse Time . . . 15
3.1.2 Determination of the NBI End Time . . . 21
3.1.3 Density Signal Reconstruction . . . 22
3.2 Support Vector Machines . . . 27
3.2.1 Discharge Classification . . . 29
3.2.2 Density Classification . . . 29
3.3 PCAI Tool . . . 30
3.3.1 Software Architecture . . . 30

4 Results **33**
4.1 Signal Processing Results . . . 33
4.1.1 Collapse Time Algorithms Comparison . . . 33
4.1.2 NBI End Time Algorithm Results . . . 36
4.1.3 Density Signal Reconstruction Algorithm Results . . . 37

4.2 PCAI Tool Use Case . . . 38
4.3 SVM Classification Results . . . 49
4.3.1 Confusion matrix and derived metrics . . . 49
4.3.2 Shots Classification Results . . . 50
4.3.3 Density Classification Results . . . 50
4.4 Analysis . . . 52
4.4.1 Anticipation Time Analysis . . . 52
4.4.2 Density Analysis . . . 52

5 Conclusions and Future Work **55**
5.1 Conclusions . . . 55
5.2 Future Work . . . 55

Bibliography and References **57**

A PCAI Tool User Manual **59**
A.1 Installation Guide . . . 59
A.2 User Guide . . . 63
A.2.1 Main Screen - Signal/Download . . . 64
A.2.2 Signal/First Assessment . . . 67
A.2.3 Signal/Download Additional Signals . . . 73
A.2.4 Labels . . . 75
A.2.5 Training . . . 77
A.2.6 Prediction/Known Labels . . . 80
A.2.7 Prediction/Unknown Labels . . . 82
A.2.8 Analysis/Density classification (manual) . . . 84
A.2.9 Analysis/Density classification (model creation) . . . 86
A.2.10 Analysis/Classification (known labels) . . . 89
A.2.11 Analysis/Classification (unknown labels) . . . 91
A.2.12 Analysis/Anticipation time . . . 93
A.2.13 Analysis/Density histograms . . . 94

List of Figures

1.1 D–T fusion reaction. . . . 1
1.2 3D model of the TJ-II stellarator. . . . 2
1.3 TJ-II heating systems . . . 3
1.4 Critical TJ-II signals for collapse diagnosis. . . . 5
1.5 Collapsing vs non-collapsing discharges. . . . 5

2.1 Joint European Torus (JET). . . . 10
2.2 TJ-II. . . . 11

3.1 Signal of energy contained in the plasma. . . . 16
3.2 Energy signal with noise and decrease areas. . . . 18
3.3 Butterworth filters family. . . . 20
3.4 Anomalous density signal plot. . . . 23
3.5 Anomalous vs regular density signals. . . . 30
3.6 PCAI Tool architecture. . . . 31

4.1 Energy signal with attenuated noise and decrease areas. . . . 34
4.2 Integral derivative algorithm decay time. . . . 34
4.3 DWT decay time. . . . 35
4.4 NBIs end times. . . . 36
4.5 Density signal reconstruction algorithm comparison. . . . 37
4.6 PCAI Tool Download screen. . . . 39
4.7 PCAI Tool First Assessment screen. . . . 40
4.8 PCAI Tool Download Additional Signals screen. . . . 41
4.9 PCAI Tool Labels screen . . . 42
4.10 PCAI Tool Training screen . . . 43
4.11 PCAI Tool Known Labels screen. . . . 44
4.12 PCAI Tool Unknown Labels screen. . . . 45
4.13 PCAI Tool Density classification (manual) screen. . . . 46
4.14 PCAI Tool Anticipation Time screen. . . . 47
4.15 PCAI Tool Density Histograms screen. . . . 48
4.16 Collapsing discharges time difference histogram. . . . 52

4.17 Density histograms. 53

A.1 PCAI Tool web installer executable. 59
A.2 Installation wizard welcome screen. 60
A.3 Installation directory and desktop shortcut screen. 60
A.4 Begin installation screen. 61
A.5 Installation completed screen. 61
A.6 PCAI Tool desktop shortcut. 62
A.7 Initial signal selection dialog wizard. 62
A.8 Main window log. 63
A.9 PCAI Tool desktop shortcut. 63
A.10 PCAI Tool main screen. 64
A.11 Discharge range confirmation. 65
A.12 Save log to ".txt" file. 65
A.13 Save request. 66
A.14 First Assessment Screen. 67
A.15 "Temporary Shots List" filter options. 68
A.16 "Temporary Shots List" save options. 68
A.17 NBI signals check warning. 68
A.18 Shot overwrite warning. 69
A.19 Delete temporary shots warning. 69
A.20 Delete all temporary shots warning. 69
A.21 Empty temporary shots list warning. 70
A.22 "Definitive Shots List" options. 70
A.23 Delete definitive shot warning. 70
A.24 First Assessment Screen plot and characteristics table. 71
A.25 Download Additional Signals screen. 73
A.26 Density signal overwrite warning. 74
A.27 Labels screen. 75
A.28 Selection warning. 76
A.29 Training screen. 77
A.30 Characteristics vector options. 78
A.31 Training options. 78
A.32 SVM training options. 79
A.33 Known Labels screen. 80
A.34 Prediction sets management options. 81
A.35 Prediction results options. 81
A.36 Unknown Labels screen. 82
A.37 Prediction sets management options. 83

A.38 Prediction results options. 83
A.39 Labels screen. 84
A.40 Selection warning. 85
A.41 Density classification (model creation) screen. 86
A.42 Characteristics vector options. 87
A.43 Training options. 87
A.44 SVM training options. 88
A.45 Classification (known labels). 89
A.46 Prediction sets management options. 90
A.47 Prediction results options. 90
A.48 Classification (unknown labels) screen. 91
A.49 Prediction sets management options. 92
A.50 Prediction results options. 92
A.51 Anticipation Time Screen. 93
A.52 Density histograms screen. 94

List of Tables

1.1 Critical TJ-II signals for collapse diagnosis 4

2.1 Main characteristics of TJ-II 11

3.1 Butterworth Denominator Polynomials 19
3.2 Butterworth filter parameters 19
3.3 NBI End Time Parameters 22
3.4 Density Signal Reconstruction Algorithm Threshold 27

4.1 Energy signal decay time and value difference between tables 35
4.2 NBIs end times 36
4.3 Generic form of a confusion matrix. 49
4.4 SVM metrics vs. signal, sample size and decomposition 51
4.5 Density classifier performance by wavelet coefficient and decomposition level 51

A.1 PC Specifications 59
A.2 Toolbar options available in MATLAB figure windows 71
A.3 Definitive shots characteristics 72

Chapter 1

Introduction

This chapter outlines the motivation behind this master's thesis and identifies the objectives that enabled the successful realization of the project, i.e., the development of a MATLAB-based graphical user interface (GUI) for the automatic recognition of collapsing discharges from the database of the TJ-II stellarator. The chapter ends with a detailed description of the structure of the document.

1.1 Context

Nuclear fusion represents one of the most promising avenues for sustainable energy production [1], which is the process of combining light atomic nuclei into heavier ones under extreme conditions, releasing energy according to the mass–energy equivalence. For future fusion reactors, the most frequently studied reaction is deuterium–tritium (D–T), where two heavy hydrogen isotopes fuse to form a helium nucleus and a neutron, releasing approximately 17.6 MeV of energy, as shown in Equation 1.1 and represented in Figure 1.1.

$$\mathrm{D} + \mathrm{T} \longrightarrow {}^{4}\mathrm{He}\,(3.5\,\mathrm{MeV}) + {}^{1}\mathrm{n}\,(14.1\,\mathrm{MeV}) \tag{1.1}$$

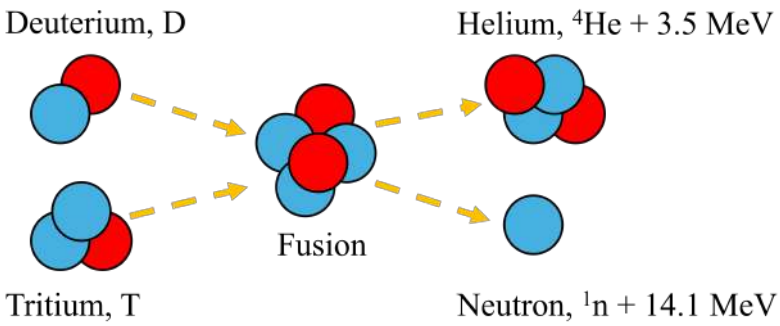

Figure 1.1: D–T fusion reaction.

Broadly speaking, two approaches have been developed to confine and ignite fusion plasmas: magnetic confinement and inertial confinement. In inertial confinement, powerful lasers or implosion devices compress and heat tiny fuel pellets for a very short time. In magnetic confinement, strong magnetic fields are used to confine the hot, ionized gas (plasma) within a vacuum chamber, so that it does not touch material walls. Over the past seven decades, magnetic confinement has been explored through various device concepts, among which the tokamak [2] and the stellarator [3] have emerged as the leading designs. Both approaches are explained in depth in Chapter 2. However, this work focuses in the TJ-II stellarator (see Figure 1.2), the experimental fusion device from the Laboratorio Nacional de Fusión, located in Madrid, Spain.

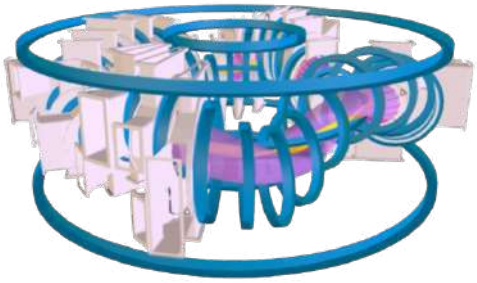

Figure 1.2: 3D model of the TJ-II stellarator.

The fundamental requirement for fusion ignition is commonly expressed through the Lawson criterion [4], which states that the product of plasma density (n), temperature (T), and energy confinement time (τ_E) must exceed a critical threshold:

$$n \cdot T \cdot \tau_E > 5 \times 10^{21}\,\mathrm{m}^{-3}\,\mathrm{keV\,s} \tag{1.2}$$

If we heat the nuclei to about 100 million K, they gain enough kinetic energy to fuse, and at that temperature deuterium and tritium stop behaving as gases and instead form a highly ionized, yet overall electrically neutral, *plasma*.

Within fusion devices, discharges (also known as shots) occur, which can be defined as *"the time during which the plasma is born, lives, and dies."*.

Maintaining the conditions explained before is difficult and collapses may happen. A *plasma collapse* corresponds to a sudden and severe deterioration of the key parameters mentioned in Lawson criterion 1.2. During a collapse event:

- The **plasma temperature**, T, rapidly decreases.
- The **plasma density**, n, lowers.
- The **energy confinement time** significantly shortens as energy escapes the plasma, decreasing τ_E.

Diagnosis and prediction of collapse events in fusion plasmas remains a critical challenge in the field. In stellarators, understanding and avoiding plasma collapses is crucial to ensure stable operation, protect internal components, and maintain good energy confinement. Collapses, often driven by impurity accumulation and radiative losses, can severely degrade plasma performance, interrupt experiments, and reduce overall efficiency. Preventing these events is essential for the longevity of the device and for achieving reliable, continuous plasma operation.

Focusing on TJ-II, individual discharges typically last around 0.25 s and recur every 7–8 minutes. Hydrogen is most often used as the working gas, although deuterium or helium may also be employed. The plasma is generally generated and sustained by two heating systems, which are illustrated in Figure 1.3 and described in the following paragraphs.

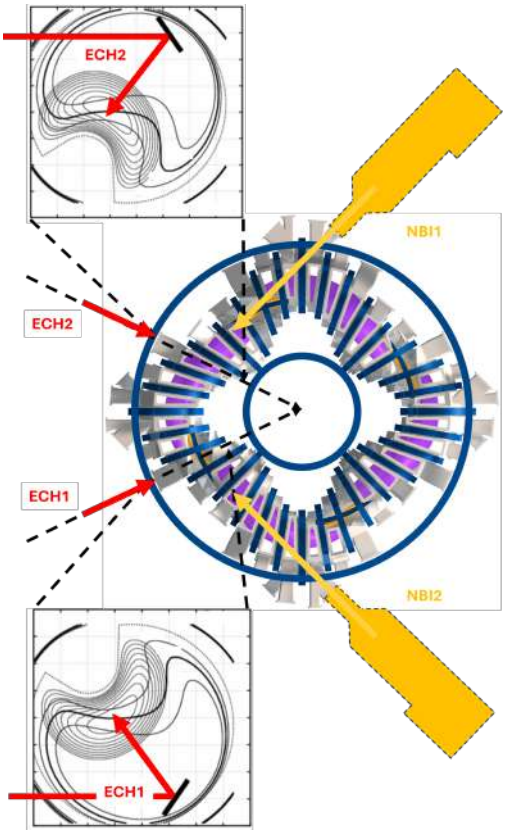

Figure 1.3: TJ-II heating systems, which consists of two gyrotrons, ECH1 and ECH2, the power is transmitted to the plasma by two quasi-optical transmission lines. And two neutral beam injectors, NBI1, the co-injector, and NBI2, counter-injector.

The TJ-II device employs two primary heating mechanisms:

1. *Electron Cyclotron Resonance Heating (ECRH).* In this scheme, microwaves at the electron cyclotron frequency are injected into the plasma, in a similar way to how a kitchen microwave operates, so that microwaves resonate with the plasma and it is heated. TJ-II is equipped with two gyrotrons, each capable of delivering approximately

200–250 kW performing at 53.2 GHz. Under these conditions, central electron densities up to $n_e(0) \leq 1.7 \times 10^{19}\,\mathrm{m}^{-3}$ and temperatures approaching $T_e(0) \leq 2\,\mathrm{keV}$ $(\approx 2.3 \times 10^7\,^\circ\mathrm{C})$ can be achieved.

2. *Neutral Beam Injection (NBI).* First, a hydrogen gas flow is ionized and then accelerated to 30 keV; these fast protons are subsequently neutralized so they can penetrate the magnetic confinement. TJ-II uses two neutral beam sources (co-injection and counter-injection), each providing on the order of 500 kW for pulses of about 300 ms. The momentum transfer from the injected atoms to the background plasma heats the ions, achieving typical core ion temperatures up to $\leq 120\,\mathrm{eV}$ $(\approx 1.4 \times 10^6\,\mathrm{K})$.

In the TJ-II, radiative collapses happen because impurities enter the plasma, start to radiate, energy is lost, the plasma cools, and the collapse occurs. To better understand this process and diagnose collapses, TJ-II signals are measured and analyzed. There are several types of signals to analyze from the TJ-II, but the critic ones for this process are presented in Table 1.1 and displayed in Figure 1.4.

Table 1.1: Critical TJ-II signals for collapse diagnosis.

Signal	**Unit**	**Description**
`W_b4_corr_`	kJ	Energy contained in the plasma; used to detect collapses.
`ACTON275`	a.u.	Radiation monitor for C V ion emission.
`Density2_`	10^{19} m^{-3}	Line-averaged electron density from interferometry.
`IACCEL1`	a.u.	Accelerator current of Neutral Beam Injection 1, NBI1 (co-injection). It can determine the end of a discharge.
`IACCEL2`	a.u.	Accelerator current of Neutral Beam Injection 2, NBI2 (counter-injection). It can determine the end of a discharge.

The plasma energy signal (`W_b4_corr_`) is well suited for collapse detection because it directly reflects the overall confinement performance. A collapse typically manifests as a sudden drop in the stored energy, making it a clear and physically meaningful indicator of the event's timing.

Finally, a criteria for differentiating between collapsing and non-collapsing discharges can be established taking into account these signals, which is that a shot is considered collapsing if its energy signal (also ACTON 275) decays before the shot end time defined by both NBIs. In Figure 1.5, a clear case of a collapsing discharge and a non-collapsing discharge are shown. The criteria to classify whether a discharge is collapsing or not is that the decay time of the energy signal happens earlier the shot end time.

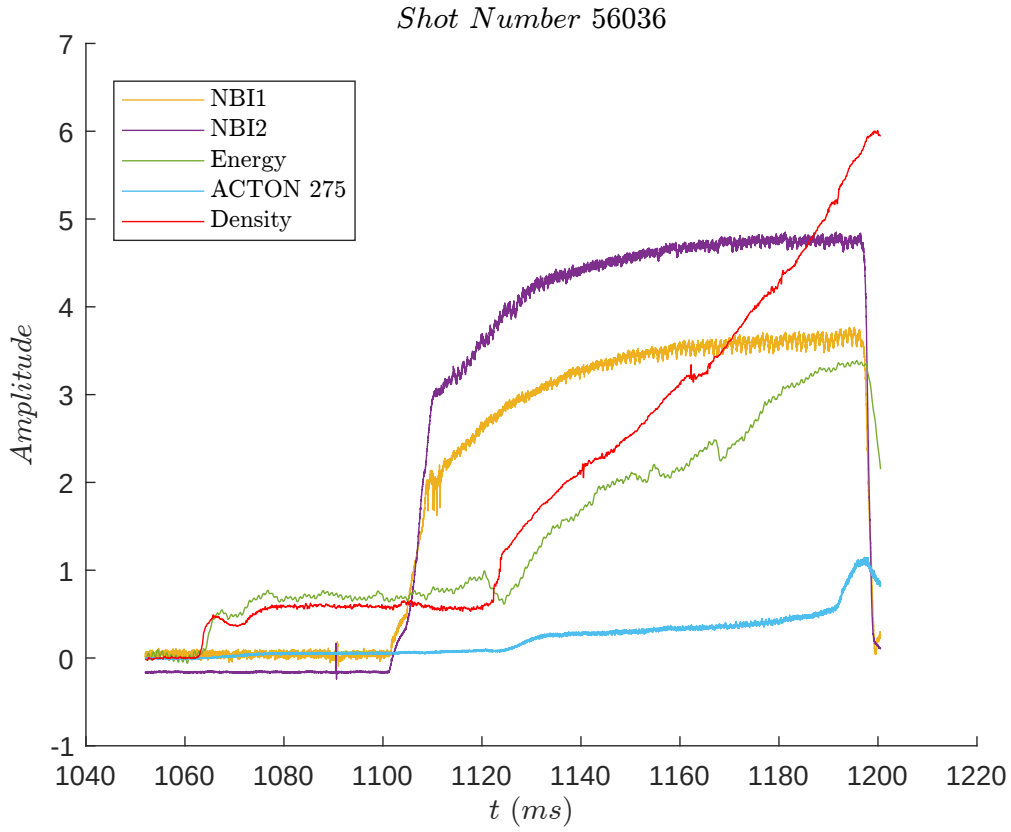

Figure 1.4: Critical TJ-II signals for collapse diagnosis.

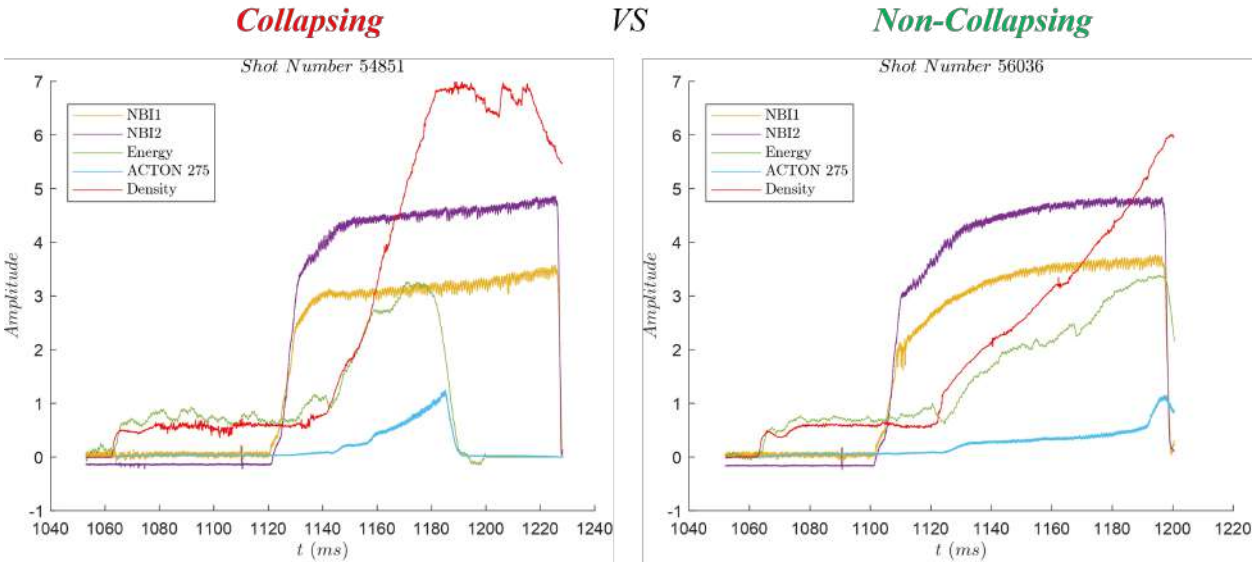

Figure 1.5: Collapsing vs non-collapsing discharges. The collapsing shot's energy signal decays before the end of the heating, determined by both NBIs. In the non-collapsing case, the energy signal has not finished its decay before the end of the heating.

1.2 Motivation

The motivation behind this Master's Final Thesis is to enhance plasma diagnostics in the TJ-II stellarator by developing a MATLAB-based GUI that integrates advanced signal processing with a discharge classifier based on Support Vector Machines (SVM). This tool allows users to interactively modify feature vectors, train, test and predict with SVM models.

The software application is called Plasma Collapse Analysis Identifier Tool (PCAI Tool), before its implementation collapsing discharges classification was made manually by a human agent, but thanks to development of this tool, plasma diagnosis is now completely automatized and is processing a database of 60000 discharges.

1.3 Proposal and Objectives

The aim of this work is to develop a MATLAB-based GUI which automatically processes and classifies TJ-II measured signals to determine whether a collapse has occurred. This goal is achieved by addressing the following objectives:

- Develop a MATLAB-based graphical user interface (GUI) for processing and analyzing plasma signals from the TJ-II stellarator.
- Design and implement advanced signal processing methods to extract relevant features from measured data.
- Generate a database capturing plasma density values at collapse times.
- Integrate an SVM classifier into the GUI for automatic classification of plasma discharges, particularly to identify collapse events.
- Enable the selection of different feature vectors within the GUI to optimize the SVM performance.
- Evaluate the performance of the SVM classifier.

1.4 Document Structure

- **Glossary:** A comprehensive list of technical terms and abbreviations used throughout the thesis.
- **Chapter 1: Introduction**

  1.1 **Context:** Presents the context of this work.

  1.2 **Motivation:** Outlines the relevance of fusion energy and the challenges in plasma diagnostics, providing the context for this research.

1.3 **Proposal and Objectives:** Defines the project's goals, including the development of a MATLAB-based GUI for processing and classifying TJ-II discharges.

1.4 **Document Structure:** Summarizes the organization and content of the subsequent chapters.

- **Chapter 2: Foundations and Background**

  2.1 **Fusion Devices:** Reviews fundamental fusion principles and its history.

  2.2 **Machine Learning:** Discusses current advances in machine learning techniques relevant to plasma diagnostics and classification.

- **Chapter 3: Algorithms and Software Architecture**

  3.1 **Signal Processing:** Describes the techniques for processing TJ-II signals, including:

    3.1.1 Determination of the Collapse Time.

    3.1.2 Determination of the NBI End Time.

    3.1.3 Density Signal Reconstruction.

  3.2 **SVM:** Explains the implementation of supervised classification systems using Support Vector Machines, covering:

    3.2.1 Shot Classification.

    3.2.1 Density Classification.

  3.3 **PCAI Tool:** Details the development of the Plasma Collapse Analysis Identifier Tool in MATLAB.

- **Chapter 4: Results** Presents and analyzes key findings from the application of the developed methods, with performance metrics used to assess the SVM classifier.

- **Chapter 5: Conclusions and Future Work** Summarizes the contributions of the research and outlines directions for future improvements.

- **Bibliography and References**
  Lists all cited works and additional resources that informed the research.

- **Appendix A: PCAI Tool User Manual**
  Presents detailed instructions about the installation and the use of the PCAI Tool.

Chapter 2

Foundations and Background

This chapter presents a comprehensive background in nuclear fusion research and machine learning applications for plasma diagnostics. Here, fundamental principles of fusion, key experimental breakthroughs, and innovative diagnostic methods used in fusion devices are introduced. Furthermore, it explores state-of-the-art techniques, including SVM-based classification, that are integral to modern plasma analysis. And finally, the scope and context of this work are explained.

2.1 Fusion Devices

As mentioned in Section 1.1 magnetic confinement has been explored through various device concepts, among which the tokamak and the stellarator have emerged as the leading designs. In this section a historical context of both devices is provided.

2.1.1 Tokamak

Tokamaks use a combination of external magnetic coils and a strong electric current driven through the plasma to create helical magnetic field lines that confine the plasma in a torus. The tokamak concept was first proposed in the early 1950s by Soviet physicists Andrei Sakharov and Igor Tamm [5]. The name tokamak originates from a Russian acronym for "toroidal chamber with magnetic coils".

Later in the 1980s, in Europe, the pinnacle was the Joint European Torus (JET) in Culham, UK, which became the world's largest operational tokamak when it achieved first plasma in 1983 [6], Figure 2.1. Nowadays, it is inoperative; however, relevant results for the development of future fusion reactors are still being obtained from the analysis of its data.

The latest and most ambitious tokamak to date is the ITER (that means "the way" in Latin), whose goal is to demonstrate the scientific and technological feasibility of fusion energy on a reactor scale. Its construction started in 2010 [7], and is still under development.

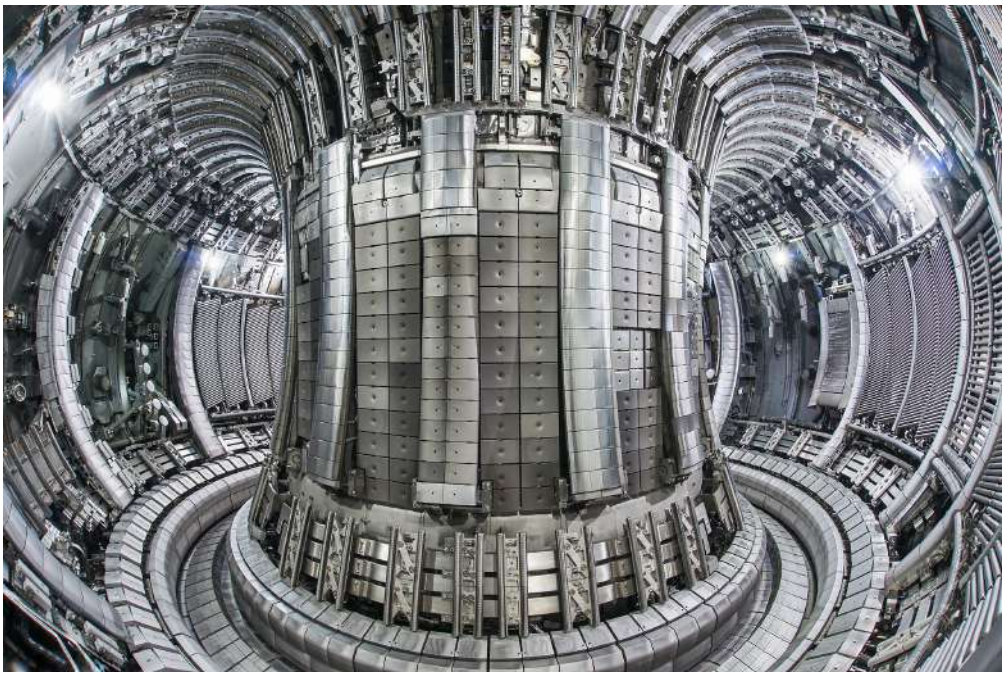

Figure 2.1: Joint European Torus (JET).

2.1.2 Stellarator

Unlike a tokamak, a stellarator confines plasma solely with complex arrangements of external coils, without relying on a large current flowing in the plasma itself. This eliminates the pulsed-operation constraint and certain instability risks of tokamaks.

The stellarator was invented in 1951 by Lyman Spitzer at Princeton University [3]. By 1953 Spitzer's team had built the Model A stellarator, which succeeded in confining a hot plasma [8].

Later, in 1997, the stellarator TJ-II Figure 2.2, a medium-scale flexible-heliac device at the Laboratorio Nacional de Fusión (LNF), at CIEMAT in Madrid, began its operation [9]. By adjusting the electric currents in its circular and helical coils [10], one can alter the magnetic configuration, the orientation of the field lines, and the plasma's shape and size. TJ-II characteristics can be found in Table 2.1.

This work focuses on plasma diagnostics in the TJ-II stellarator, where collapses can severely degrade performance, interrupt experiments, and reduce overall efficiency. To address this, the PCAI tool was developed to automate processing of the TJ-II's 60,000-discharge database.

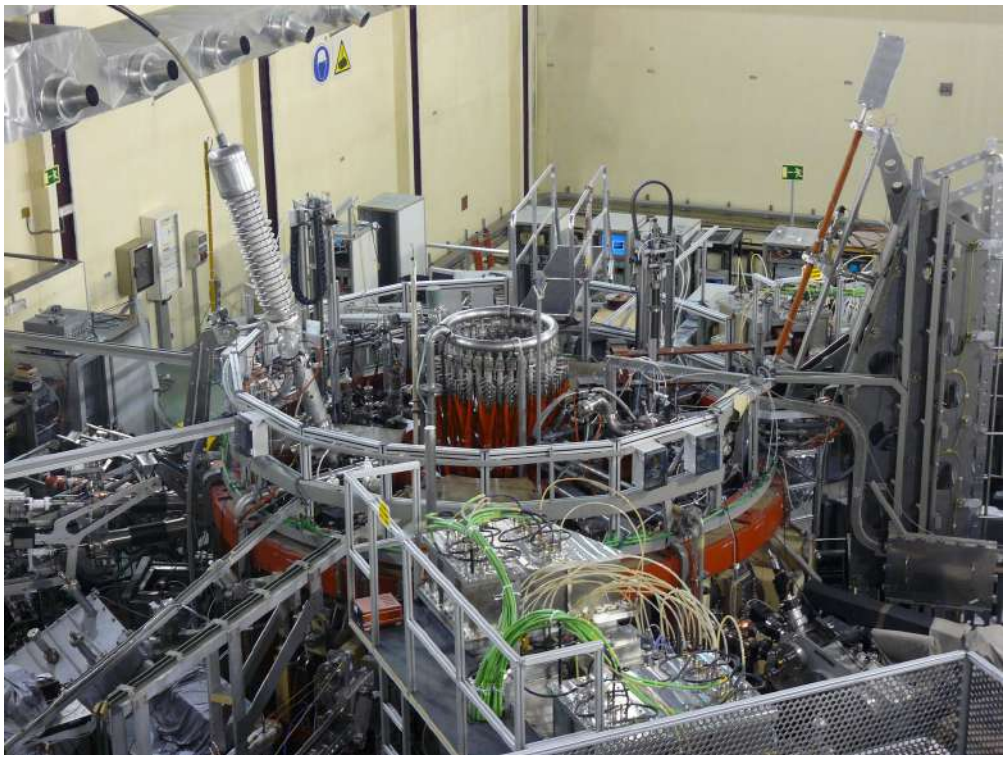

Figure 2.2: TJ-II.

Table 2.1: Main characteristics of TJ-II

Characteristic	**Value**
Device type	Medium-size heliac stellarator
Major radius R_0	1.5 m
Average minor radius a_0	≤ 0.22 m
Plasma volume	$\approx 1.1\,\mathrm{m}^3$
Toroidal field coils	32
Magnetic field at plasma center	1.1 T
Discharge duration	≈ 0.25 s
Repetition period	$\approx$ 7–8 min
Working gas	Hydrogen
ECRH System	
Number of ECRH injectors	2
Microwave frequency	53.2 GHz
Power per injector	200 kW - 250 kW
Central electron density $n_e(0)$ (ECRH)	$\leq 1.7 \times 10^{19}\,\mathrm{m}^{-3}$
Central electron temperature $T_e(0)$ (ECRH)	≤ 2 keV $\left(\sim 23.2 \times 10^6\,^\circ\mathrm{C}\right)$
NBI System	
Number of NBI injectors	2
Power per injector	600 MW
Central electron density $n_e(0)$ (NBI)	$\leq 5 \times 10^{19}\,\mathrm{m}^{-3}$
Central electron temperature $T_e(0)$ (NBI)	≤ 400 eV $\left(\sim 4.6 \times 10^6\,^\circ\mathrm{C}\right)$
Central ion temperature	≤ 120 eV $\left(\sim 1.4 \times 10^6\,^\circ\mathrm{C}\right)$

2.2 Machine Learning Applied to Nuclear Fusion

Artificial Intelligence (AI) is a broad field focused on creating systems that can perform tasks that require intelligent behavior, such as reasoning, decision-making, or understanding language.

On the other hand, Machine Learning (ML) is a subset of AI that focuses specifically on algorithms that learn from data to make predictions or decisions without being explicitly programmed. To do so, several algorithms have been developed throughout history to solve all types of problems, including nuclear fusion applications.

Machine learning in fusion has been successfully applied in JET. In particular, the prediction of plasma disruptions was carried out. Disruptions are very dangerous instabilities in tokamaks that abruptly finish the plasma with an extremely high energy deposition in the walls of the devices. Their effects can cause irreversible damage in the machine. A summary of different disruption predictors in JET can be found in [11].

Regarding the TJ-II stellarator, the application of machine learning to the classification of its discharges has evolved significantly over the past two decades. Early efforts combined Wavelet Transform (WT) and Support Vector Machines (SVM) to automatically identify patterns in large waveform databases. Dormido-Canto et al. [12] demonstrated that WT could effectively reduce the dimensionality of plasma signals, while SVM offered robust performance for binary classification of waveform features. This combination enabled efficient retrieval and sorting of thousands of TJ-II signals.

In a related study [13], Dormido-Canto et al. extended the use of WT+SVM for general-purpose classification and information retrieval in fusion databases, highlighting its advantages in terms of data compression and time-frequency representation. These works laid the groundwork for more domain-specific applications.

Vega et al. [14] proposed an intelligent classification pipeline specifically for Thomson Scattering (TS) diagnostic images. Their method used SVM and wavelet-based feature extraction to classify images into five physical categories (e.g., NBI), achieving a feature reduction from 221,760 to 900 attributes with high accuracy and integration into the TJ-II operation workflow.

Further work by Farias et al. [15] introduced a Feedforward Neural Network classifier trained on wavelet-extracted features from TS images. The system achieved over 90% accuracy and enabled automated profile reconstruction, showing strong potential for real-time classification of diagnostic outputs.

These studies illustrate the potential of combining signal processing techniques like wavelets with supervised learning algorithms for shot classification and diagnostic automation in TJ-II.

2.3 Scope

Given the signals explained in Section 1.1, in Table 1.1 from the TJ-II stellarator introduced in that section as well and in Section 2.1, Discrete Wavelet Transforms (DTW) are going to be applied to the signals, continuing with the work presented in Section 2.2. These transforms play a crucial role in determining the end of the discharge, defining the start of the collapse, i.e., the decay time of the plasma energy, explained in depth in Section 3.1.1 and allowing the composition of feature vectors so that the SVM classifier can be trained, tested and used for predicting the labels of the 60000 discharges stored in the database, process which was manually made before the creation of the PCAI Tool.

Chapter 3

Algorithms and Software Architecture

In this chapter, the algorithms and software architecture used to achieve the objective of this project are explained. It begins describing the techniques for processing the TJ-II signals, continues explaining with the methods for classifying both the shots and the density signals and ends with the integration of these methods and the architecture of the MATLAB-based GUI, the PCAI Tool.

3.1 Signal Processing

Within this subsection all the algorithms related to signal processing are explained, which are integrated in the PCAI Tool.

3.1.1 Determination of the Collapse Time

The collapse time is the instant when the energy signal begins to decay, i.e. the decay time of the energy signal, and indicates the start of the collapse of the discharge. The energy signal usually takes the form shown in Figure 3.1.

To obtain the collapse time, two methods are developed and applied to the energy signal:

- Discrete Wavelet Transform (DWT) method
- Integral derivative method

In the results section, both methods are compared. Shot number 56030 from the database is taken as an example for the figures and posterior evaluation as it contains a typical energy signal.

Discrete Wavelet Transform (DWT) Method

In the context of plasma diagnostics, signals often exhibit non-stationary behavior. Traditional Fourier analysis, while effective for stationary signals, lacks the ability to capture

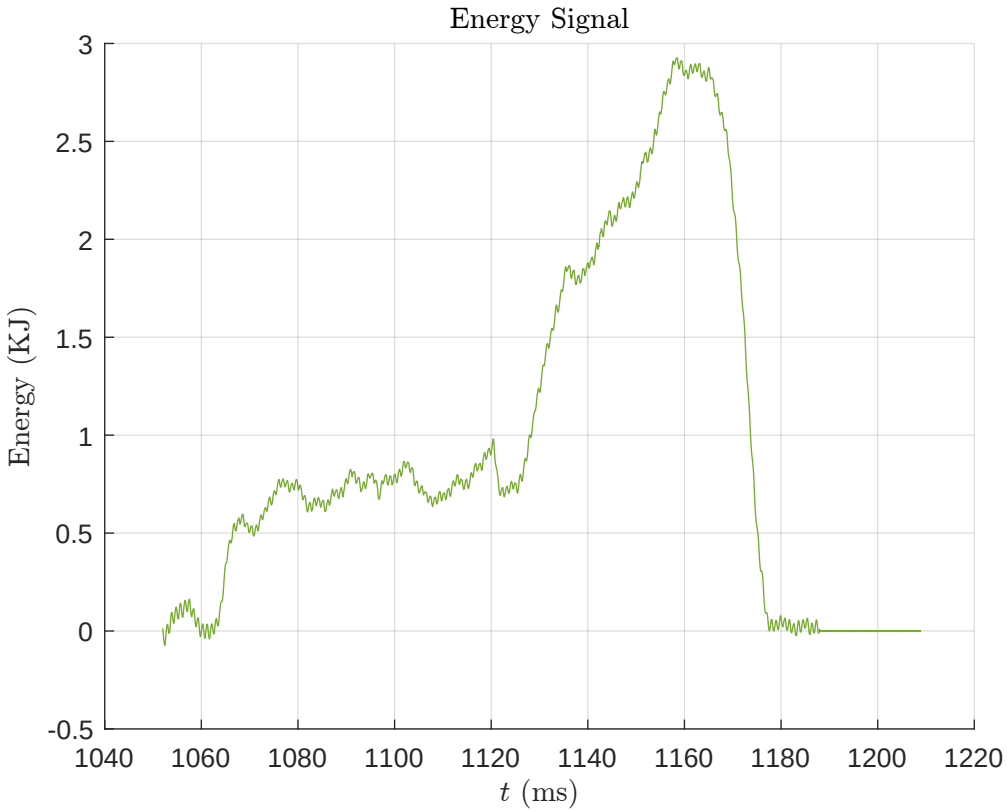

Figure 3.1: Signal of energy contained in the plasma.

localized time-frequency features that are crucial for characterizing rapid transitions. To overcome this limitation, wavelet analysis was employed as a first approach, enabling the decomposition of the signal with different resolutions. This approach provides both time and frequency localization, making it suitable for identifying the decay time.

The Discrete Wavelet Transform (DWT) provides a multiresolution representation of a signal by projecting it onto a set of orthonormal basis functions; those functions' actions correspond to a high-pass filter and a low-pass filter. In other words, given a discrete signal $x(k)$, the DWT hierarchically decomposes it into approximation (low-pass) and detail (high-pass) components on multiple scales. Formally, the approximation coefficients $a(m)$ and the detail coefficients $d(m)$, using the Haar wavelet [16], can be calculated as follows, being j the *decomposition level* in the multiresolution pyramid, the :

$$a_{j+1}(m) = \frac{x_j(2m) + x_j(2m+1)}{\sqrt{2}}, \quad \forall\, m \in \left\{0, 1, \ldots, \tfrac{N_j}{2} - 1\right\} \tag{3.1}$$

$$d_{j+1}(m) = \frac{x_j(2m) - x_j(2m+1)}{\sqrt{2}}, \quad \forall\, m \in \left\{0, 1, \ldots, \tfrac{N_j}{2} - 1\right\} \tag{3.2}$$

where:

- The higher the decomposition level j is, the more compressed is the signal

  - When $j = 0$, $x_0[k]$ is the original input signal
  - Applying one step of filtering and down-sampling produces the level-$(j+1)$ coefficients a_{j+1} and d_{j+1}

- m is the *discrete index* after down-sampling by 2

$$m = 0, 1, 2, \ldots, \frac{N_j}{2} - 1 \tag{3.3}$$

  where N_j is the length of the signal x_j at the level j. The signal sample size is usually a power of 2.

The decay of the energy signal must be studied from a global perspective rather than a local one. Consequently, the approximation coefficients, which capture the low-frequency, large-scale behavior of the signal, provide a more solid and proper basis for identifying the decay time, compared to the detail coefficients, which reflect high-frequency noise and small-scale variations.

The absolute value of these coefficients is then computed as it is required to know at what point the net magnitude of the signal is maximum:

$$M(m) = |a_j(m)| \tag{3.4}$$

The index $k_{\max}$ corresponding to the maximum value of E_k is identified:

$$k_{\max} = \arg\max[M(m)] \tag{3.5}$$

Given the level of decomposition and the fact that sample sizes are rescaled by a factor of 2^L), the original time index corresponding to this maximum is rescaled as follows:

$$i_{\text{decay}} = 2^j \times k_{\max} \tag{3.6}$$

Finally, the decay time t_{decay} is determined by mapping this index to the original time vector:

$$t_{\text{decay}} = t(i_{\text{decay}}) \tag{3.7}$$

If i_{decay} exceeds the signal length due to boundary effects, it is truncated to the last valid index.

Results of this algorithm will be compared with the integral derivative one in Section 4.1.1.

Integral Derivative Method

The moment the energy signal collapses, its derivative at that point must be negative. However, all the points where the signal is decreasing have a negative derivative. If the negative area under the curve of the derivative of the whole signal could be visualized, it would be clear that the first timestep of the largest negative area is the decay time. To achieve this, the plasma energy signal must be derivative. Instead, the raw signal is noisy, as can be seen in Figure 3.1. If the derivative is applied to the whole signal and only the negative areas under the curve are shown, the following phenomenon can be seen in Figure 3.2.

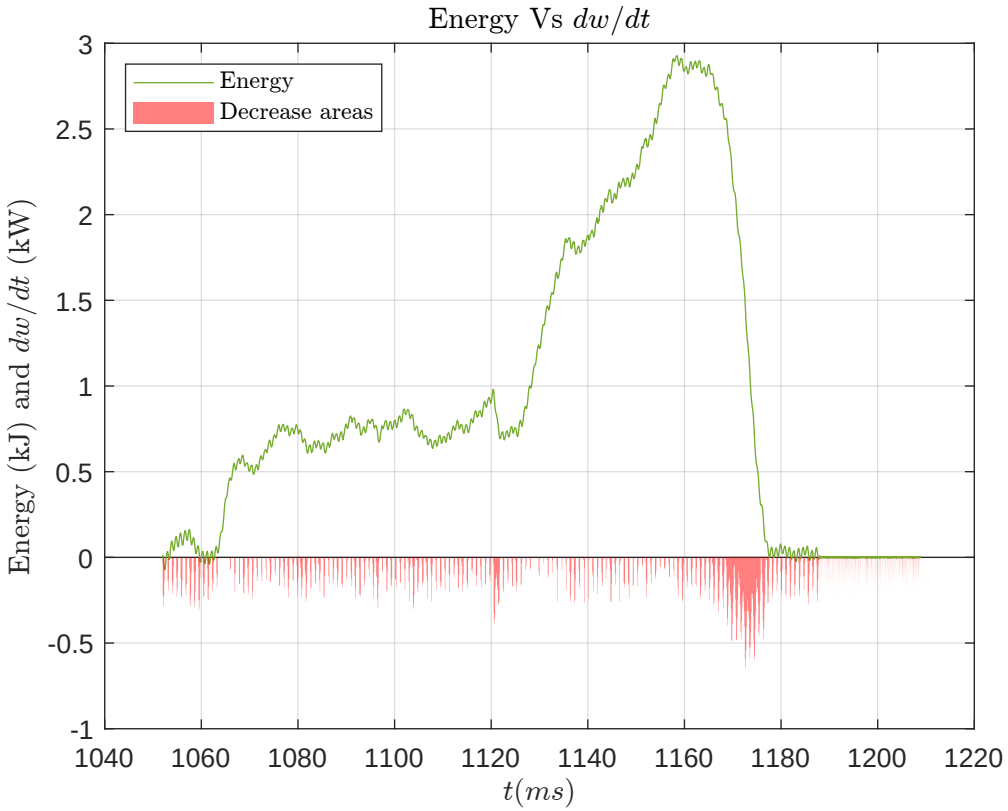

Figure 3.2: Energy signal with noise and decrease areas.

It can be noticed that the noise of the signal does not let visualize the intervals where the energy signal is decreasing, i.e., the area of the derivative is negative. To avoid this, and given the fact that the noise has a clear frequency, a Butterworth filter is applied to the signal, so it is easier to proceed with the derivative.

According to this, the integral derivative algorithm is divided into two steps:

1. Noise attenuation with Butterworth filter.
2. Definition of the intervals where the derivative of the signal is negative.

1. <u>Noise attenuation with Butterworth filter.</u>

To filter higher-frequency components of the energy signal, the noise in this case, and proceed with its derivative, a Butterworth filter was implemented via software in the PCAI Tool.

The complex transfer function of any Butterworth filter is:

$$H(s) = \frac{1}{D(s)} \tag{3.8}$$

where the polynomial of the denominator $D(s)$ is tabulated as the products of factors of first and second order.

Normalized denominator polynomials up to order 8 are shown below:

Table 3.1: Normalized Denominator Polynomials $D(s)$

Order n	$D(s)$
1	$(s+1)$
2	$(s^2+1.414\,s+1)$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.765\,s+1)(s^2+1.848\,s+1)$
5	$(s+1)(s^2+0.618\,s+1)(s^2+1.618\,s+1)$
6	$(s^2+0.518\,s+1)(s^2+1.414\,s+1)(s^2+1.932\,s+1)$
7	$(s+1)(s^2+0.445\,s+1)(s^2+1.247\,s+1)(s^2+1.802\,s+1)$
8	$(s^2+0.390\,s+1)(s^2+1.111\,s+1)(s^2+1.663\,s+1)(s^2+1.962\,s+1)$

By changing the filter order and the cutoff frequency, a whole family of Butterworth filters can be designed as shown in Figure 3.3, where the filter gain (in dB) is plotted against the frequency (in rad/s). Depending on these parameters, the filter response varies.

The best experimentally tested option for this case, is a Butterwoth filter with the characteristics given in Table 3.2.

Table 3.2: Butterworth filter parameters.

Algorithm	Decay Time (ms)
Parameters	*Value*
Order (n)	1
Cutoff Frequency (fc)	0.01

In this case, the cutoff frequency does not have units because the function used in MATLAB to implement the filter, `butter`, requires a value of fc that ranges from 0 to 1. The values assigned in the table are the ones that bring the best results, higher values of them over-filter the characteristics of the energy signal, and lower values let pass noise.

2. Definition of the intervals where the derivative of the signal is negative.

Let $\dot{w}$ be the derivative of the smoothed energy function:

$$\dot{w} = \frac{dw(t)}{dt} \tag{3.9}$$

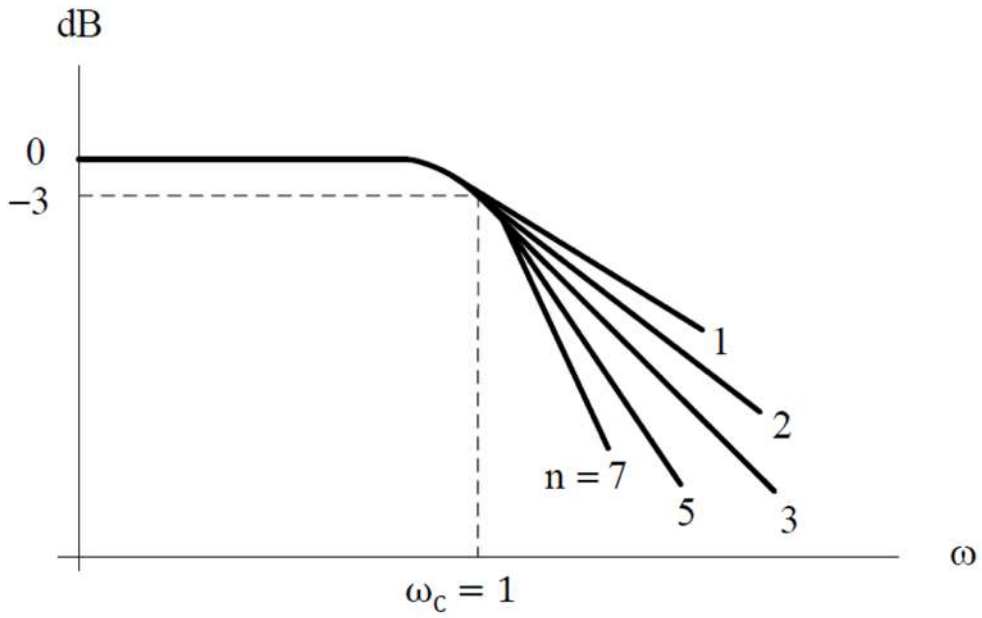

Figure 3.3: Butterworth filters family.

The set of intervals where the derivative of the energy signal is negative, I^-, are the ones where the derivative in each index of the interval, $\dot{w}_i$, is lower than zero:

$$I^- = \{i \mid \dot{w}_i < 0\} \tag{3.10}$$

Consequently, negative intervals can be defined as the connected components of I^- along the integer index line. Let

$$\mathcal{I}_1^-,\ \mathcal{I}_2^-,\ \ldots,\ \mathcal{I}_J^- \tag{3.11}$$

be the family of disjoint sets such that

$$I^- = \bigcup_{j=1}^{J} \mathcal{I}_j^-, \quad \mathcal{I}_j^- = \{\, i_s^j,\ i_s^j + 1,\ \ldots,\ i_e^j\}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.12}$$

where:

- J is the total number of negative intervals
- i_s^j is the start index of the interval j. The subindex "s" stands for "start"
- i_e^j is the end index of the interval j. The subindex "e" stands for "end"

In other words, each $\mathcal{I}_j^-$ consists of consecutive indices with $\dot{w}_i < 0$ which are the intervals where the derivative of the energy signal is negative, and cannot be extended without ceasing to belong to I^-.

To determine the magnitude of each interval, the area of each one can be integrated, which at the end, is the difference between the first point of the interval and the last.

$$\Delta w_j^- = \int_{t_{i_s}}^{t_{i_e}} \dot{w}(t) = \frac{w_{i_s} - w_{i_e}}{t_{i_s} - t_{i_e}}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.13}$$

The interval with the most negative area of the derivative interval, is the one where the signal starts to decay:

$$j_{min} = \arg\min_j \Delta w_j^-, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.14}$$

$$\mathcal{I}_{decay}^- = \mathcal{I}_{j_{min}}^- \tag{3.15}$$

being:

$$\mathcal{I}_{j_{min}}^- = \{\, i_s^{j_{min}},\ i_s^{j_{min}} + 1,\ \ldots,\ i_e^{j_{min}} \} \tag{3.16}$$

where $i_s^{j_{min}}$ is the decay index of the energy signal function, consequently:

$$t_{decay} = t(i_e^{j_{min}}) \tag{3.17}$$

$$w_{decay} = w(t_{decay}) \tag{3.18}$$

The decay time obtained thanks to this algorithm will be compared with the result of the DWT method in Section 4.1.1, and an equivalent filtered energy signal with derivatives analogous to that shown in Figure 3.2 is presented.

3.1.2 Determination of the NBI End Time

NBI signals can be defined as P_1 for NBI1, and P_2 for NBI2. Using the same method to obtain the decay time of the energy function in Section 3.1.1, the decay time for both NBIs, is obtained, $t_{decay,1}$ and $t_{decay,2}$. The index of the function where it decays are $i_{decay,1}$ and $i_{decay,2}$.

The end of the NBI signal is considered the relative minimum value after the decay time. Consequently, a window W can be defined to obtain the indices of the points of that window, which is the set $\mathcal{W}_|$, where j stands for the NBI number, 1 or 2.

$$\mathcal{W}_j = \{\, i_{decay,j} - W/2,\ i_{decay,j} - W/2 + 1,\ \ldots,\ i_{decay,j} + W/2\}, \quad \forall j \in \{1, 2\} \tag{3.19}$$

Consequently, the end time for every NBI signal is defined as follows.

$$i_{end,j} = \arg\max_i P_j(i), \quad \forall i \in \{\mathcal{W}_j\}, \quad \forall j \in \{1, 2\} \tag{3.20}$$

Then, the end time of the NBI signals and their values can be computed.

$$t_{end,j} = t_j(i_{end,j}), \quad \forall j \in \{1, 2\} \tag{3.21}$$

$$P_{end,j} = P_j(i_{end,j}), \quad \forall j \in \{1, 2\} \tag{3.22}$$

The end time of the shot is the maximum value of the two end times calculated.

$$t_{end} = \max_j (t_{end,j}), \quad \forall j \in \{1, 2\} \tag{3.23}$$

Sometimes, there are discharges in which there is only one NBI signal or none of them. To check whether a NBI signal is available or not, the maximum value of each one is compared with a threshold, θ. So, the condition is:

$$\max_j (P_j) > \theta, \quad \forall j \in \{1, 2\} \tag{3.24}$$

If there are not any NBI signals available, the discharge is not relevant for the purposes of analyzing collapsing shots. If there is only one, t_{end} takes the value of the only present NBI signal. The parameters mentioned in this section take the values shown in Table 3.3 in the PCAI Tool.

Table 3.3: NBI End Time Parameters

Parameter	Value	Unit
W	512	*steps*
θ	0.3	*a.u.*

This algorithm will be evaluated in Section 4.1.2.

3.1.3 Density Signal Reconstruction

One of the objectives of this work is to determine the value of the density signal when a collapse has occurred. However, signals of this type are acquired using an interferometer and can exhibit a phenomenon called fringe jump, as can be seen in Figure 3.4, so they must be preprocessed before their use to avoid overestimating the line averaged-electron density.

To reconstruct the function, the steps of the function must be determined and removed. To achieve this, the following algorithm has been determined with the next steps, where the positive and negative intervals of the density's function derivative must be determined:

1. **Definition of the positive intervals that are steps.**

The derivative of the density function can be defined as follows:

$$\dot{n} = \frac{dn(t)}{dt} \tag{3.25}$$

For a discrete time domain, the derivative can be calculated as:

$$\dot{n}_i = \frac{n_{i+1} - n_i}{t_{i+1} - t_i}, \quad \forall i \in \{1, 2, \ldots, N-1\} \tag{3.26}$$

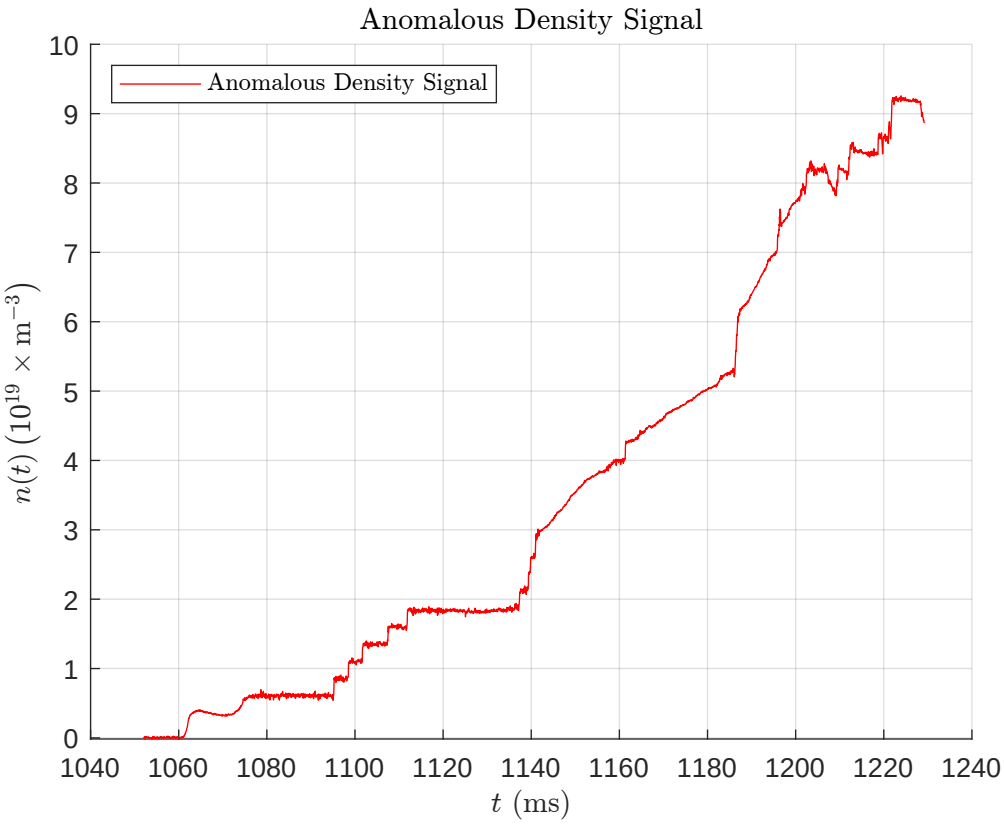

Figure 3.4: Anomalous density signal plot with fringe jumps that overestimate the value of the line-averaged electron density.

where N is the total number of points of the discrete density function.

According to this, the set of positive intervals where there are possible steps, I^+, are the ones where the derivative in each index of the interval, $\dot{n}_i$, is higher than a defined threshold, ρ^+:

$$I^+ = \{i \mid \dot{n}_i > \rho^+\} \tag{3.27}$$

Consequently, positive intervals can be defined as the connected components of I^+ along the integer index line. Let

$$\mathcal{I}_1^+,\ \mathcal{I}_2^+,\ \ldots,\ \mathcal{I}_J^+ \tag{3.28}$$

be the family of disjoint sets such that

$$I^+ = \bigcup_{j=1}^{J} \mathcal{I}_j^+, \quad \mathcal{I}_j^+ = \{\, i_s^j,\ i_s^j + 1,\ \ldots,\ i_e^j\}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.29}$$

where:

- J is the total number of positive intervals
- i_s^j is the start index of the interval j. The subindex "s" stands for "start"

- i_e^j is the end index of the interval j. The subindex "e" stands for "end"

On balance, each $\mathcal{I}_j^+$ consists of consecutive indices with $\dot{n}_i > \rho^+$ which are possible steps, and cannot be extended without ceasing to belong to I^+.

To evaluate whether each positive interval could be a fringe jump or not, the magnitude of the step, Δ^+, and the calculated area of the positive interval, A^+, must be compared with the thresholds δ^+ and τ^+, respectively. Let

$$A_j^+ = \int_{t_{i_s}}^{t_{i_e}} n(t)\, dt, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.30}$$

$$\Delta_j^+ = \frac{n_{i_s} - n_{i_e}}{t_{i_s} - t_{i_e}}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.31}$$

be the magnitude and area of each positive interval. To consider whether the interval $\mathcal{I}_j^+$ is a step or not, the following conditions must be met:

$$\Delta_j^+ > \delta^+ \quad \wedge \quad A_j^+ > \tau^+ \tag{3.32}$$

The set of positive steps can be defined as follows:

$$S^+ = \{j \mid \Delta_j^+ > \delta^+ \quad \wedge \quad A_j^+ > \tau^+\} \tag{3.33}$$

Consequently, positive intervals that are steps can be defined as the connected components of S^+ along the integer index line. Let

$$\mathcal{S}_1^+,\ \mathcal{S}_2^+,\ \ldots,\ \mathcal{S}_K^+ \tag{3.34}$$

be the family of disjoint sets such that

$$S^+ \ = \ \bigcup_{k=1}^{K} \mathcal{S}_k^+, \quad \mathcal{S}_k^+ = \{\, i_s^k,\ i_s^k + 1,\ \ldots,\ i_e^k\}, \quad \forall k \in \{1, 2, \ldots, K\} \tag{3.35}$$

where:

- K is the total number of positive intervals that are steps
- i_s^k is the start index of the interval k. The subindex "s" stands for "start"
- i_e^k is the end index of the interval k. The subindex "e" stands for "end"

2. Compensation of the positive steps.

To mitigate the effects of the steps, the following equation must be applied to the defined positive intervals categorized as steps.

$$\epsilon_i = n(t_{i+1}) - n(t_i), \quad \forall k \in \{1, 2, \ldots, K\}, \quad \forall i \in \mathcal{S}_k^+ \tag{3.36}$$

$$n(t_l) = n(t_l) - \epsilon_i, \quad \forall k \in \{1, 2, \ldots, K\}, \quad \forall i \in \mathcal{S}_k^+, \quad \forall l \in \{i+1, i+2, \ldots, N\} \tag{3.37}$$

Where:

- K is the total number of positive intervals that are steps
- N is the total number of points of the discrete density function

3. Definition of the negative intervals that are steps.

The set of negative intervals where there are possible steps, I^-, are the ones where the derivative in each index of the interval, $\dot{n}_i$, is lower than a defined threshold, ρ^-:

$$I^- = \{i \mid \dot{n}_i < \rho^-\} \tag{3.38}$$

Consequently, negative intervals can be defined as the connected components of I^- along the integer index line. Let

$$\mathcal{I}_1^-,\ \mathcal{I}_2^-,\ \ldots,\ \mathcal{I}_J^- \tag{3.39}$$

be the family of disjoint sets such that

$$I^- = \bigcup_{j=1}^{J} \mathcal{I}_j^-, \quad \mathcal{I}_j^- = \{\, i_s^j,\ i_s^j + 1,\ \ldots,\ i_e^j\}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.40}$$

where:

- J is the total number of negative intervals
- i_s^j is the start index of the interval j
- i_e^j is the end index of the interval j

In summary, each $\mathcal{I}_j^-$ consists of consecutive indices with $\dot{n}_i < \rho^-$ which are possible steps, and cannot be extended without ceasing to belong to I^-.

To evaluate whether each negative interval could be a fringe jump or not, the magnitude of the step, Δ^-, and the calculated area of the negative interval, A^-, must be compared with the thresholds δ^- and τ^-, respectively. Let

$$A_j^- = \int_{t_{i_s}}^{t_{i_e}} n(t)\, dt, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.41}$$

$$\Delta_j^- = \frac{n_{i_s} - n_{i_e}}{t_{i_s} - t_{i_e}}, \quad \forall j \in \{1, 2, \ldots, J\} \tag{3.42}$$

be the magnitude and area of each negative interval. To consider whether the interval $\mathcal{I}_j^-$ is a step or not, the following conditions must be met:

$$\Delta_j^- < \delta^- \quad \wedge \quad A_j^- < \tau^- \tag{3.43}$$

The set of negative steps can be defined as follows:

$$S^- = \{j \mid \Delta_j^- < \delta^- \quad \wedge \quad A_j^- < \tau^-\} \tag{3.44}$$

Consequently, negative intervals that are steps can be defined as the connected components of S^- along the integer index line. Let

$$\mathcal{S}_1^-,\ \mathcal{S}_2^-,\ \ldots,\ \mathcal{S}_K^- \tag{3.45}$$

be the family of disjoint sets such that

$$S^- = \bigcup_{k=1}^{K} \mathcal{S}_k^-, \quad \mathcal{S}_k^- = \{\, i_s^k,\ i_s^k + 1,\ \ldots,\ i_e^k\}, \quad \forall k \in \{1, 2, \ldots, K\} \tag{3.46}$$

where:

- K is the total number of negative intervals that are steps
- i_s^k is the start index of the interval k. The subindex "s" stands for "start"
- i_e^k is the end index of the interval k. The subindex "e" stands for "end"

4. Compensation of the negative steps.

As a final step, to mitigate the effects of the negative steps, the following equation must be applied to the defined negative intervals categorized as steps.

$$\epsilon_i = n(t_{i+1}) - n(t_i), \quad \forall k \in \{1, 2, \ldots, K\}, \quad \forall i \in \mathcal{S}_k^- \tag{3.47}$$

$$n(t_l) = n(t_l) - \epsilon_i, \quad \forall k \in \{1, 2, \ldots, K\}, \quad \forall i \in \mathcal{S}_k^-, \quad \forall l \in \{i+1, i+2, \ldots, N\} \tag{3.48}$$

where:

- K is the total number of negative intervals that are steps
- N is the total number of points of the discrete density function

Having tested several values for all the mentioned thresholds, the most appropriate experimental values are gathered in Table 3.4.

The algorithm for the density signal reconstruction will be evaluated in Section 4.1.3.

Table 3.4: Density Signal Reconstruction Algorithm Thresholds

Threshold	Value	Unit
ρ^+	0.01	$10^{19}\,\mathrm{m}^{-3}\,s^{-1}$
δ^+	0.5	$10^{19}\,\mathrm{m}^{-3}\,s^{-1}$
τ^+	0.15	$10^{19}\,\mathrm{m}^{-3}\,s$
ρ^-	-0.01	$10^{19}\,\mathrm{m}^{-3}\,s^{-1}$
δ^-	-0.5	$10^{19}\,\mathrm{m}^{-3}\,s^{-1}$
τ^-	-0.13	$10^{19}\,\mathrm{m}^{-3}\,s$

3.2 Support Vector Machines

SVMs are supervised learning models for binary classification (with extensions to multi–class problems). An SVM seeks for the hyperplane in a high-dimensional feature space that best separates two classes by maximizing the margin between them.

Given a training set $(\mathbf{x}_i, y_i)$ with $y_i \in \{+1, -1\}$ and $i = 1, ..., n$, being n the total number of training samples, the SVM finds a weight vector $\mathbf{w}$ and bias b defining the decision function

$$D(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b \tag{3.49}$$

such that

$$y_i\big[(\mathbf{w} \cdot \mathbf{x}_i) + b\big] \geq \Delta, \quad \forall i \in \{1, 2, \ldots, n\} \tag{3.50}$$

and the margin $2\Delta/\|\mathbf{w}\|$ between classes is maximized (optimal hyperplane) by solving

$$\min_{\mathbf{w},b,\Delta>0} \ \frac{\|\mathbf{w}\|^2}{\Delta^2} \tag{3.51}$$

where:

- $D(x)$ is the decision function that enables classification
- $\mathbf{x}_i$ is the feature vector
- y_i is the labels vector
- $\mathbf{x}$ is the point to be classified
- $\mathbf{w}$ is the weight vector that defines the orientation (normal direction) of the separating hyperplane in feature space
- b is the bias (or intercept) of the SVM and represents the displacement of the hyperplane relative to the origin
- i is the index for each training example in the dataset
- n is the total number of training samples

- Δ is the shortest distance between the separating hyperplane and the nearest data point

The dual form of the SVM decision function [17] can be written as:

$$D(\mathbf{x}) \;=\; \sum_{i=1}^{n} \alpha_i^*\, y_i\, (\mathbf{x}\cdot\mathbf{x}_i) \;+\; b^* \tag{3.52}$$

where:

- α_i^* are the Lagrange multipliers
- b^* is the bias term

For the optimal separating hyperplane, the constraints below must be met.

1. The optimal Lagrange multipliers $\{\alpha_i^*\}_{i=1}^n$ must satisfy

$$\sum_{i=1}^{n} \alpha_i^*\, y_i = 0, \qquad \alpha_i^* \geq 0, \qquad \forall i \in \{1, 2, \ldots, n\} \tag{3.53}$$

2. The optimal weight vector $\mathbf{w}^*$ defining the hyperplane is a weighted sum of the training vectors,

$$\mathbf{w}^* \;=\; \sum_{i=1}^{n} \alpha_i^*\, y_i\, \mathbf{x}_i, \qquad \alpha_i^* \geq 0, \qquad \forall i \in \{1, 2, \ldots, n\} \tag{3.54}$$

By replacing the inner product $\mathbf{x}\cdot\mathbf{x}_i$ with a Kernel function $K(\mathbf{x}, \mathbf{x}_i)$, the SVM decision function becomes

$$D(\mathbf{x}) \;=\; \sum_{i=1}^{n} \alpha_i^*\, y_i\, H(\mathbf{x}, \mathbf{x}_i) \;+\; b^* \tag{3.55}$$

Common Kernels are:

- Linear Kernel:

$$H(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i^\top \mathbf{x} \tag{3.56}$$

- Polynomial of degree q kernel:

$$H(\mathbf{x}_i, \mathbf{x}) = \left[(\mathbf{x}_i^\top \mathbf{x}) + 1\right]^q \tag{3.57}$$

For this project, the Kernel function used is the linear case. In the following subsections, the particularities of each classification problem are explained.

3.2.1 Discharge Classification

The particularities of each classification problem are given by the feature vectors $\mathbf{x}_i$. For the shot classification case, feature vectors are defined by the signals used, the sample size, S, and the decomposition level as the DWT is used to compress the signals by means of the approximation coefficients, formally described in Equation 3.1. Last two points mentioned are applied equally to every signal of the vector. Using the energy, $w(t)$, and ACTON 275, $a(t)$, signals as an example, the feature vector takes the form below.

$$\mathbf{x}_i = [w_{i,(T-S-1):T}, a_{i,(T-S-1):T}], \quad \forall i \in \{1, 2, \ldots, n\} \tag{3.58}$$

where:

- $\mathbf{x}_i$ is the feature vector
- i is the index for each training example in the dataset
- n is the total number of training samples
- S is the sample size, applied to all signals
- T is the total number of points of each signal
- w_i is the energy signal
- a_i is the ACTON 275 signal

A more general equation can be defined taking into account a total number of signals of m, which can be defined as $\{s_1, s_2, \ldots, s_m\}$, so that the generalized equation would be

$$\mathbf{x}_i = [s_{i,j,(T-S-1):T}, \ldots, s_{i,m,(T-S-1):T}], \quad \forall i \in \{1, 2, \ldots, n\}, \quad \forall j \in \{1, 2, \ldots, m\} \tag{3.59}$$

Taking into account the criteria presented in Section 1.1, which can be seen perfectly in the example in Figure 1.5, collapsing shots are labeled with the value -1 and non-collapsing with the value 1.

The results on the classification of shots are shown in Subsection 4.3.2.

3.2.2 Density Classification

For the density signal classification case, vectors are defined by the density signal, which is the only one used, the decomposition level as the DWT is used to compress the signals, and both types of coefficients are selectable as well, approximation coefficients are formally described in Equation 3.1 and detail coefficients in Equation 3.2.

The criteria to consider whether a density signal is anomalous (the label values -1) or not is the fact that it exhibits a phenomenon called fringe jump, as explained in Section 3.1.3,

otherwise, it is considered regular (the label values 1). A comparison between both cases can be visualized in Figure 3.5.

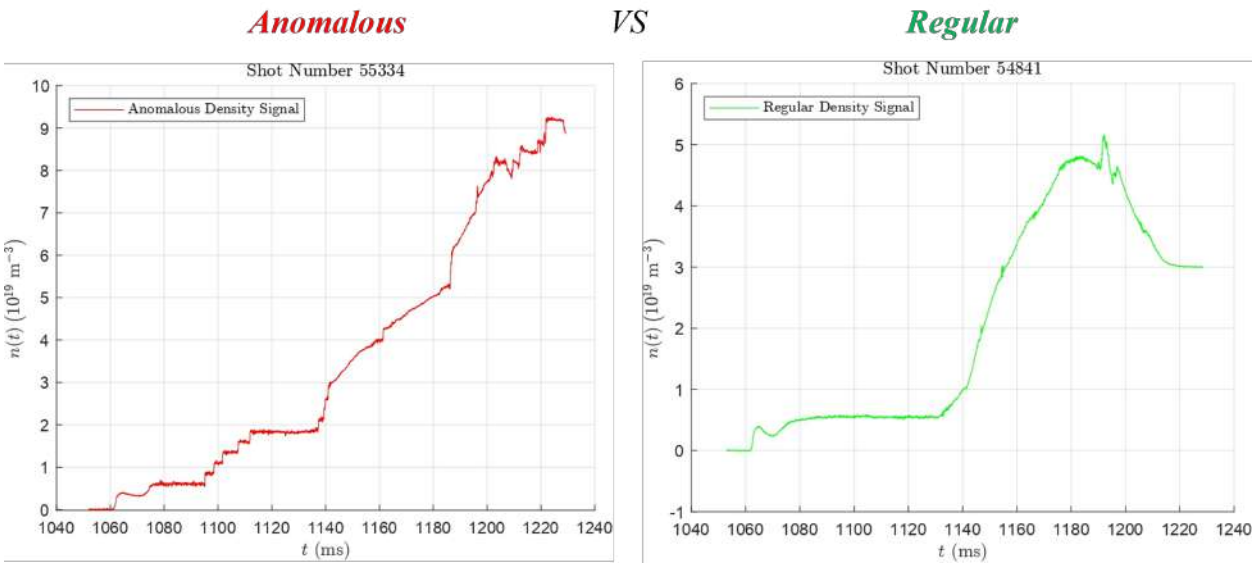

Figure 3.5: Anomalous vs regular density signals.

The results on the classification of the density signals are shown in Subsection 4.3.3.

3.3 PCAI Tool

The PCAI Tool is a MATLAB-based compiled software application which integrates advanced signal processing and a discharge classifier based on SVMs. As it is compiled, the final user does not require a MATLAB license to run it, so that it is easily distributed among all the users of the LNF that are in charge to perform signal analysis and/or discharge classification on the TJ-II database.

As mentioned before, discharge classification was done manually, which is an unapproachable mission for a database of 60000 discharges. Here lies the importance of the PCAI Tool, which will allow LNF users to rapidly perform analysis on existing and new discharges. For this reason, during the installation process, the user can configure the signals that the app will work with, so that future updates of the TJ-II signal acquisition system are taken into account and the app can adapt to those changes. To install the app, instructions available in the Installation Guide in Appendix A.1 must be followed. Within this section the software architecture is explained. For further details about the usage instructions, the User Guide is available in Appendix A.2.

3.3.1 Software Architecture

The PCAI Tool consists of the functional modules illustrated in Figure 3.6.

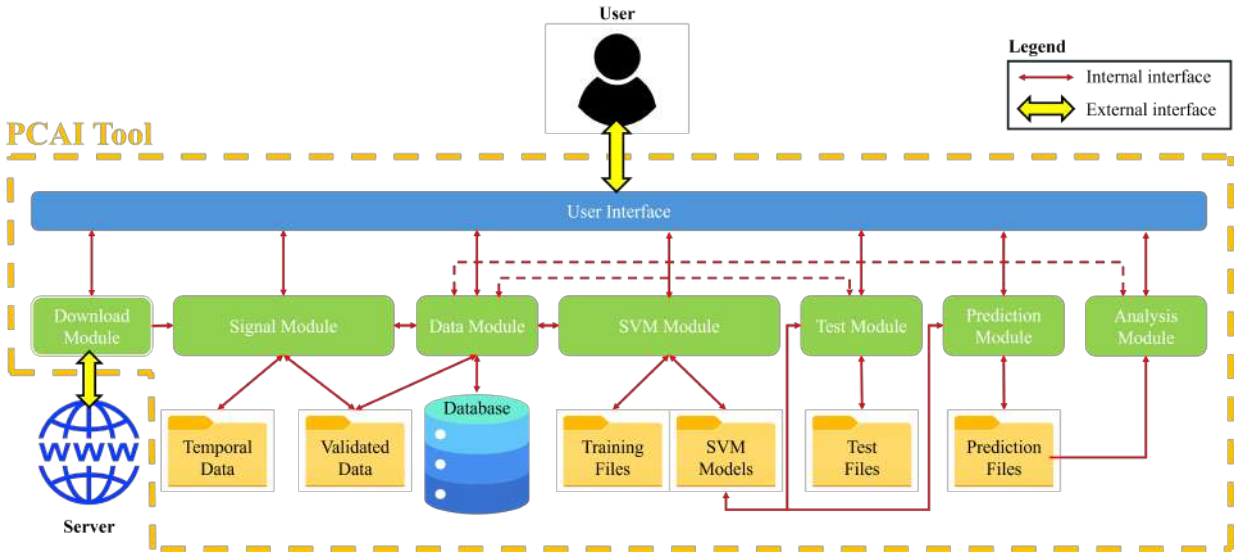

Figure 3.6: PCAI Tool architecture.

Every functional module interacts with the user interface, from which the user manages them. The functionalities of each module are described below.

- **Download Module**, downloads the signals of each discharge from the Web server where the TJ-II discharges are stored, and passes the downloaded information to the Signal Module.

- **Signal Module**, integrates the algorithms explained in this section and stores each discharge with its corresponding signals in the "Temporal Data" folder, in a ".mat" file format. Shots within these files must be assessed by the user through the GUI. Once they are evaluated, by a command of the user interface they are passed from the temporary folder to the definitive one, "Validated Data". Discharges within the latter folder are used for further operations. Within this module, functions of the MATLAB Signal Processing Toolbox are used to use the Butterworth filter, like `butter`, and `filter`. Also, for the DWT functions of the MATLAB Wavelet Toolbox are used, like `dwt` and `wavedec`.

- **Data Module**, stores and retrieves information from the database and the discharges available in the validated data folder. It communicates with the user through the GUI to store the discharges and density signals' labels assigned by the expert system, i.e. the user. If necessary, communicates back with the signal module for further signal processing. All modules that require data retrieval are connected to this one. Within this module, functions of the MATLAB Database Toolbox are used to handle the database.

- **SVM Module**, generates the feature vectors to train the SVM models and saves them in a ".mat" file within the "Training Files" folder. With those files, the module trains

SVM models and saves them in a ".mat" file within the "SVM Models" folder. Within this module, functions of the MATLAB Statistics & Machine Learning Toolbox are used to operate with the SVM models, like `fitcsvm` for training and `fitSVMPosterior`.

- **Test Module**, uses an SVM model to test it with labeled discharges or density signals. Results are stored in a ".mat" file within the "Test Files" folder. Within this module, functions of the MATLAB Statistics & Machine Learning Toolbox are used to operate with the SVM models, like `predict` for prediction.

- **Prediction Module**, uses an SVM model to predict unlabeled discharges or density signals. Results are stored in a ".mat" file within the "Prediction Files" folder. Within this module, functions of the MATLAB Statistics & Machine Learning Toolbox are used to operate with the SVM models, like `predict` for prediction.

- **Analysis Module**, uses the Prediction Model files and the manually labeled data retrieved by the Data Module to perform further analysis, which is output through the user interface.

With the PCAI Tool architecture in mind, a brief use case is introduced in Section 4.2.

Chapter 4

Results

This chapter presents the key results obtained through the application of the PCAI Tool and the associated processing and classification algorithms. Performance metrics are used to quantify the effectiveness of the SVM classifiers.

4.1 Signal Processing Results

In this section, methods regarding signal processing will be evaluated and, if applicable, compared between them.

4.1.1 Collapse Time Algorithms Comparison

Here, the following methods to obtain the decay time are compared, so that the best option to estimate that time can be assessed.

- Integral derivative method
- Discrete Wavelet Transform (DWT) method

As an example, shot number 56030 from the database is taken to proceed with this evaluation as it contains a good example of energy signal.

The first point to evaluate is noise attenuation. As shown in Figure 3.2, noise does not allow to clearly distinguish decrease areas. By applying the Butterwoth filter, most of the noise is eliminated so that the extension of the areas can be compared, as can be seen in Figure 4.1. It can be distinguished clearly that the first timestep of the largest area is the decay time.

When the Integral Derivative algorithm is executed, it returns the following decay time with its associated energy (Eq. 4.1), and, it can be visualized in Figure 4.2.

$$t_{decay} = 1166.2218\ ms, \quad w_{decay} = 2.7484\ kJ \tag{4.1}$$

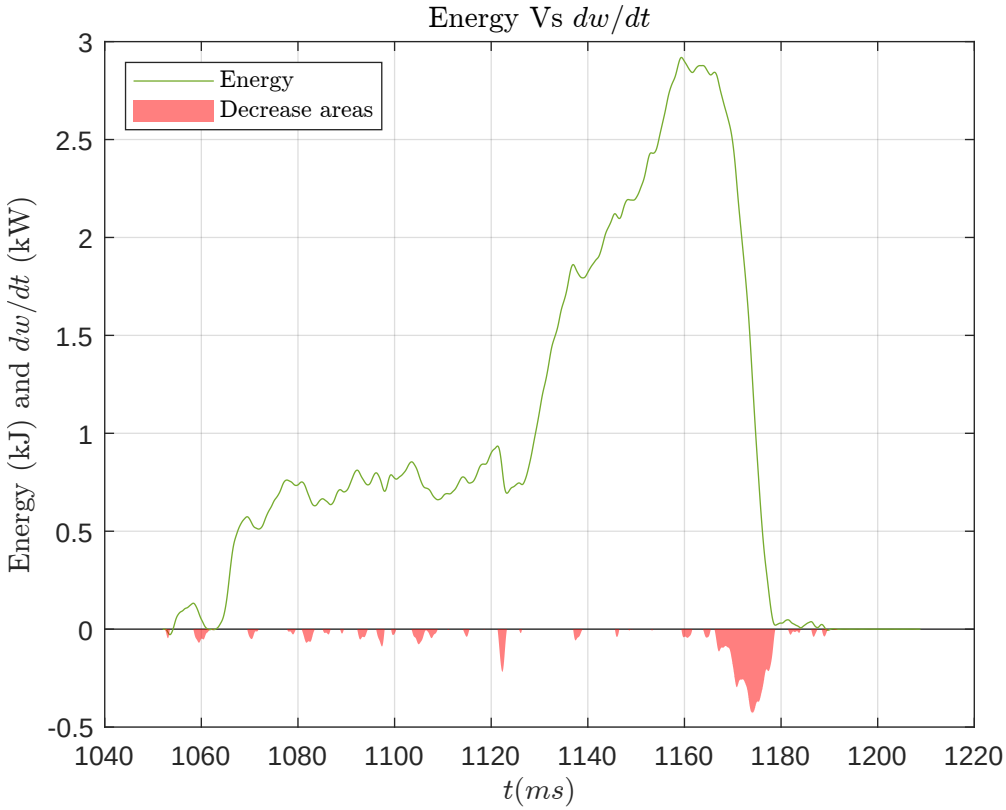

Figure 4.1: Energy signal with attenuated noise and decrease areas.

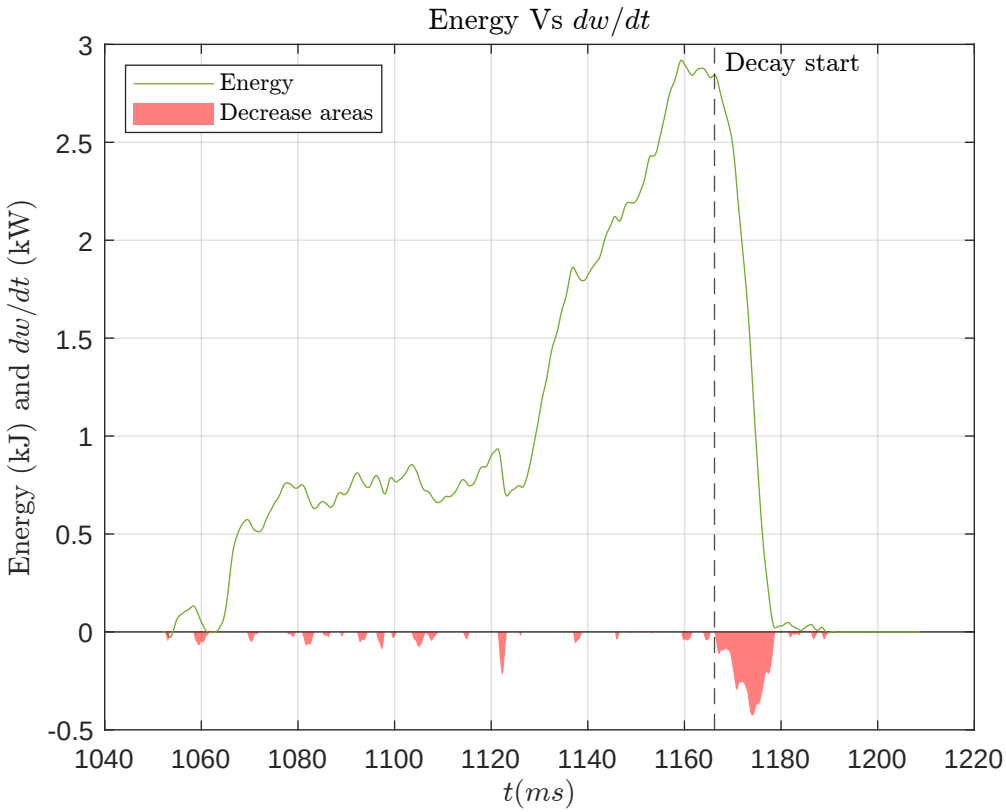

Figure 4.2: Integral derivative algorithm decay time.

On the other hand, when the DWT algorithm is run, it returns the following decay time with its associated energy (Eq. 4.2), and it can be visualized in Figure 4.3.

$$t_{decay} = 1158.4117\ ms, \quad w_{decay} = 2.9273\ kJ \tag{4.2}$$

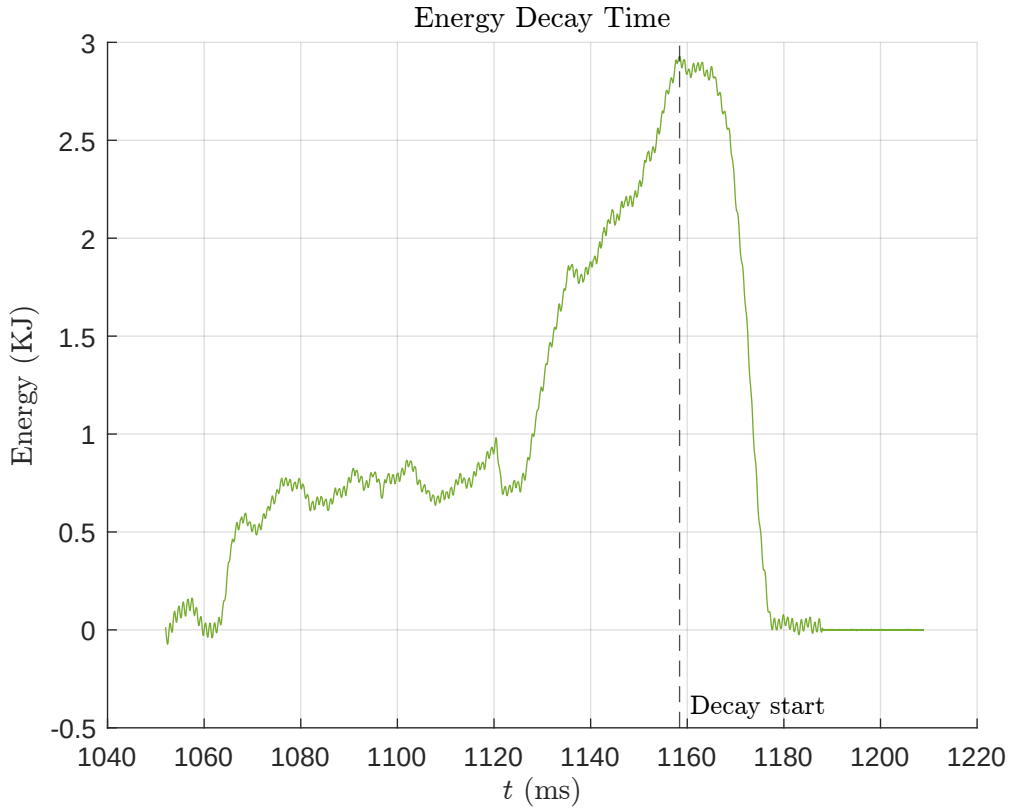

Figure 4.3: DWT decay time.

After visualizing both results, it can be concluded that the DWT algorithm usually takes as the decay time the maximum value of the signal. In contrast, the integral derivative method implements a literal approach to the definition of the decay time, so it will be the go-to method to use in further analysis. In Table 4.1, results obtained from this comparison are shown with the calculated differences between each algorithm.

Table 4.1: Energy signal decay time and value difference between tables

Algorithm	Decay Time (ms)	Energy (kJ)
DWT	1158.4117	2.9273
Integral Derivative	1166.2218	2.7484
Difference	7.8101	0.1789

4.1.2 NBI End Time Algorithm Results

Given the algorithm presented in Section 3.1.2, it can be tested by processing the shot number 56030, which contains both NBI signals. In Figure 4.4, the end times of both signals are highlighted with vertical dashed lines.

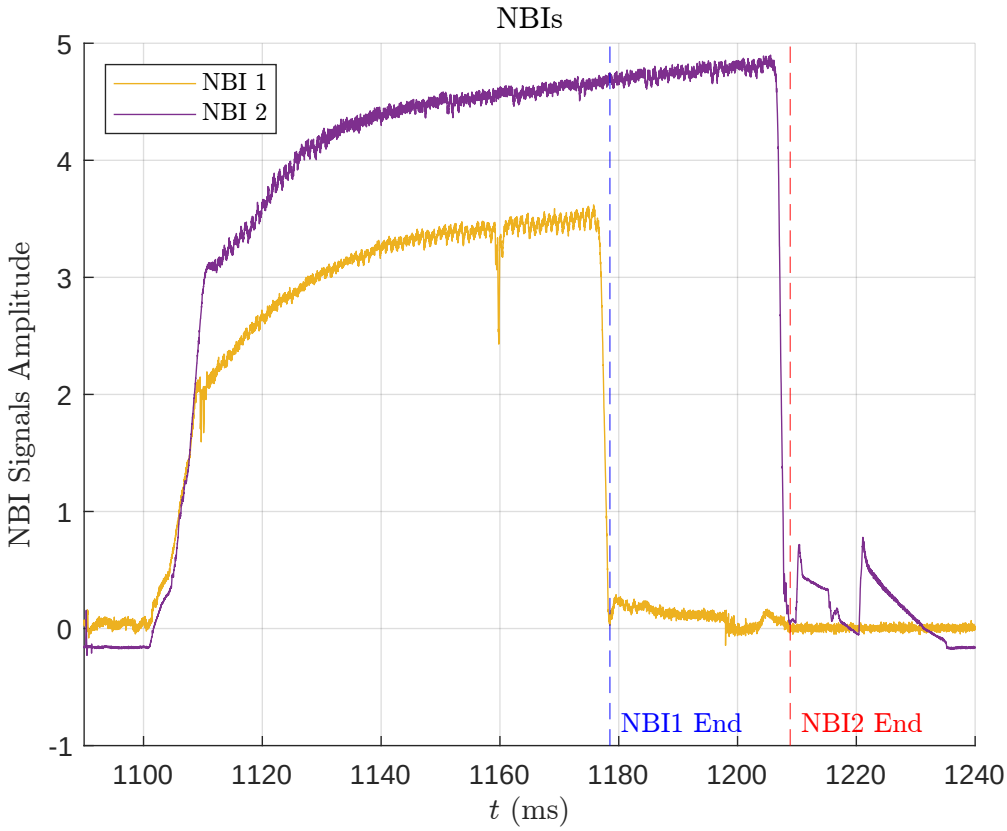

Figure 4.4: NBIs end times.

Where the computed times are presented in Table 4.2.

Table 4.2: NBIs end times.

NBI	Decay Time (ms)
$NBI1$	1178.5117
$NBI2$	1208.8718

Here,

$$t_{end,2} > t_{end,1}, \quad 1208.8718\ ms > 1178.5117\ ms \tag{4.3}$$

So

$$t_{end} = t_{end,2} = 1208.8718\ ms \tag{4.4}$$

As seen in Figure 4.4, the end time of the shot is denoted by the NBI2 signal.

4.1.3 Density Signal Reconstruction Algorithm Results

To test the reconstruction algorithm presented in Section 3.1.3, the density signal of the shot number 55334 is assessed. In Figure 4.5, the anomalous signal is displayed in red, which can be compared with the reconstructed one in green. Increase and decrease areas are shown in cyan and magenta, respectively.

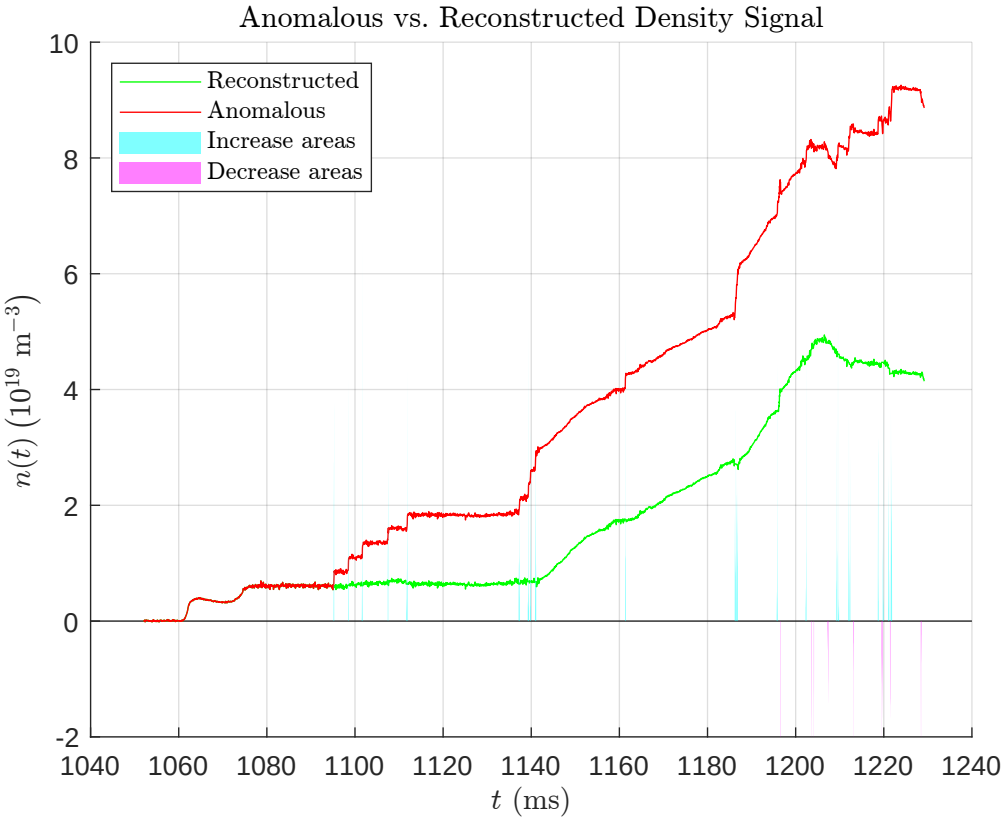

Figure 4.5: Density signal reconstruction algorithm comparison.

In Figure 4.5, it can be distinguished how the algorithm detects areas where there are positive or negative steps, thanks to filtering them with the thresholds presented in Table 3.4. By attenuating the steps, there is a significant improvement from the anomalous signal to the reconstructed one.

4.2 PCAI Tool Use Case

Once installed, see the Installation Guide in Appendix A.1, the PCAI Tool can be launched and the main screen appears, shown in Figure 4.6. The main screen of the app is the "Download" screen, where the user can download discharges, configure the download tags of every signal, i.e. the name with which the Download Module asks to the web server to download a specific signal of a shot, and visualize the log of the app. The information gathered from the Download Module is passed to the Signal Module to store the signals of the discharges within the temporary folder. Inside that folder, there is one ".mat" file per discharge.

Thanks to the tab system on top of the screen the user can navigate easily through the screens of the tool. The tab system is shown below.

- Signal
  - Download
  - First Assessment
  - Download Additional Signals
- Labels
- Training
- Prediction
  - Known Labels
  - Unknown Labels
- Analysis
  - Density classification (manual)
  - Density classification (model creation)
  - Classification (known labels)
  - Classification (unknown labels)
  - Anticipation time
  - Density histograms

After downloading several discharges and saving them in the temporary folder, the user must assess them through the "First Assessment" screen, shown in Figure 4.7, and decide whether to definitely save those shots or not. The user has the option to save discharges from the temporal data folder to the validated data folder, or to remove them, and visualize

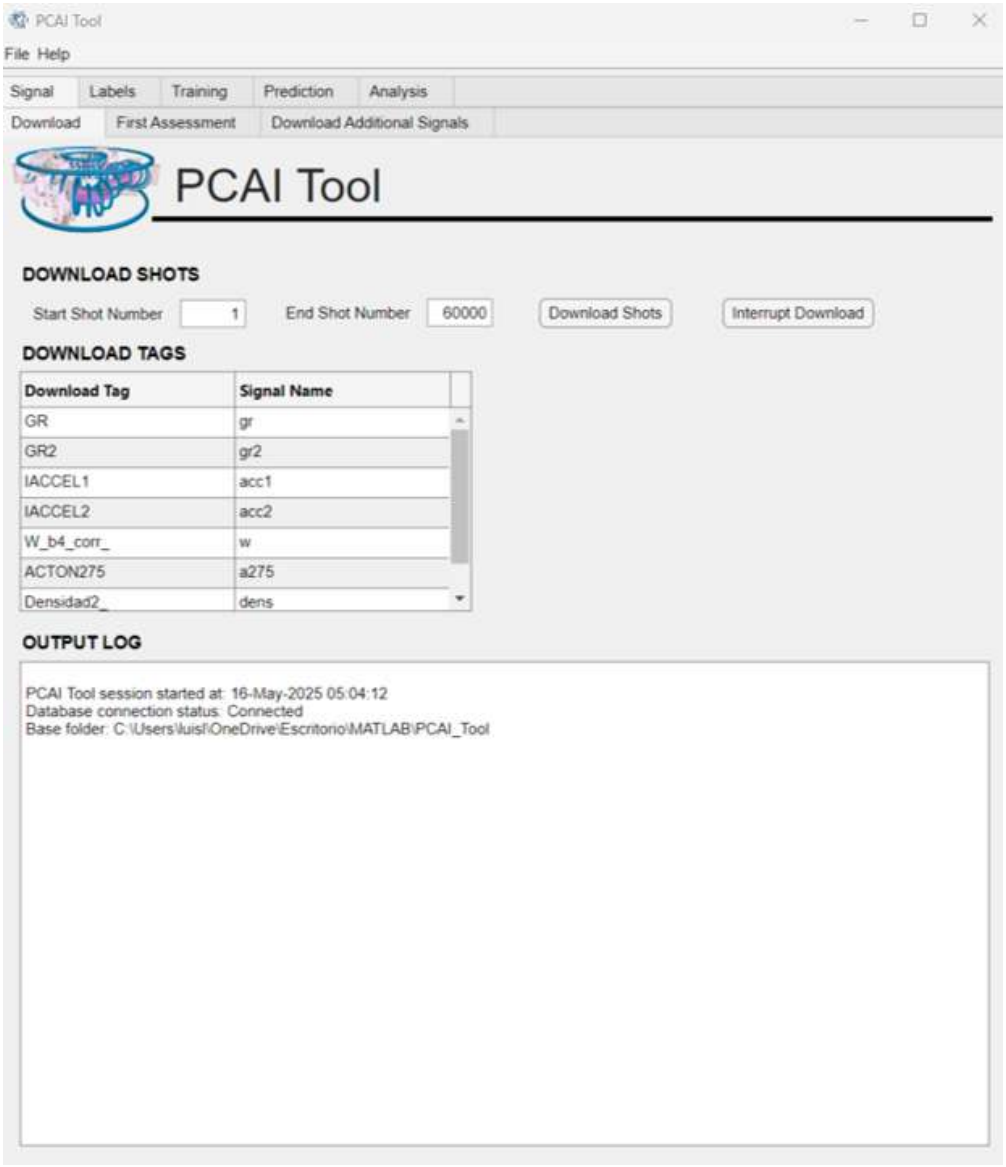

Figure 4.6: PCAI Tool Download screen, where the user can download discharges, configure download tags and visualize the log of the app.

them one by one in the plot. Definitive discharges are processed by the Signal Module and the data obtained is stored in the database thanks to the Data Module, which results into the "Definitive Shots Table with Characteristics". Each column of the mentioned table is described below.

- **Shot Nº**, discharge number.
- **Labeled**, label assigned to the discharge by the user through the "Labels" screen, which could be (-1) Collapsing, (0) Doubtful, (1) Non-Collapsing or (2) Unchecked.
- **NBI**, Indicates whether a shot has the NBI1 signal (1), the NBI2 signal (2) or both. This field is filled thanks to the algorithm presented in Subsection 3.1.2.

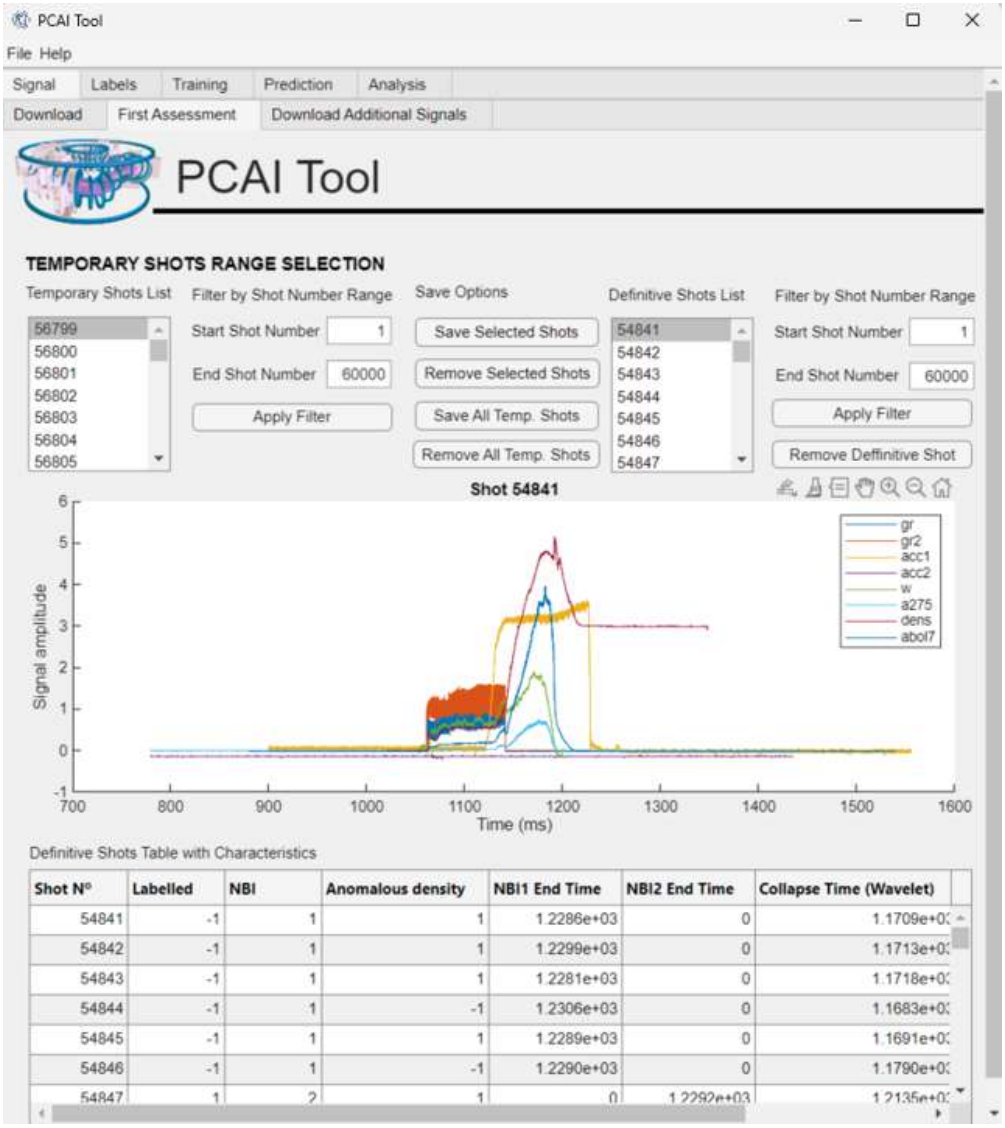

Figure 4.7: PCAI Tool First Assessment screen, where the user assess temporary discharges and decide whether to definitely save those shots or not.

- **Anomalous density**, label assigned to the density signal by the user through the "Density classification (manual)" screen, which could be (-1) Anomalous, (0) Doubtful, (1) Regular or (2) Unchecked.

- **NBI1 End Time**, instant in which the NBI1 signal has decayed to zero. This field is filled as an output of the algorithm presented in Subsection 3.1.2.

- **NBI2 End Time**, instant in which the NBI2 signal has decayed to zero. This field is filled as an output of the algorithm presented in Subsection 3.1.2.

- **Collapse Time (Wavelet)**, instant in which the energy signal starts to decay, determined by the discrete wavelet transform, DWT. This field is filled as an output of the algorithm presented in Subsection 3.1.1.

- **Collapse Time (Butterworth)**, instant in which the energy signal starts to decay, determined thanks to the Butterworth filter and an integral derivative algorithm. This field is filled as an output of the algorithm presented in Subsection 3.1.1.

If the user wants to download again the density signal for a specific discharge of the validated data folder, the "Download Additional Signals" screen allows this operation. When downloading again a density signal, the Download, Signal and Data modules coordinate to process this action.

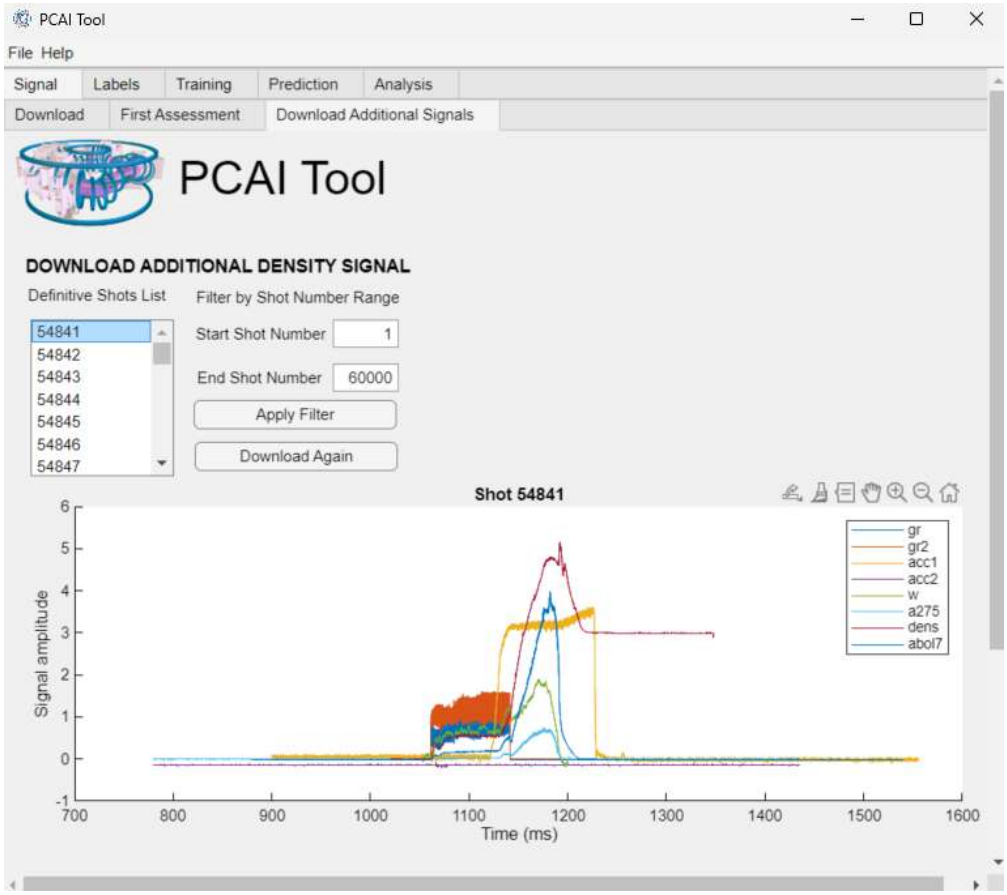

Figure 4.8: PCAI Tool Download Additional Signals screen, allows to download again the density signal for a specific discharge

To manually label discharges, the user can visualize and tag them as output of the "Labels" screen. The Data Module is in charge of storing the label that the user has assigned to each discharge in the database, in the "Shots Classification Table" which can be seen in Figure 4.9.

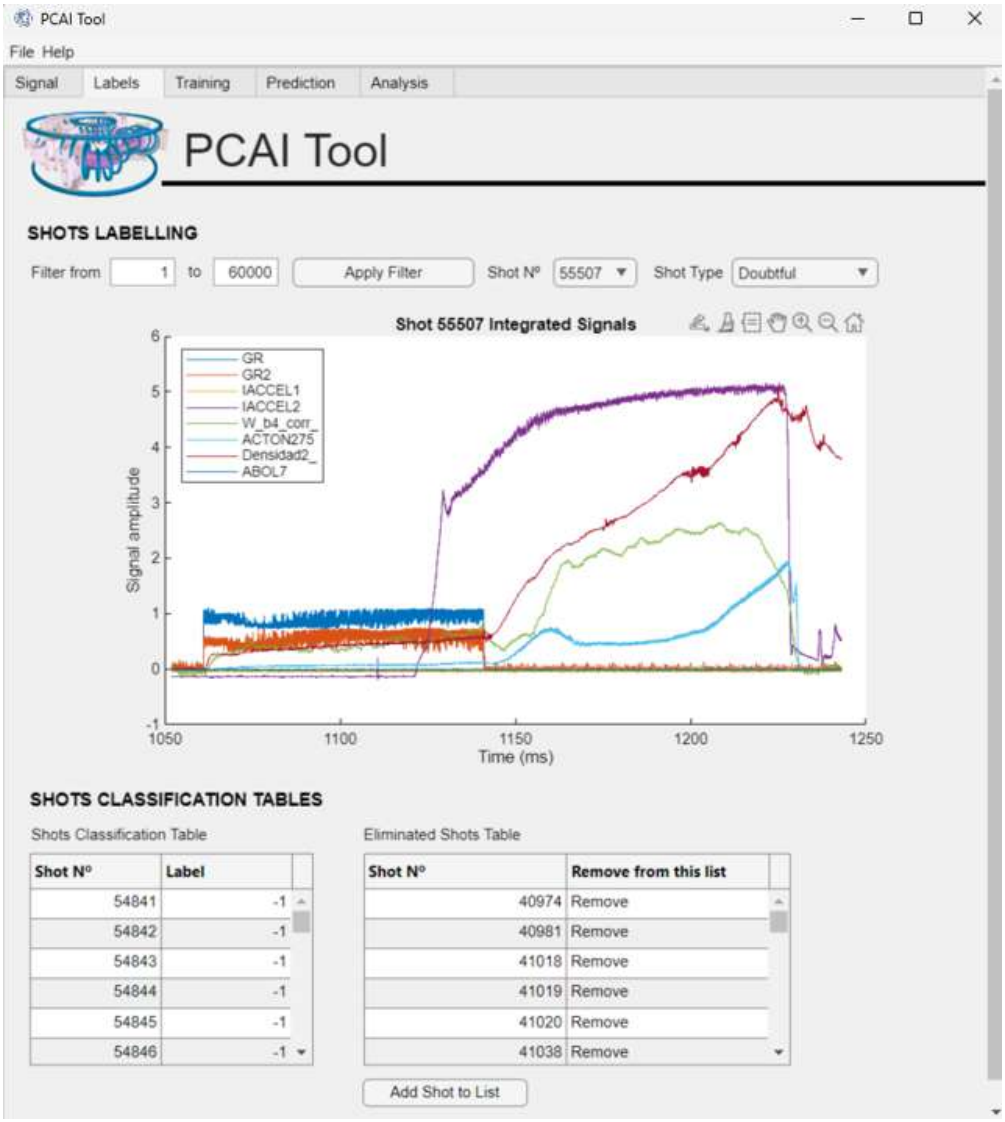

Figure 4.9: PCAI Tool Labels screen, allows to manually label discharges.

Taking into account that now every signal of each discharge of the validated data folder is successfully saved and the expert system, i.e. the user, has labeled enough discharges (for training and testing purposes), it is time to proceed with the training phase. To do so, the user must create a training file through the "Training" screen, Figure 4.10, which contains the feature vector, defined in Subsection 3.2.1, and its associated labels. After that, an SVM model can be trained. The module in charge of this process is the Training Module, and the training files are stored in the "Training Files" folder in ".mat" format.

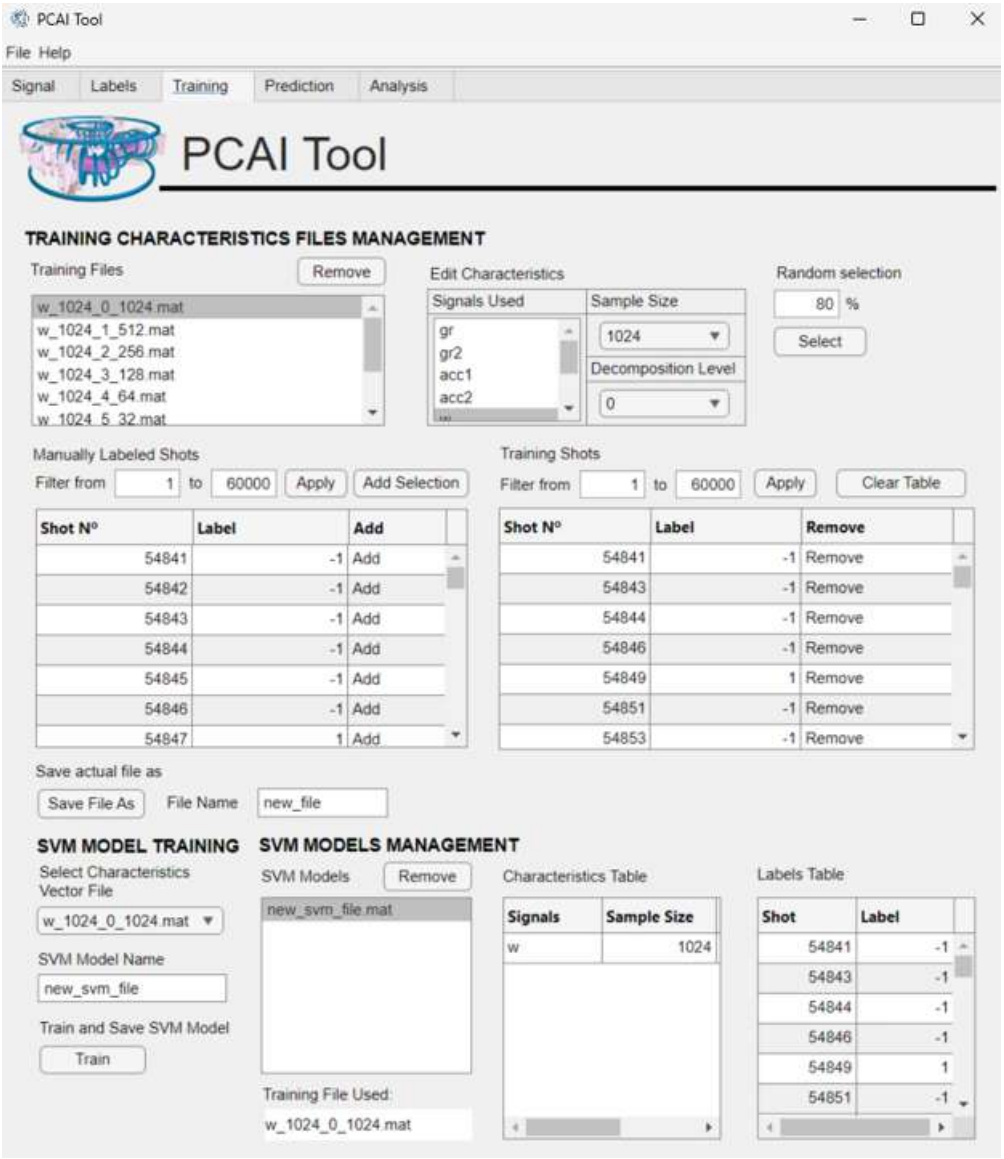

Figure 4.10: PCAI Tool Training screen, allows to configure feature vectors and train SVM models with them.

After the training phase, comes the testing. Thanks to the options given in the "Known Labels" screen, Figure 4.11, the user can evaluate each SVM model, inspect its accuracy, review the Collapsing (-1) probability assigned to every shot, and compare the labels assigned by the expert system with those predicted by the SVM classifier. The module in charge of this process is the SVM Module, and the test files are stored in the "Test Files" folder in ".mat" format.

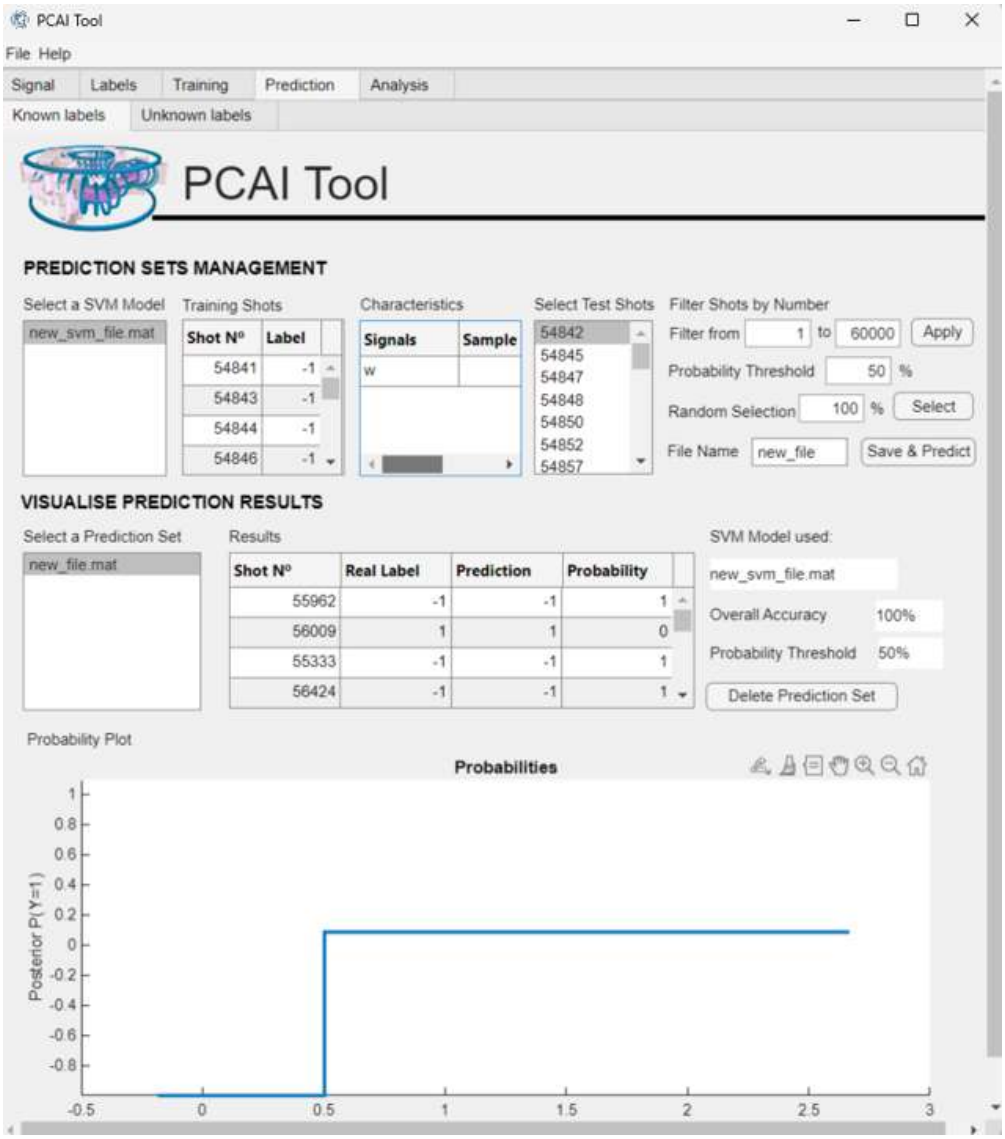

Figure 4.11: PCAI Tool Known Labels screen, allows to test SVM models.

The "Unknown Labels" screen allows the user to automatically predict unlabeled discharges, probabilities of each shot are given as well, as in the previous screen. The module in charge of this process is the Prediction Module, and the prediction files are stored in the "Prediction Files" folder in ".mat" format.

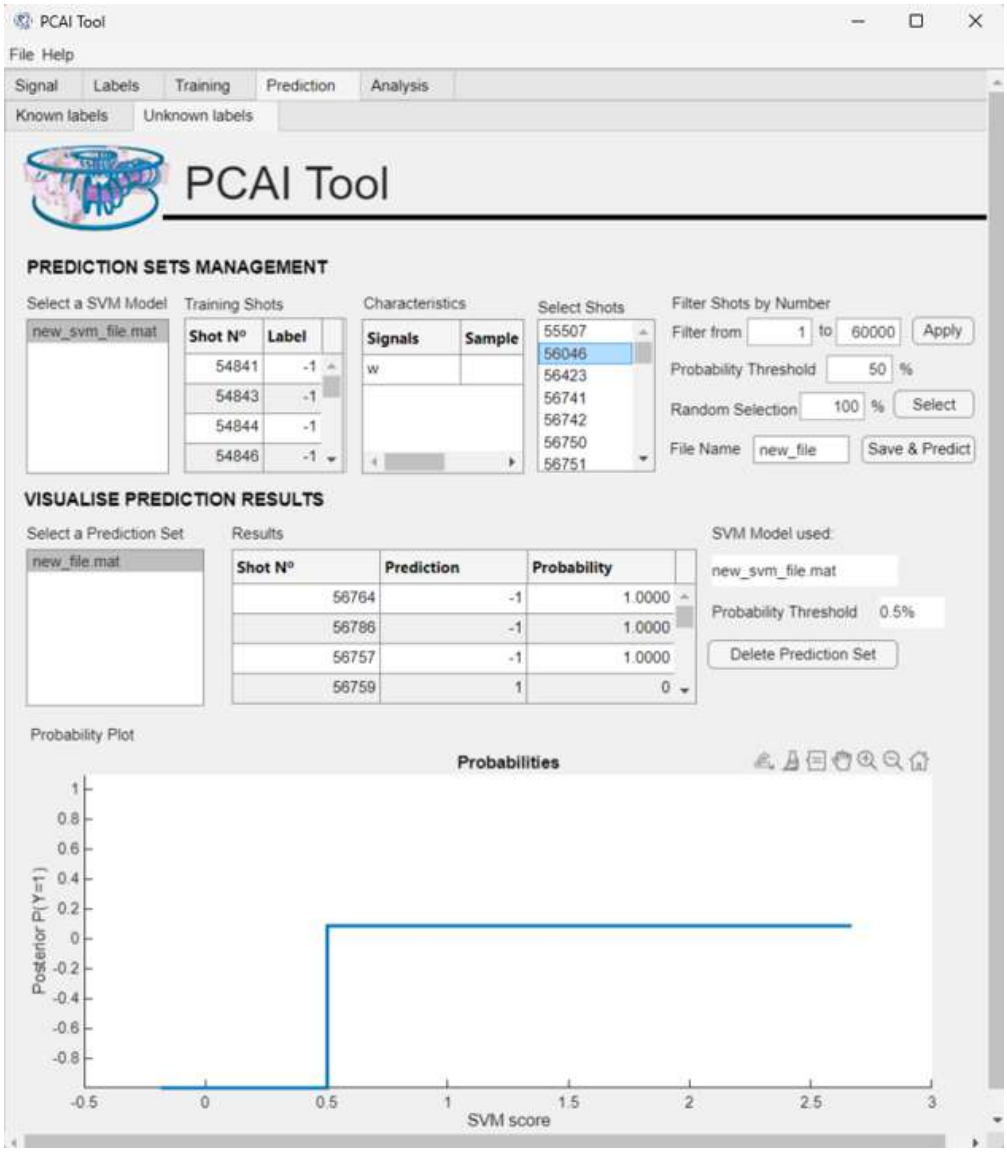

Figure 4.12: PCAI Tool Unknown Labels screen, allows to predict the tag of unlabeled discharges.

With the same philosophy as the "Labels" screen, the user can visualize and tag density signals from the "Density classification (manual)" screen. The Data Module is in charge of storing the label that the user has assigned to each discharge's density signal in the database, in the "Density Classification Table" which can be seen in Figure 4.13.

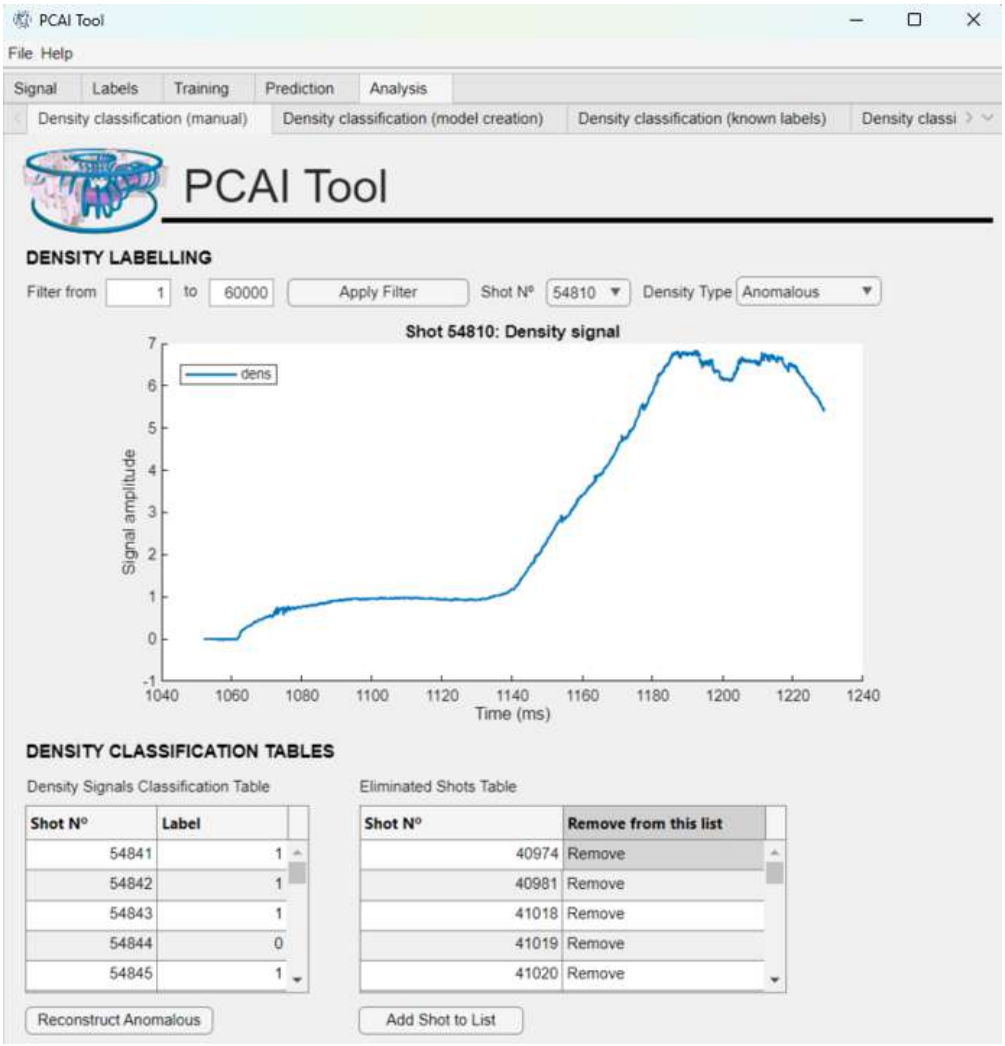

Figure 4.13: PCAI Tool Density classification (manual) screen, allows to manually label density signals.

After labeling the density signals, the procedure to train, test and predict is similar as for the discharges. Screens that enable these proceedings are defined below.

- **Train** → Density classification (model creation) → SVM Module
- **Test** → Classification (known labels) → Test Module
- **Predict** → Classification (unknown labels) → Prediction Module

The anticipation time can be defined as the difference between the instant of the end of the discharged, defined by the NBI signals as seen in Subsection 3.1.2, and the time of the collapse, defined in Subsection 3.1.1. By selecting one or several prediction files with their predicted shots and the manually labeled discharges of the database, the anticipation time of all the collapsing shots are presented in the histogram of the "Anticipation Time" screen, in Figure 4.14. The module in charge of this process is the Analysis Module, together with the Data Model, which retrieves the necessary data.

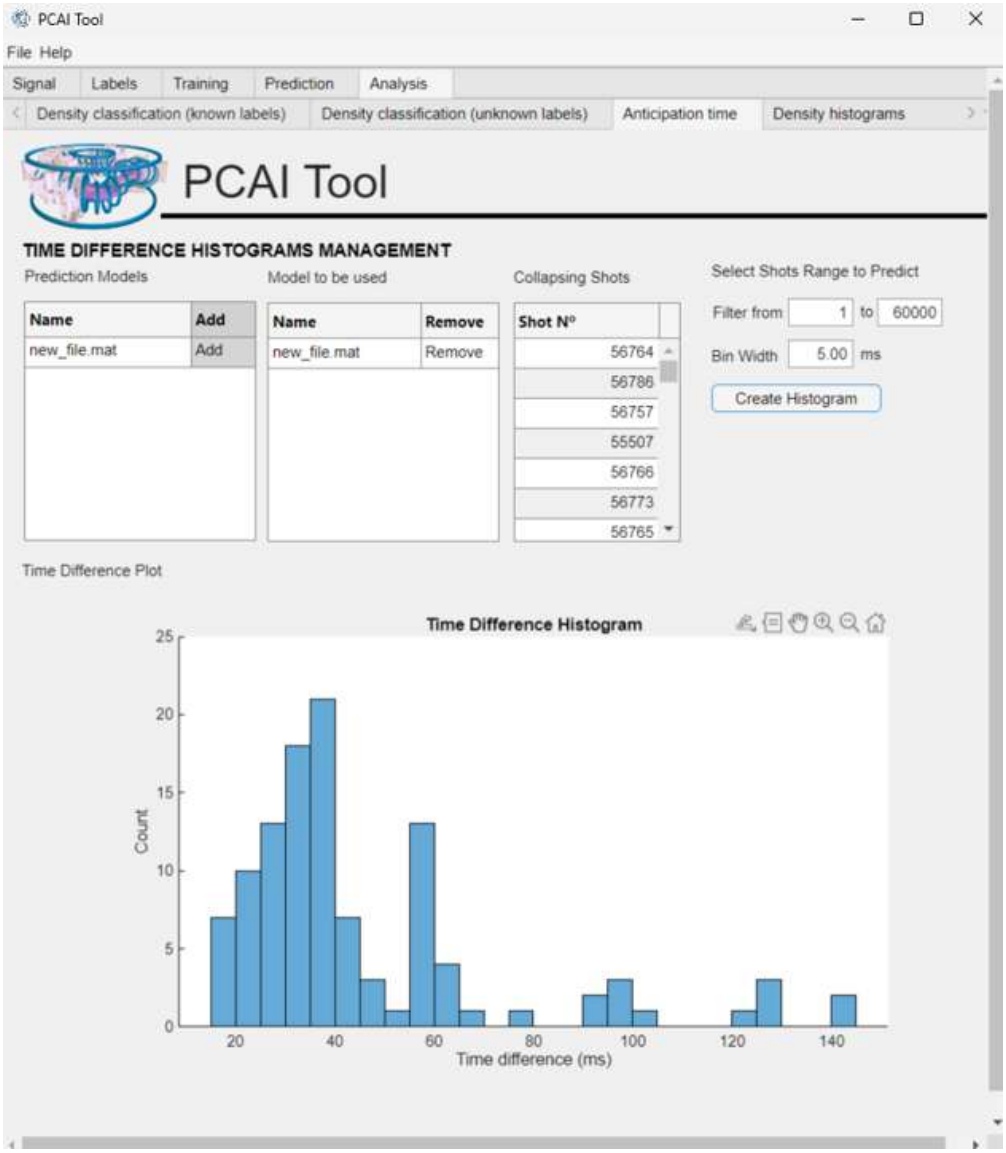

Figure 4.14: PCAI Tool Anticipation Time screen, allows select a prediction file and create a time difference histogram.

Density histograms, see Figure 4.15, are created by measuring the density signal at the time of the collapse, seen in Subsection 3.1.1. The discharges used for this case are the union of those predicted in the prediction files and those that have been manually labeled and stored in the database. The module in charge of this process is the Analysis Module, together with the Data Model, which retrieves the necessary data.

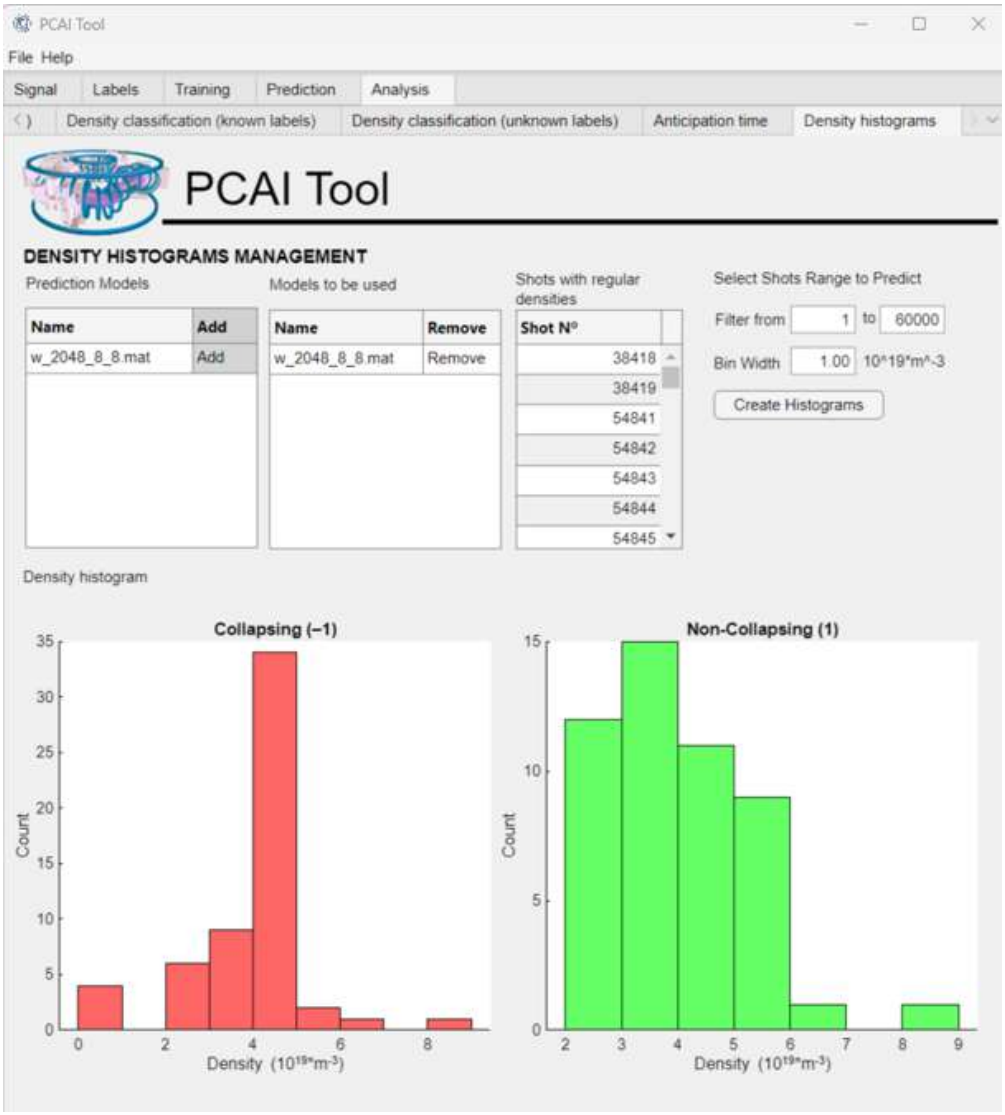

Figure 4.15: PCAI Tool Density Histograms screen, allows to create density histogram for collapsing and non-collapsing discharges.

This section described a brief but complete use case. For a complete in-depth description of the PCAI Tool usage, see the User Manual present in Appendix A.

4.3 SVM Classification Results

In this section, SVMs trained with different feature vectors are compared and evaluated with performance statistics.

4.3.1 Confusion matrix and derived metrics

A confusion matrix is a 2×2 contingency table that cross-tabulates the actual class of every instance against the predicted class produced by the classifier (Table 4.3). For a binary problem, which is the case for discharge classification, (-1) Collapsing and (1) Non-Collapsing, and density signal classification, (-1) Anomalous and (1) Regular, the four possible outcomes are:

TP (True Positives)	correctly predicted positives, (-1) Collapsing / Anomalous
TN (True Negatives)	correctly predicted negatives, (1) Non-Collapsing / Regular
FP (False Positives)	negatives (1) wrongly predicted as positive (-1),
FN (False Negatives)	positives (-1) wrongly predicted as negative (1).

As can be seen, in our case, positive refers to the tag (-1) because it is the one that wants to be detected mainly.

Table 4.3: Generic form of a confusion matrix.

Actual	**Predicted**	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

From the four counts one can derive the standard performance indicators used throughout this section:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.7}$$

$$F_1\text{-score} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}} = \frac{2\,\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.8}$$

Accuracy is the most typical performance metric, which stands for the total number of labels correctly predicted over the total number of predictions made. When the number of labels is unbalanced, this metric must be accompanied by other metrics, as the SVM model

could correctly predict only one of the labels and still obtain a high score. Precision is used when a low false-positive ratio must be taken into account. On the other hand, recall is used when a low false-negative ratio is needed. The mean of precision and recall is the $F_1 - score$. In our case, a high accuracy together with a good $F_1 - score$ is the best option.

4.3.2 Shots Classification Results

To study the effectiveness of the SVM classification, different feature vectors have been used to train it. 110 manually labeled discharges have been used for training all the SVM models and other 110 manually labeled discharges have been used for testing. For training, the same set of discharges has been used for all SVM models, thus avoiding randomness and allowing us to uniformly assess the accuracy of the models. Therefore, the other set of discharges used to test these models will be the same between models. Once trained, testing results with the metrics mentioned before are compiled in Table 4.4. Here, the Energy and ACTON275 signals are used as they provide collapse symptoms when they decay.

Among all the options tested, the one that has the highest accuracy and $F_1 - score$ is the SVM model trained with the Energy signal, a sample size of 2048 and a decomposition level of 8, achieving an accuracy of 96.4% and an $F_1 - score$ of 93.75%. Due to the fact that this model is the best of the options tested, it will be used in Subsection 4.4.1 for the anticipation time analysis.

4.3.3 Density Classification Results

To study the effectiveness of the SVM classification when it comes to the density signal classification, different feature vectors have been used to train it. 86 manually labeled signals have been used for training all the SVM models and other 86 manually labeled signals have been used for testing. For training, the same set of density signals has been used for all SVM models, thus avoiding randomness and allowing us to uniformly assess the accuracy of the models. Therefore, the other set of density signals used to test these models will be the same between models. Once trained, testing results with the metrics mentioned before are compiled in Table 4.5. Here, the feature vectors are defined by the DWT coefficient and its associated decomposition level. Among all the options tested, the one that has the highest accuracy and $F_1 - score$ is the SVM model trained with detail coefficients and a decomposition level of 12 achieving an accuracy of 73.26% and an $F_1 - score$ of 80.67%. Owing to the fact that this model is the best of the options tested, it will be used in Subsection 4.4.2 for the density analysis.

Table 4.4: Classification metrics obtained on the test set for every SVM model, grouped by signals used, original sample size (N) and wavelet decomposition level (L).

Signals Used	N	L	Accuracy	Precision	Recall	F_1
Energy	512	0	0.9369	0.8788	0.9062	0.8923
Energy	512	1	0.9459	0.9062	0.9062	0.9062
Energy	512	5	0.9459	0.9333	0.8750	0.9032
Energy	512	8	0.9459	0.9333	0.8750	0.9032
Energy	1024	0	0.9550	0.9355	0.9062	0.9206
Energy	1024	1	0.9550	0.9091	0.9375	0.9231
Energy	1024	5	0.9459	0.8824	0.9375	0.9091
Energy	1024	8	0.9550	0.9355	0.9062	0.9206
Energy	2048	0	0.9459	0.9333	0.8750	0.9032
Energy	2048	1	0.9459	0.9333	0.8750	0.9032
Energy	2048	5	0.9459	0.9333	0.8750	0.9032
Energy	2048	8	0.9640	0.9375	0.9375	0.9375
Energy + ACTON275	512	0	0.9189	0.9394	0.8158	0.8732
Energy + ACTON275	512	1	0.9369	0.9429	0.8684	0.9041
Energy + ACTON275	512	5	0.9369	0.9697	0.8421	0.9014
Energy + ACTON275	512	8	0.9369	0.9697	0.8421	0.9014
Energy + ACTON275	1024	0	0.9459	0.9444	0.8947	0.9189
Energy + ACTON275	1024	1	0.9459	0.9444	0.8947	0.9189
Energy + ACTON275	1024	5	0.9189	0.9394	0.8158	0.8732
Energy + ACTON275	1024	8	0.9459	0.9706	0.8684	0.9167
Energy + ACTON275	2048	0	0.8649	1.0000	0.6053	0.7541
Energy + ACTON275	2048	1	0.8739	1.0000	0.6316	0.7742
Energy + ACTON275	2048	5	0.8739	1.0000	0.6316	0.7742
Energy + ACTON275	2048	8	0.9459	0.9706	0.8684	0.9167

Table 4.5: Performance metrics of the density–signal classifier for each combination of wavelet coefficient type and decomposition level.

Coefficients	L	Accuracy	Precision	Recall	F_1
Approximation	1	0.6860	0.6667	0.9412	0.7805
Approximation	6	0.7093	0.7241	0.8235	0.7706
Approximation	12	0.7209	0.6957	0.9412	0.8000
Detail	1	0.5930	0.5930	1.0000	0.7445
Detail	6	0.7326	0.7059	0.9412	0.8067
Detail	12	0.7326	0.7059	0.9412	0.8067

4.4 Analysis

In this section results regarding analysis based on histograms are presented.

4.4.1 Anticipation Time Analysis

The aim of the anticipation time histogram is to detect a tendency among the time difference between the collapse time, i.e. the time in which the energy signal has definitely decayed as evaluated in Subsection 4.1.1, and the end time of the discharge, as assessed in Subsection 4.1.2. Thanks to the PCAI Tool, the histogram in Figure 4.16 has been created, taking into account the manually labeled collapsing discharges and the collapsing discharges predicted by the model achieved in Subsection 4.3.2.

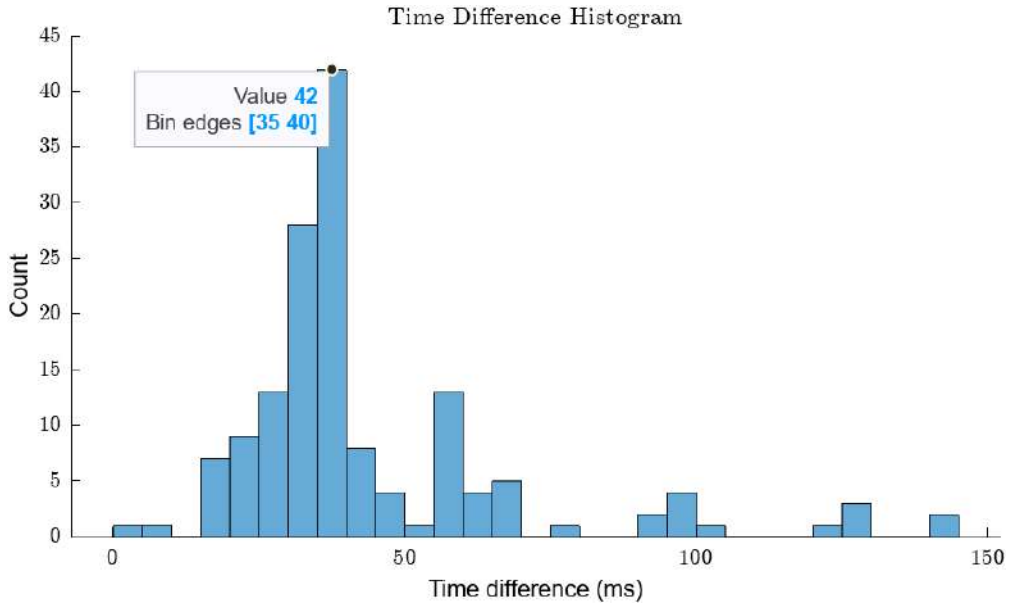

Figure 4.16: Collapsing discharges time difference histogram.

The highest count takes place between 35 ms and 40 ms, so that, this specific time lapse is critical for anticipating a collapse.

4.4.2 Density Analysis

Density analysis aims for establishing a density limit between collapsing discharges and non-collapsing ones. To do so, the value of the density signal at the moment of the collapse is measured for every discharge of the database in which the density signal is labeled as regular. With the PCAI Tool, the histogram in Figure 4.17 has been created, where the density values and their counts can be seen, both for collapsing discharges (red) and for non-collapsing discharges (green).

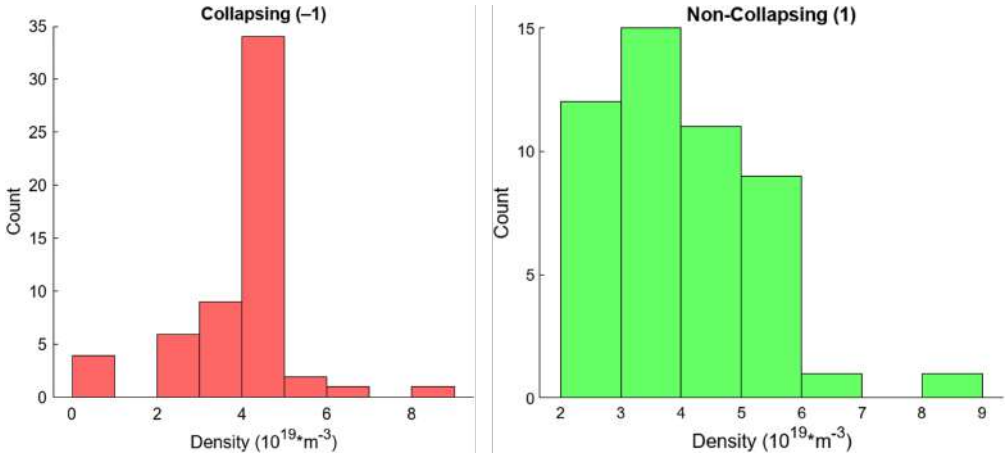

Figure 4.17: Density histograms.

Chapter 5

Conclusions and Future Work

In this final chapter, conclusions and future work are presented.

5.1 Conclusions

Thanks to the development of this work, the following goals have been met, which are of great importance to the LNF final users.

- Automated the analysis of the signals coming from the TJ-II.
- Automated collapsing/not collapsing discharge classification on the TJ-II database of 60000 shots.
- Compiled the PCAI Tool, so that every final user of the LNF could easily install it without the need of a MATLAB license.
- Developed the density signal reconstruction algorithm seen in Subsection 3.1.3, palliating the fringe jump phenomenon, which happens in interferometry diagnostics, setting an unprecedented approach.
- Implemented user-configurable installation settings for signal download, allowing future updates and the inclusion of new signals.
- Created an SVM classifier with an accuracy of 96.4% and an $F_1 - score$ of 93.75%.

5.2 Future Work

During the operation of the PCAI Tool, several aspects have been detected as possible future work.

- Integrate the application into the TJ-II remote participation system, a web-based platform secured by a distributed authentication and authorization framework. In this way, the comparison of results from other stellarators with the predictions derived from TJ-II would be enhanced.

- Within the framework of the collaboration between TJ-II and W7-X, the PCAI tool application could be installed in the experimental environment of W7-X, at present the largest stellarator in operation. This will be discussed with task for leaders of this device.

- An optimization algorithm, such as Simulated Annealing, can be implemented, so that the best-performing feature vector of each type of SVM, the one that classifies discharges and the one that detects anomalous density signals, can be found. This algorithm could take few hours to finish, but the final result will save hours to the final user trying to find the best feature vectors.

- Real-time collapse detection algorithms could be developed and implemented within the framework of the PCAI Tool.

- Thanks to the PCAI Tool capabilities, an analysis of possible scaling laws can be carried out with the aim of studying the density limit and extrapolating it to other devices and future fusion reactors.

- This work will contribute to the Plasma 2025 – International Conference on Research and Application of Plasmas

  Dates: September 15–19, 2025
  Location: Warsaw, Poland

  Title of the contribution:
  Recognition of two patterns of ion-temperature profiles in the TJ-II stellarator after the upgrade of a multichannel spectrometer

  Authors:
  B. López-Miranda[1], L. López-Sánchez[2], D. Amador[2], J. Vega[1], S. Dormido-Canto[3], A. Baciero[1], I. Pastor[1], and the TJ-II team

  Affiliations:
  [1] Laboratorio Nacional de Fusión, CIEMAT, Spain
  [2] Departamento de Arquitectura de Computadores y Automática, UCM, Spain
  [3] Departamento de Informática y Automática, UNED, Spain

Bibliography

[1] F. F. Chen and G. Weisel. An Indispensable Truth: How Fusion Power can save the Planet. *American Journal of Physics*, 79(12):1276–1277, December 2011. doi: 10.1119/1.3636656.

[2] J. Wesson and D. J. Campbell. *Tokamaks*, volume 149 of *International Series of Monographs on Physics*. Oxford University Press, 4 edition, 2011. ISBN 978-0-19-959223-4.

[3] L. Spitzer, Jr. The stellarator concept. *The Physics of Fluids*, 1(4):253–264, 07 1958. ISSN 0031-9171. doi: 10.1063/1.1705883.

[4] J. D. Lawson. Some criteria for a power producing thermonuclear reactor. *Proceedings of the Physical Society. Section B*, 70(1):6, jan 1957. doi: 10.1088/0370-1301/70/1/303.

[5] M. A. Leontovich, J. B. Sykes, J. Turkevich, et al. Plasma physics and the problem of controlled thermonuclear reactions, volume 3. *Physics Today*, 13(7):48–50, 07 1960. ISSN 0031-9228. doi: 10.1063/1.3057041.

[6] G. Casini. Fusion technology (Report on the 13th Symposium, Varese, Italy, 24–28 September 1984). *Nuclear Fusion*, 25(5):633, may 1985. doi: 10.1088/0029-5515/25/5/007.

[7] ITER. History of the ITER, 2025. URL https://www.iter.org/about/history.

[8] L. M. Goldman and L. Spitzer, Jr. Preliminary Experimental Results with the Model A Stellarator. Technical report, Princeton Univ., N.J. Project Matterhorn, 05 1953. URL https://www.osti.gov/biblio/4285864.

[9] C. Alejaldre, J. Alonso, L. Almoguera, et al. First Plasmas in the TJ-II Flexible Heliac. *Plasma Physics and Controlled Fusion*, 41(3A):A539, mar 1999. doi: 10.1088/0741-3335/41/3A/047.

[10] E. Ascasíbar, T. Estrada, F. Castejón, et al. Magnetic Configuration and Plasma Parameter dependence of the Energy Confinement Time in ECR heated plasmas from the TJ-II Stellarator. *Nuclear Fusion*, 45(4):276, mar 2005. doi: 10.1088/0029-5515/45/4/009.

[11] J. Vega, A. Murari, S. Dormido-Canto, et al. Disruption prediction with artificial intelligence techniques in tokamak plasmas. *Nature Physics*, 18(7):741–750, July 2022. doi: 10.1038/s41567-022-01602-2.

[12] S. Dormido-Canto, G. Farias, R. Dormido, et al. TJ-II Waveforms Analysis with Wavelets and Support Vector Machines. *Review of Scientific Instruments*, 75(10):4254–4257, 2004. doi: 10.1063/1.1787611.

[13] S. Dormido-Canto, J. Vega, J. Sánchez, et al. Information Retrieval and Classification with Wavelets and Support Vector Machines. In *Proceedings of IWINAC 2005*, 2004. doi: 10.1007/11499305_56.

[14] J. Vega, I. Pastor, J. L. Cereceda, et al. Application of Intelligent Classification Techniques to the TJ-II Thomson Scattering Diagnostic. In *32nd EPS Conference on Plasma Physics*, 2005. doi: 10.5281/zenodo.4568120.

[15] G. Farias, R. Dormido, M. Santos, et al. Image classifier for the TJ-II Thomson Scattering Diagnostic: Evaluation with a Feed Forward Neural Network. In *Proceedings of IWINAC 2006*, 2006. doi: https://doi.org/10.1007/11767954_38.

[16] P. S. Addison. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*. CRC Press, Boca Raton, 2 edition, 2017. ISBN 9781315372556. doi: 10.1201/9781315372556.

[17] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Ltd, 2nd edition, 2007. ISBN 9780470140529. doi: 10.1002/9780470140529.ch9.

Appendix A

PCAI Tool User Manual

A.1 Installation Guide

The prerequisites for a successful installation of the PCAI Tool are listed below:

- The web installer executable, `PCAI_Tool_web_installer.exe`.
- A PC meeting the recommended specifications in Table A.1.

Table A.1: PC Specifications

Component	Specification
Operating system	Windows 10 or 11
Processor	AMD Ryzen 7 5700U with Radeon Graphics, 1.80 GHz
RAM Memory	16.0 GB
Disk space required	3.13 GB
System type	64-bit operating system, x64-based processor

To begin the installation and initial configuration of the PCAI Tool, the following steps must be completed.

1. Execute the web installer, Figure A.1.

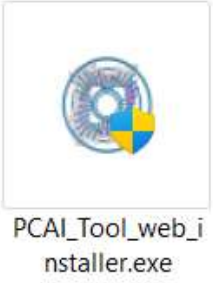

Figure A.1: PCAI Tool web installer executable.

2. In the installation wizard, click “Next / Siguiente” to continue, Figure A.2.

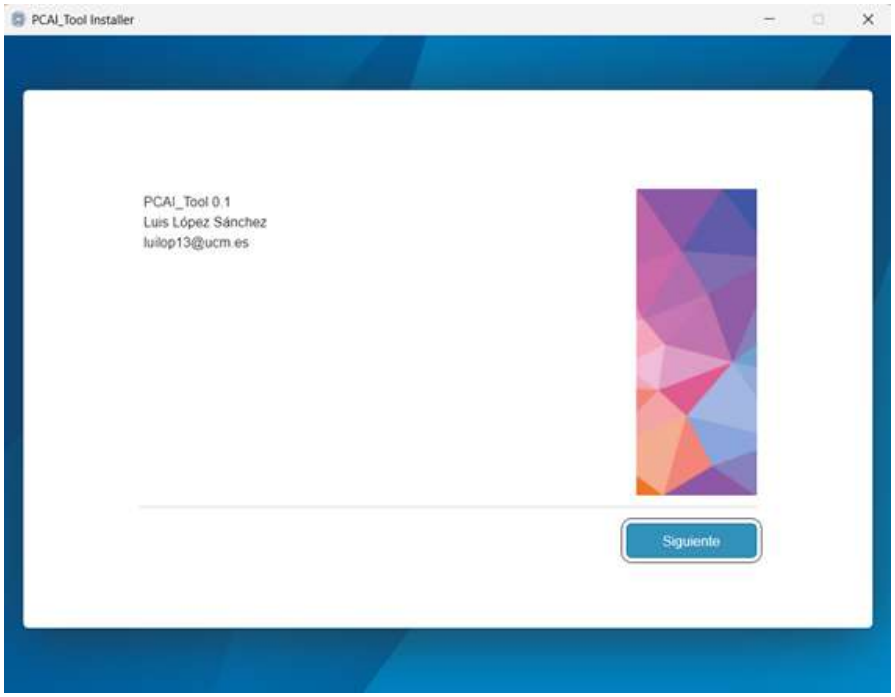

Figure A.2: Installation wizard welcome screen.

3. Select the target folder and enable creation of a desktop shortcut, then click “Next / Siguiente” Figure A.3.

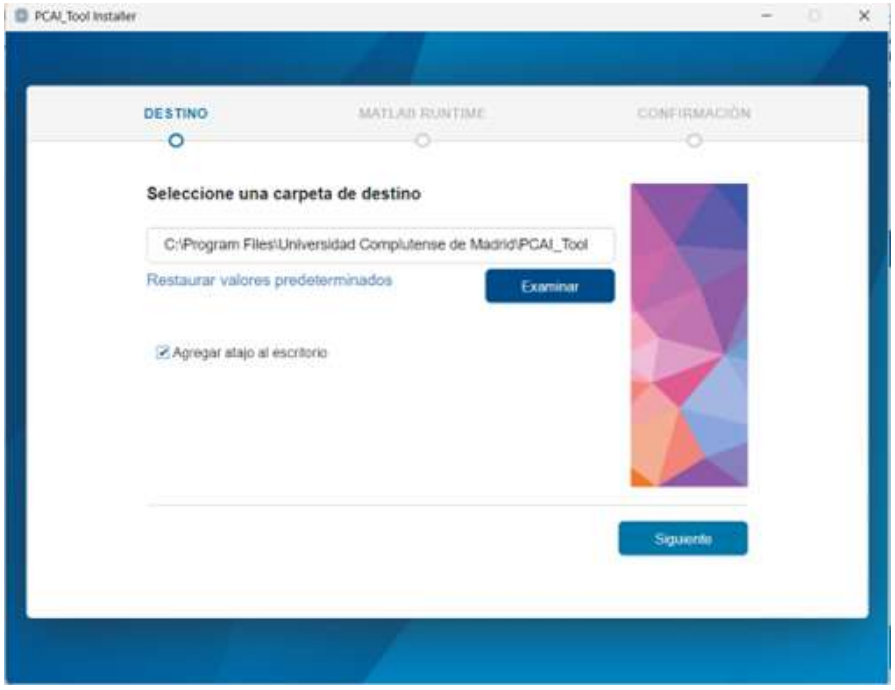

Figure A.3: Installation directory and desktop shortcut screen.

4. Press “Begin Installation / Comenzar Installación” Figure A.4.

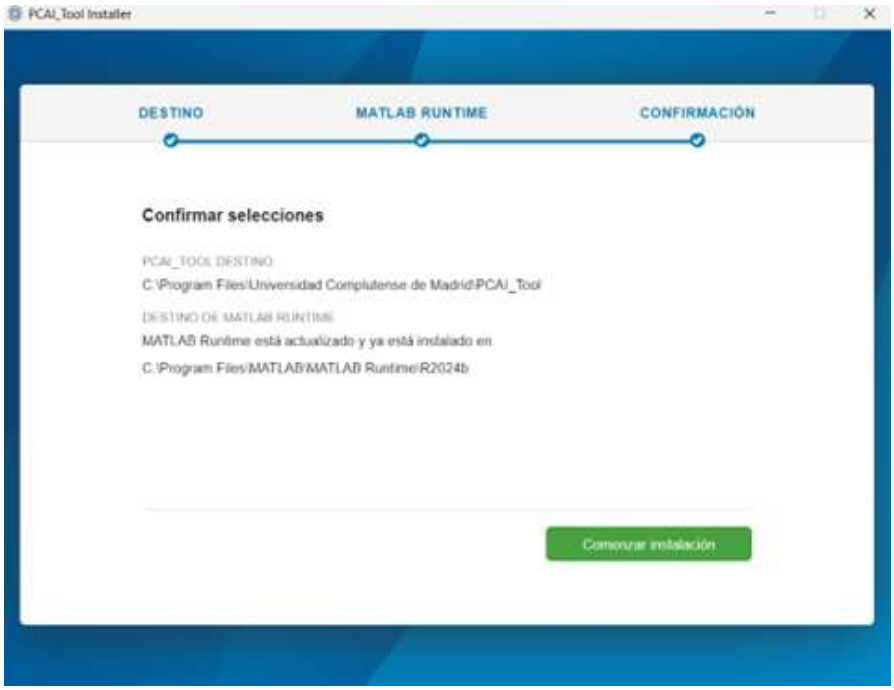

Figure A.4: Begin installation screen.

5. Installation completed, press “Close / Cerrar” Figure A.5.

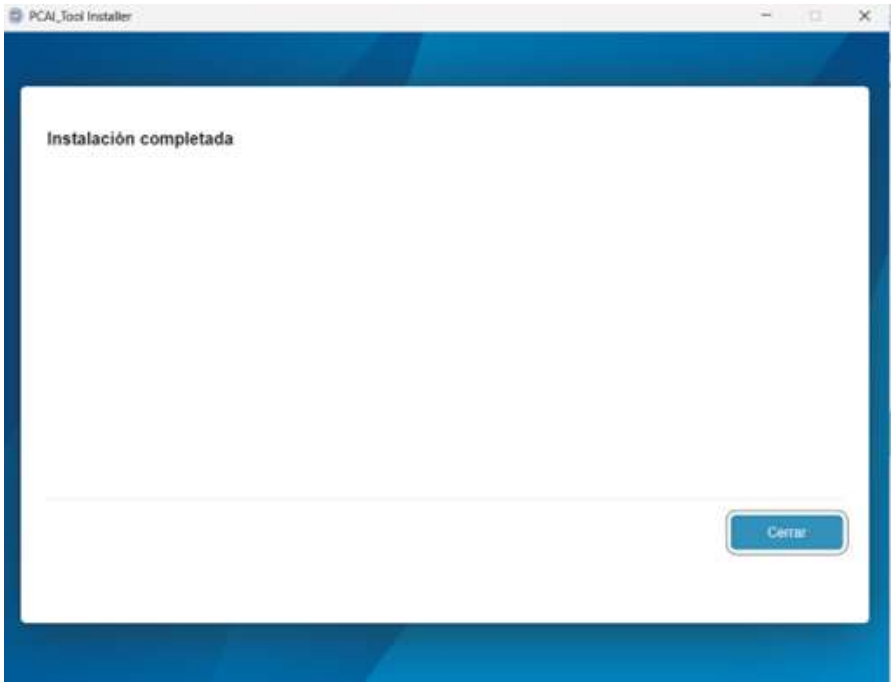

Figure A.5: Installation completed screen.

6. After completion, locate the application shortcut on the desktop, Figure A.6.

Figure A.6: PCAI Tool desktop shortcut.

7. Launch the PCAI Tool by double-clicking its desktop icon. On first run, the initial signal selection dialog appears. Accept the default options or add additional signals, then click "Finish PCAI Tool Installation," Figure A.7. The "Download Tag" of a signal is the nickname with which the PCAI Tool asks the TJ-II web server to download the signal. On the other hand, the "Signal Name" is the alias used for that signal within the application. "Download Tags" can be changed during installation and once installed, "Signal Names" can only be changed for additional signals during installation, once installed they can not be changed.

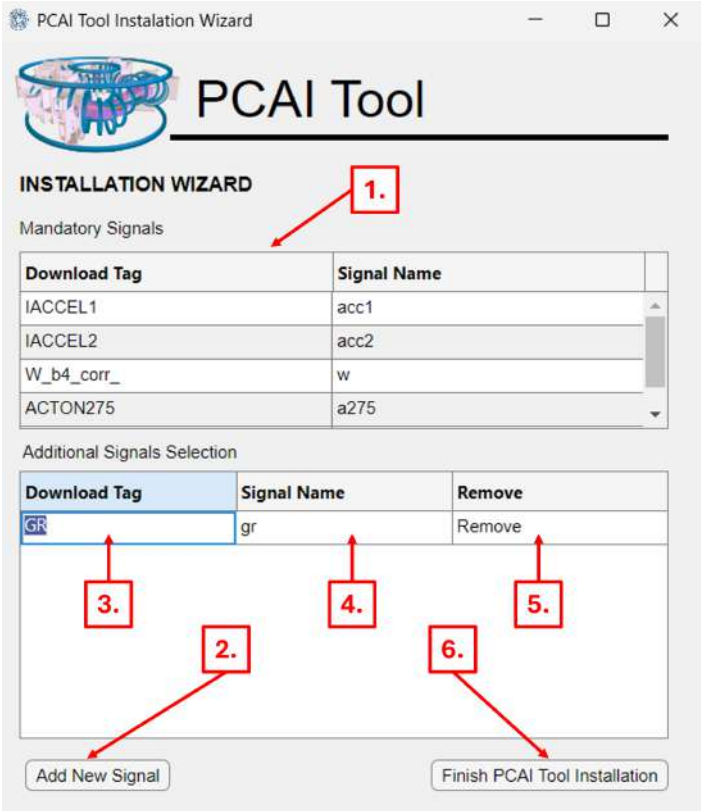

Figure A.7: Initial signal selection dialog wizard.

In Figure A.7, table (1.) allows the user to see the mandatory signals and change their download tag. In the "Additional Signal Selection" table, the user can modify the "Download Tag" (3.), the "Signal Name" (4.), or remove that signal double clicking in

the cell "Remove" (5.). To add a new signal the user must press the "Add New Signal" button (2.). Once signal configuration is completed, the user must press the "Finish PCAI Tool Installation" button (6.).

8. The main application window will open and create its data directories. Confirm successful creation via the log panel, Figure A.8.

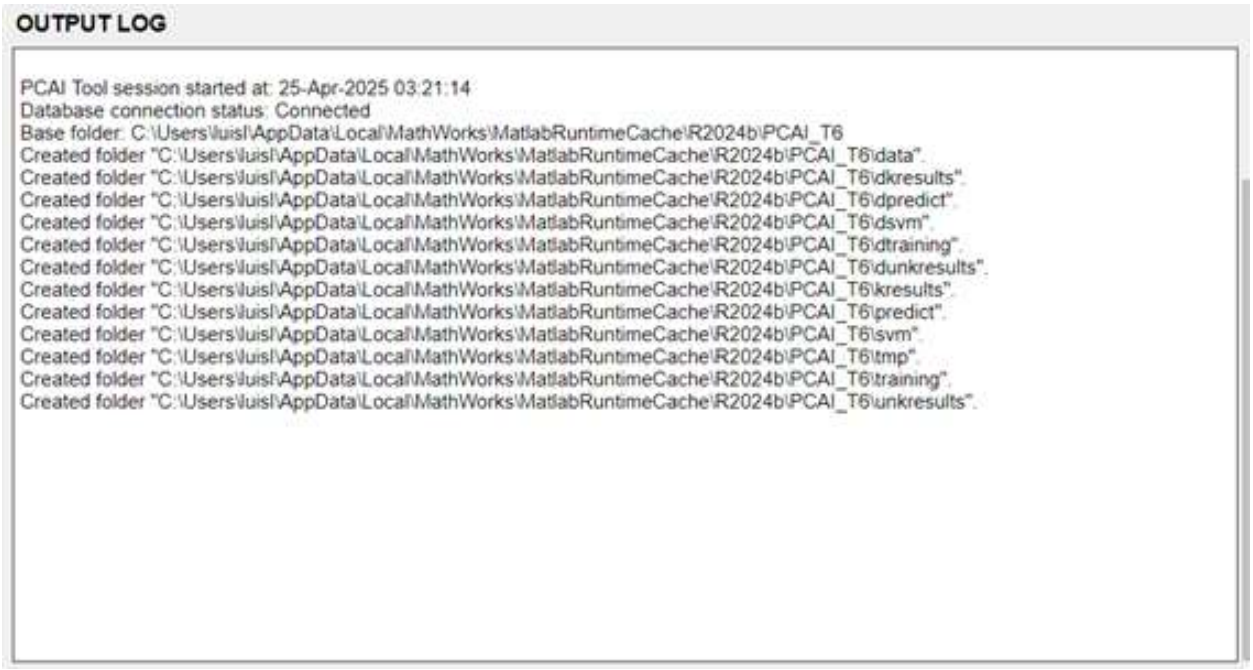

Figure A.8: Main window log.

After this step, the installation of the PCAI Tool is finally completed.

A.2 User Guide

To launch the PCAI Tool, double-click on the desktop icon, Figure A.9.

Figure A.9: PCAI Tool desktop shortcut.

A.2.1 Main Screen - Signal/Download

Once launched, the main screen appears, Figure A.10, where the user can change the "Download Tag" of every signal, specify a range of shots to start downloading them, and visualize the application log, where every activity is registered.

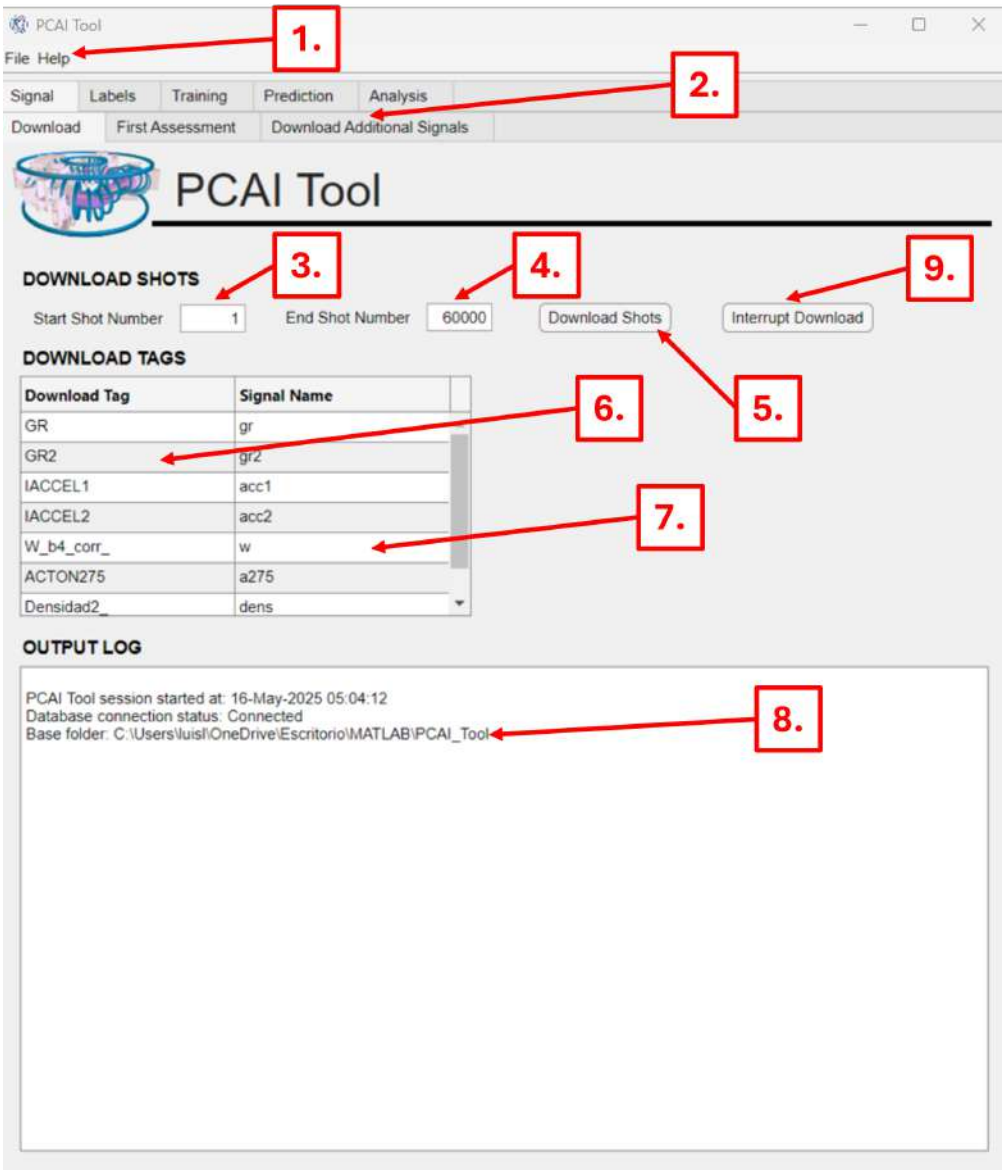

Figure A.10: PCAI Tool main screen, which corresponds to the "Signal/Download" tab.

In Figure A.10 screen, the user can set a minimum (3.) and a maximum (4.) shot values to configure the download range of shots. Once set, the "Download Shots" button (5.) can be pressed so that shots within the range begin to download (this process, depending on how wide the range is, can last minutes). Download tags of the signals can be changed by editing the corresponding cell of the "Download Tag" column (6.) from the "Download Tags" table. To interrupt a download process, press the "Interrupt Download" button (9.). If the user

press the download button (5.) to start the download process, the message in Figure A.11 will appear to ensure that the download shot range is correct.

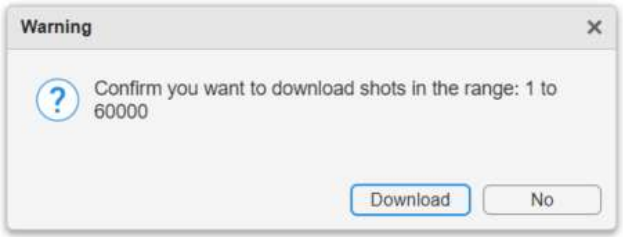

Figure A.11: Discharge range confirmation.

On the other hand, in Figure A.10, Signal names from column "Signal Names" (7.) can not be edited. In the "Output Log" (8.), every activity of the PCAI Tool is registered. In the upper menu bar (1.), the following menus and submenus can be found.

- File
  - Save log to file
- Help

By clicking in the "Save log to file" submenu, the log can be saved in a ".txt" file format with the form `pcai_log_YYYY_MM_DD_hh_mm.txt`, as shown in Figure A.12. Meanwhile, by selecting the "Help" menu, this exact user manual appears in ".pdf" format.

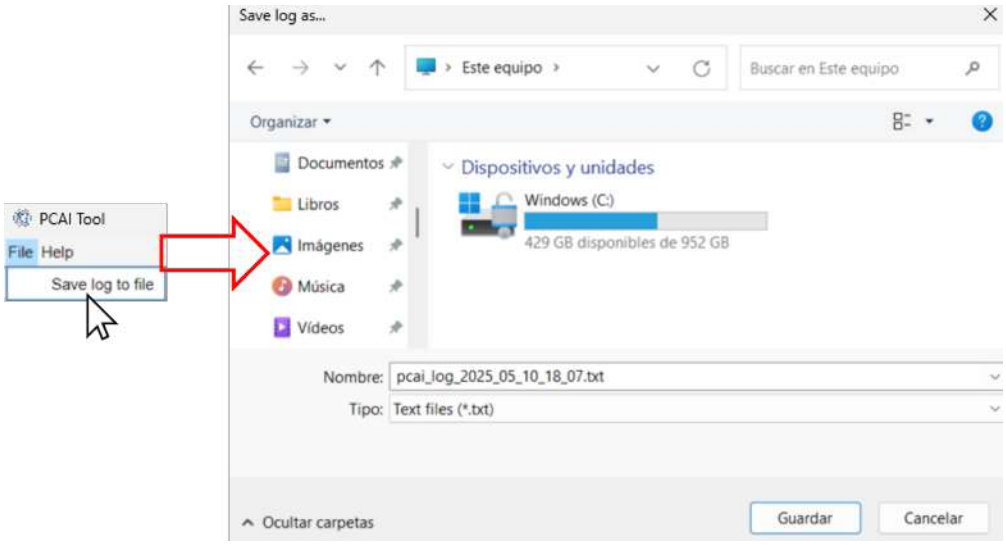

Figure A.12: Save log to ".txt" file.

The other option to save the log is that when the application is being closed, the user is asked whether to save the log in a ".txt" file or not, as shown in Figure A.13.

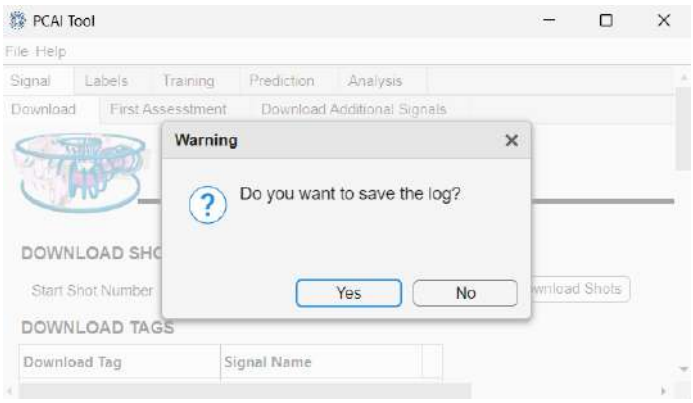

Figure A.13: Save request.

Finally, in point (2.) in Figure A.10, the user can select the screen that is being visualized. All screens are described in the following subsections.

A.2.2 Signal/First Assessment

The "First Assessment" screen, shown in Figure A.14 is displayed when the "Signal/First Assessment" tab is clicked. The options available in this tab allow the user to assess the downloaded shots, as they are, at a first instance, saved within temporal folder called "\tmp". Proper shots that are adequate to be used are saved within a definitive folder called "\data".

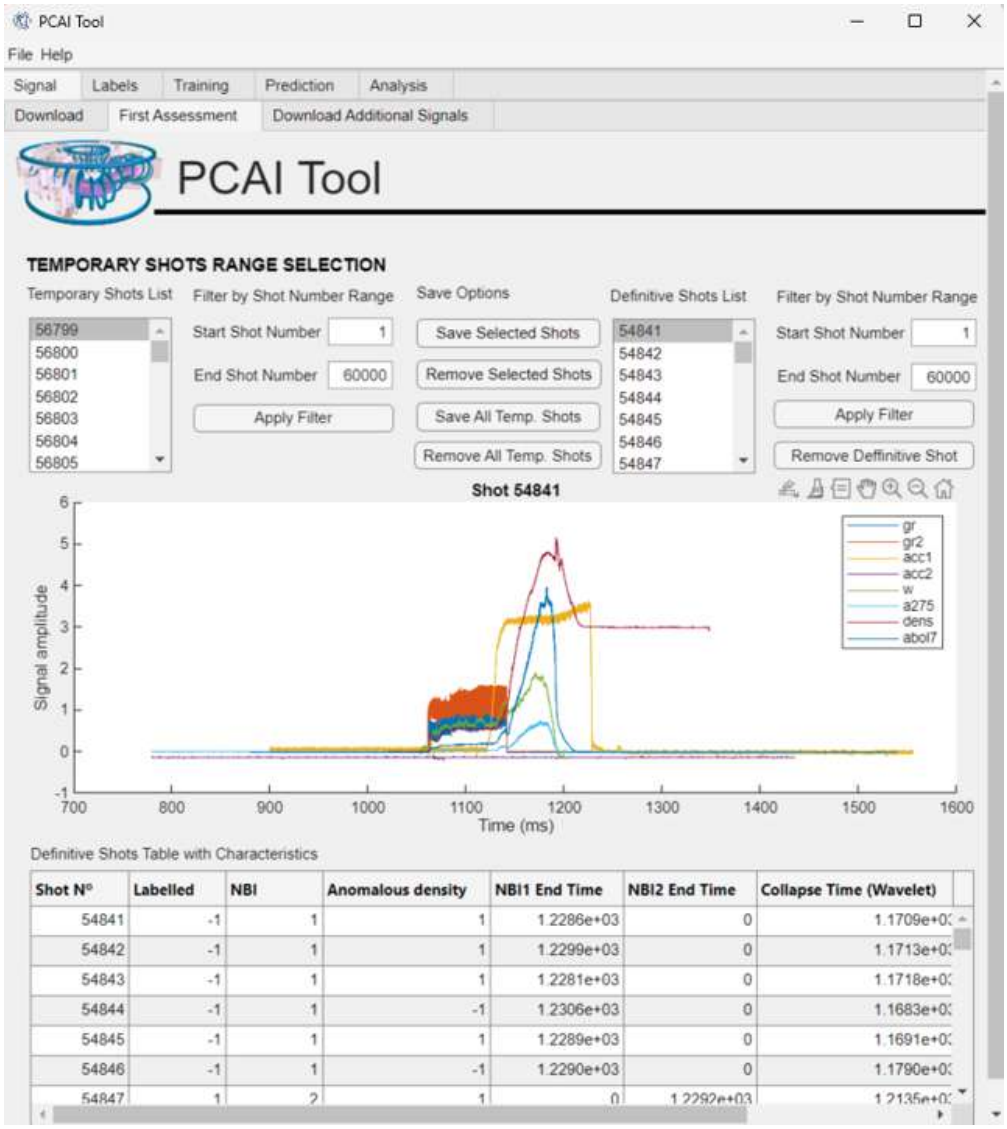

Figure A.14: First Assessment Screen.

The first step to evaluate a shot is to visualize it, to do so, as can be seen in Figure A.15, by double-clicking a shot on the "Temporary Shot List" (1.), all its signals appear in the plot of the screen, Figure A.14. See Figure A.15, to filter the shots of the list (1.) the user can define a minimum (2.) and maximum (3.) range and apply the filter by clicking the toggle button (4.). Once clicked, the filter can be disabled by clicking again on the toggle button.

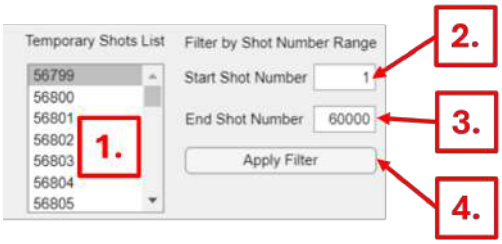

Figure A.15: "Temporary Shots List" filter options.

In Figure A.16, the options available allow the user to save all selected shots from the "Temporary Shots List", which is multiselectable (the user can click on a shot to select it), by clicking the "Save Selected Shots" button (1.). Instead of saving them, the user can remove them by clicking the "Remove Selected Shots" button (2.). Both options are also available when the user wants to save or remove all shots present in the "Temporary Shots List", so that if the user wants to save or remove of the "\tmp" folder, the filter must be disabled. To save all shots in the list, the user must click on the "Save All Temp. Shots" button (3.), to remove them, click on the "Remove All Temp. Shots" button (4.).

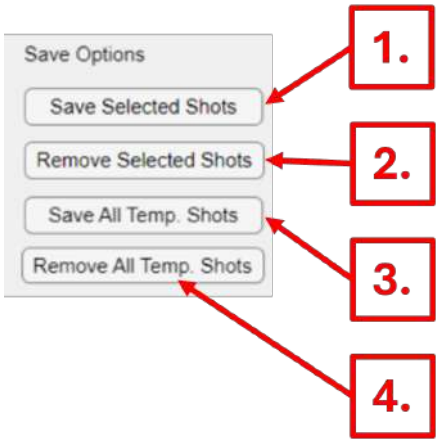

Figure A.16: "Temporary Shots List" save options.

If the user tries to save a shot that does not have any NBI signal, the application blocks this movement (as it is mandatory for further operations with that shot to have at least one NBI signal), and gives the option to delete all shots without at least one NBI signal.

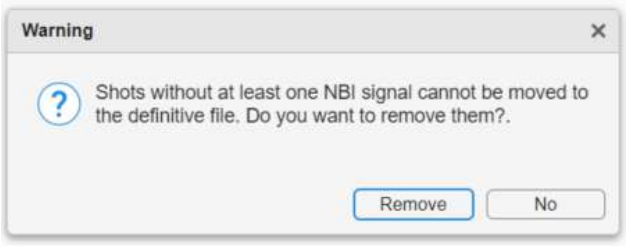

Figure A.17: NBI signals check warning.

If the user tries to save an already saved shot, the warning message in Figure A.18 appears. The user has the option to overwrite the shots or not. If the user clicks on the option "Overwrite" all files that depend on the overwritten shots will be removed, i.e. training, SVM model, and prediction files that are described in further subsections.

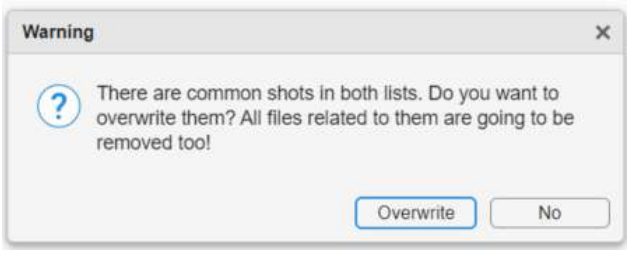

Figure A.18: Shot overwrite warning.

If the user presses the "Remove Selected Shots" button (2.) shown in Figure A.16, the warning message shown in Figure A.19 appears, so that the user has the option to remove those shots or not.

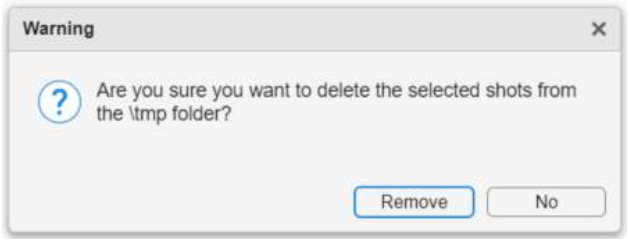

Figure A.19: Delete temporary shots warning.

If the user presses the "Remove All Temp. Shots" button (3.) shown in Figure A.16, the warning message shown in Figure A.20 appears, so that the user has the option to remove all shots from the temporary list or not.

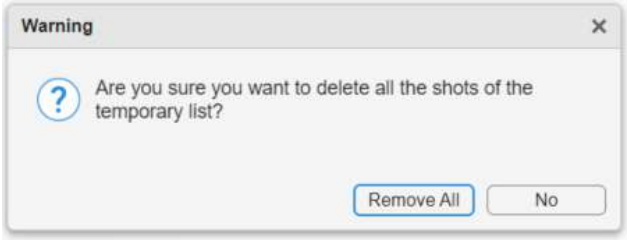

Figure A.20: Delete all temporary shots warning.

If the user tries to make an action that involves the "Temporary Shots List" and it is empty or has no selection, the warning message shown in Figure A.21 appears, indicating that there must be at least one shot selected to save it or remove it.

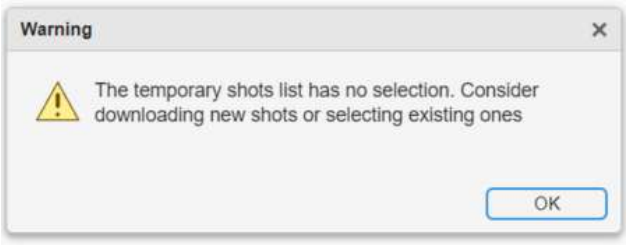

Figure A.21: Empty temporary shots list warning.

Once the user has moved one or several shots to the definitive folder, "\dat", they appear on the "Definitive Shots List" (1.), in Figure A.22. This list also has filtering options, minimum (2.), maximum (3.) and apply button (4.) as described before with the "Temporary Shots List".

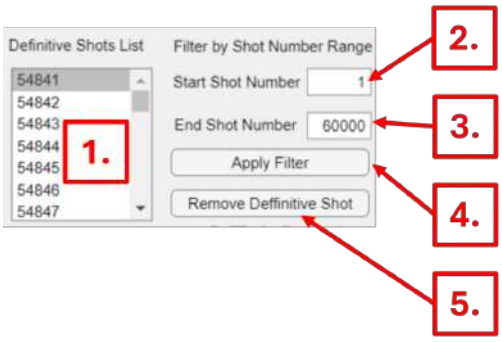

Figure A.22: "Definitive Shots List" options.

By double-clicking on a shot of the definitive list, it is displayed in the plot of the screen. The user can also delete the selected shot by clicking on the "Remove Definitive Shot" button, but all related files with that shot will be removed as well and the user will be warned as seen in Figure A.23.

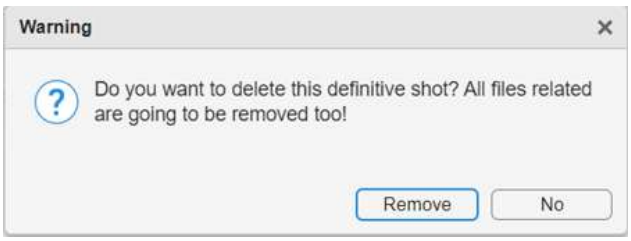

Figure A.23: Delete definitive shot warning.

Whether the user double-clicks on a shot from the "Temporary Shots List" or the "Definitive Shots List", it is displayed in the plot (1.), in Figure A.24.

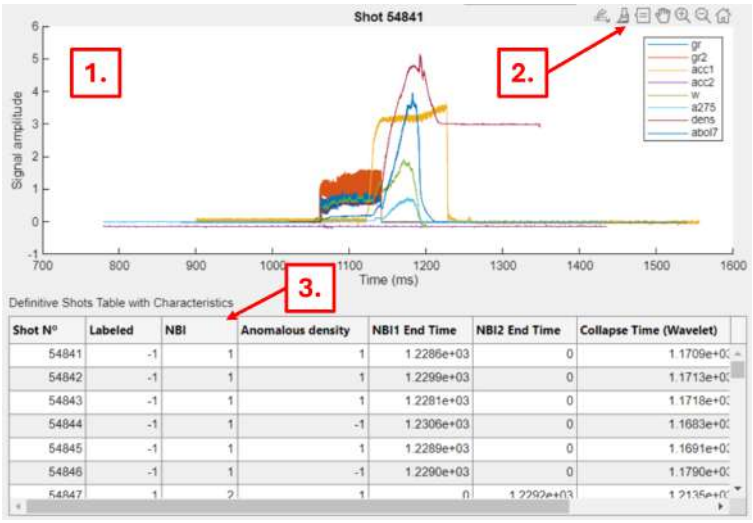

Figure A.24: First Assessment Screen plot and characteristics table.

In Figure A.24, plot (1.) has several options (2.), from icon on the left to right, the options are described in Table A.2.

Table A.2: Toolbar options available in MATLAB figure windows.

Option	Description
Save as	Opens a dialog to save the figure (e.g., `.fig`, `.png`, `.jpg`, etc.). Also allows the user to copy the plot as image or copy as vector graphic.
Brush data	Allows selecting and highlighting data points directly on the plot. The selected points can be edited, deleted, or exported to the workspace.
Data tips	Displays the coordinates of selected data points as labels. Useful for inspecting values or annotating plots.
Pan	Allows moving the figure by clicking and dragging.
Zoom out	Zooms out from the current view.
Zoom in	Zooms in on the selected region or centered area.
Restore view	Restores the original view (resets zoom and pan).

Finally, in Figure A.24, "Definitive Shots Table with Characteristics" stores different characteristics of every definitive shot stored in the database, which are described in Table A.3.

Table A.3: Definitive shots characteristics.

Characteristic	**Description**
Shot Nº	Shot number.
Labeled	Label of the shot, which could be (-1) Collapsing, (0) Doubtful, (1) Non-Collapsing or (2) Unchecked. This labels are user-defined by means of the "Analysis/Density classification (manual)" screen options, explained in Subsection A.2.4.
NBI	Indicates whether a shot has the NBI1 signal (1), the NBI2 signal (2) or both (0).
Anomalous density	Label of the density signal, which could be (-1) Anomalous, (0) Fixed, (1) Regular or (2) Unchecked. These labels are user-defined by means of the "Labels" screen options, explained in Subsection A.2.8.
NBI1 End Time	Instant in which the NBI1 signal has decayed definitely to zero.
NBI2 End Time	Instant in which the NBI2 signal has decayed definitely to zero.
Collapse Time (Wavelet)	Instant in which the energy signal starts to decay, determined thanks to the Discrete Wavelet Transform, DWT.
Collapse Time (Butterworth)	Instant in which the energy signal starts to decay, determined thanks to the Butterworth filter and an integral derivative algorithm.

A.2.3 Signal/Download Additional Signals

The "Download Additional Signals" screen, shown in Figure A.25 is displayed when the "Signal/Download Additional Signals" tab is clicked. The options available in this tab allow the user to download and save again density signals from a given shot. The user can select one or multiple shots from the "Definitive Shots List" (1.), which is the same list shown in Subsection A.2.2, and downloading again by pressing the "Download Again" button (3.). Typical filtering options (2.) are available as well. By double-clicking a shot from the list it is displayed in the plot (4.), which has the options shown in Table A.2.

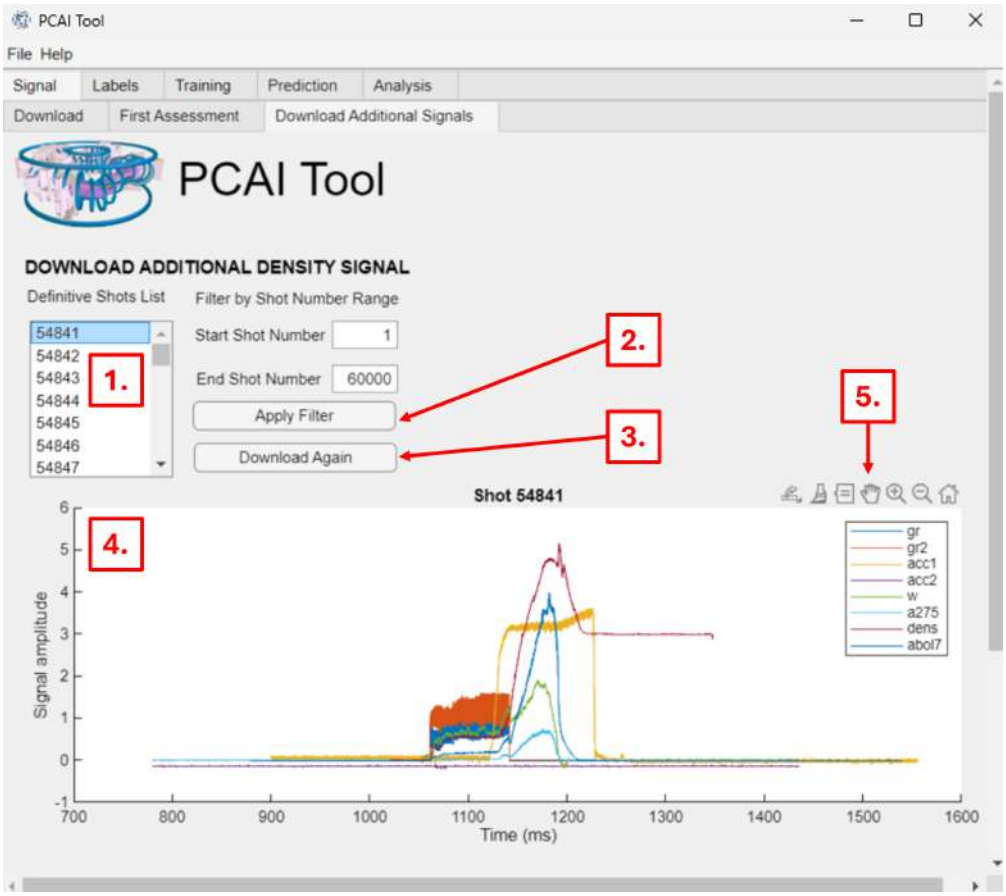

Figure A.25: Download Additional Signals screen.

If the user clicks on the "Download Again" button, the warning message shown in Figure A.26 appears, asking the user whether to overwrite the density signals from the selected shots or not. If the user presses "Overwrite", related files to the overwritten density signals' shots are removed.

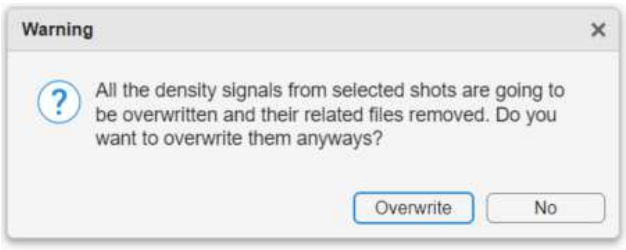

Figure A.26: Density signal overwrite warning.

A.2.4 Labels

The "Labels" screen, shown in Figure A.27 is displayed when the "Labels" tab is clicked. The options available in this tab allow the user to display a shot in the plot (4.) by selecting it in the menu "Shot Nº" (2.) and label it with security i.e., the label must be clear, thanks to the menu "Shot Type" (3.). Labels could be (-1) Collapsing, (0) Doubtful, (1) Non-Collapsing or (2) Unchecked. Plot (4.) has the options shown in Table A.2. Results of manual classification are saved in "Shots Classification Table" (5.). Finally, the user has the option to add to the "Eliminated Shots Table" (6.) shots that are not proper to operate with, to add one, press the "Add Shots to List" button (7.) and edit the cell of the "Shot Nº" column. To remove it, double-click on the "Remove" cell of the "Remove from this list" column.

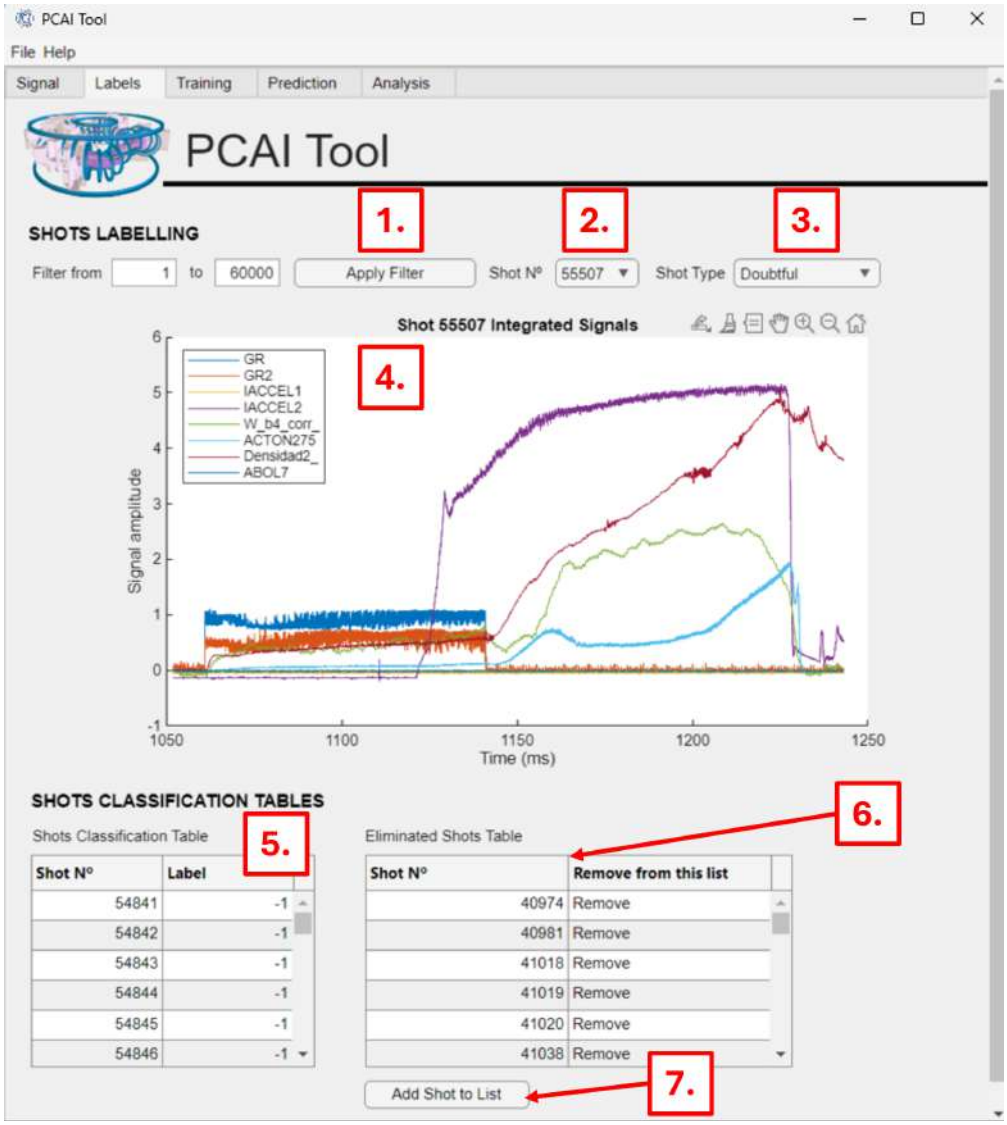

Figure A.27: Labels screen.

If the menu "Shot Nº" (2.) is empty, i.e., the "\data" folder is empty, the warning

message shown in Figure A.28 appears, so that the user is advised to save one or several in the definitive folder.

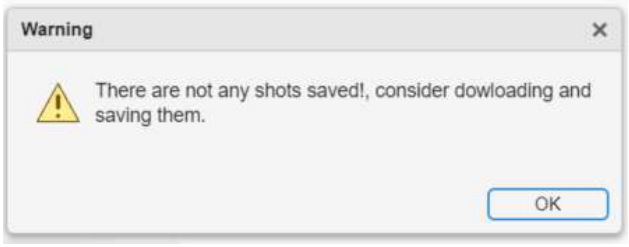

Figure A.28: Selection warning.

A.2.5 Training

The "Training" screen, shown in Figure A.29, appears when the "Training" tab is selected. This tab provides options for selecting a training dataset for the Support Vector Machine (SVM), which is used to classify shots as Collapsing (-1) or Non-Collapsing (1), and for training the model.

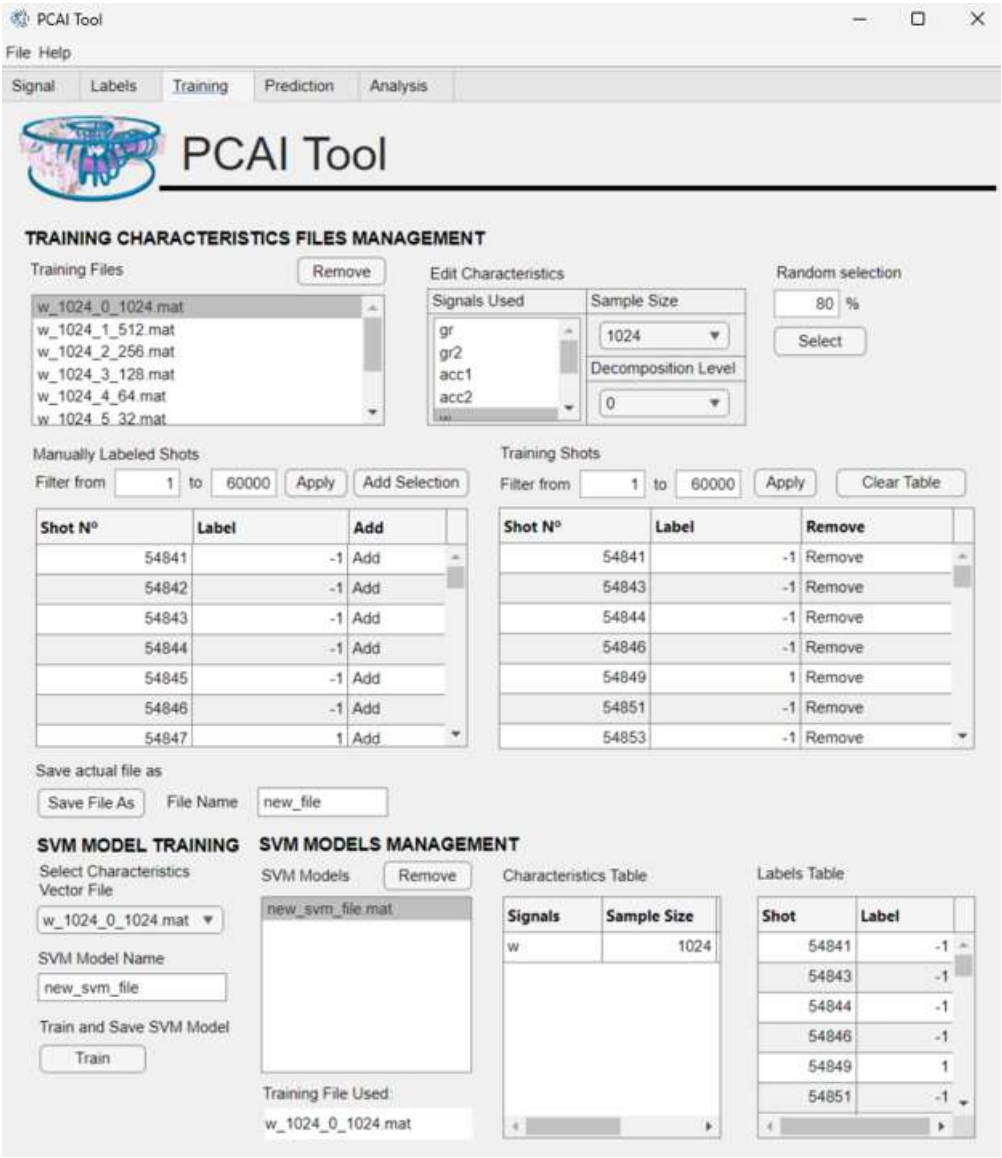

Figure A.29: Training screen.

To train an SVM, a characteristics vector with its associated labels must be created, which is saved within a training file in ".mat" format. To do so, a characteristics vector must be configured, Figure A.30, so that the user can select the signals to train the SVM (3.), the sample size of all signals used (4.) and the decomposition level for the DWT applied to the signals (5.). Previously saved training files can be seen in the list (1.). By selecting one, the selected options for the training file appears in screen, the selected file can

be removed by pressing the "Remove" button (2.). To create the training file, the user can select a percentage of the manually labeled shots to be used in the edit field (6.) and press the "Select" button (7.).

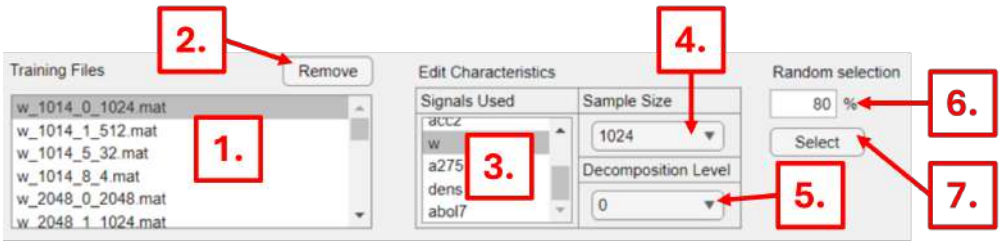

Figure A.30: Characteristics vector options.

When the "Select" button is pressed, the selected percentage of manually labeled shots appears in table (4.) in Figure A.31, those shots are the ones used to train the SVM. Available shots to train are displayed in the "Manually Labeled Shots" table (1.), which can be individually added by clicking the "Add" cell. All labeled shots can be added at once by pressing the "Add Selection" button. Typical filtering options for each table can be found in (2.) and (5.). The "Training Shots" table can be cleared by clicking the "Clear Table" button (6.). Finally, the training file can be created by pressing the button "Save File As" button (8.) with the name written in the field (7.). Created files appear in the "Training Files" list, mentioned previously.

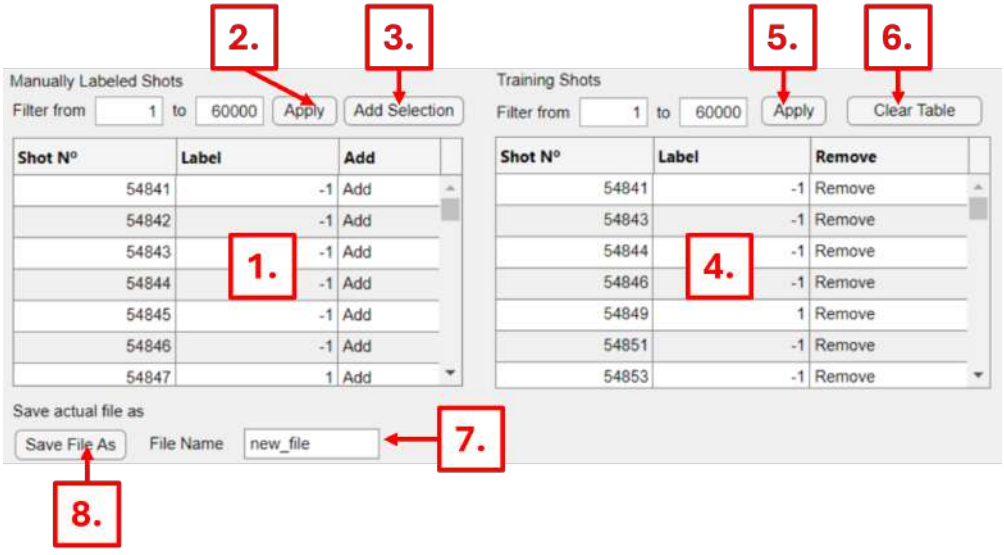

Figure A.31: Training options.

Finally, in Figure A.32 the user can select a characteristics vector file with the menu (1.), edit the name of the SVM file with the menu (2.) and train it by pressing the "Train" button (3.). Trained SVM files will appear in "SVM Models" list (4.). By selecting a model of the list, the training file will appear on screen (5.) together with the characteristics used

(7.) and its labels (8.). To remove a selected SVM model, the user can press the "Remove" button (6.).

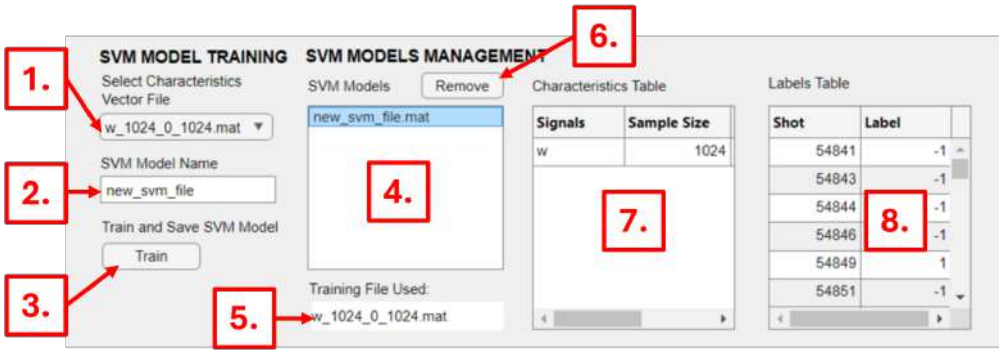

Figure A.32: SVM training options.

A.2.6 Prediction/Known Labels

The "Known Labels" screen, shown in Figure A.33 is displayed when the "Prediction/Known Labels" tab is clicked. The options available in this tab allow the user to test trained SVM models.

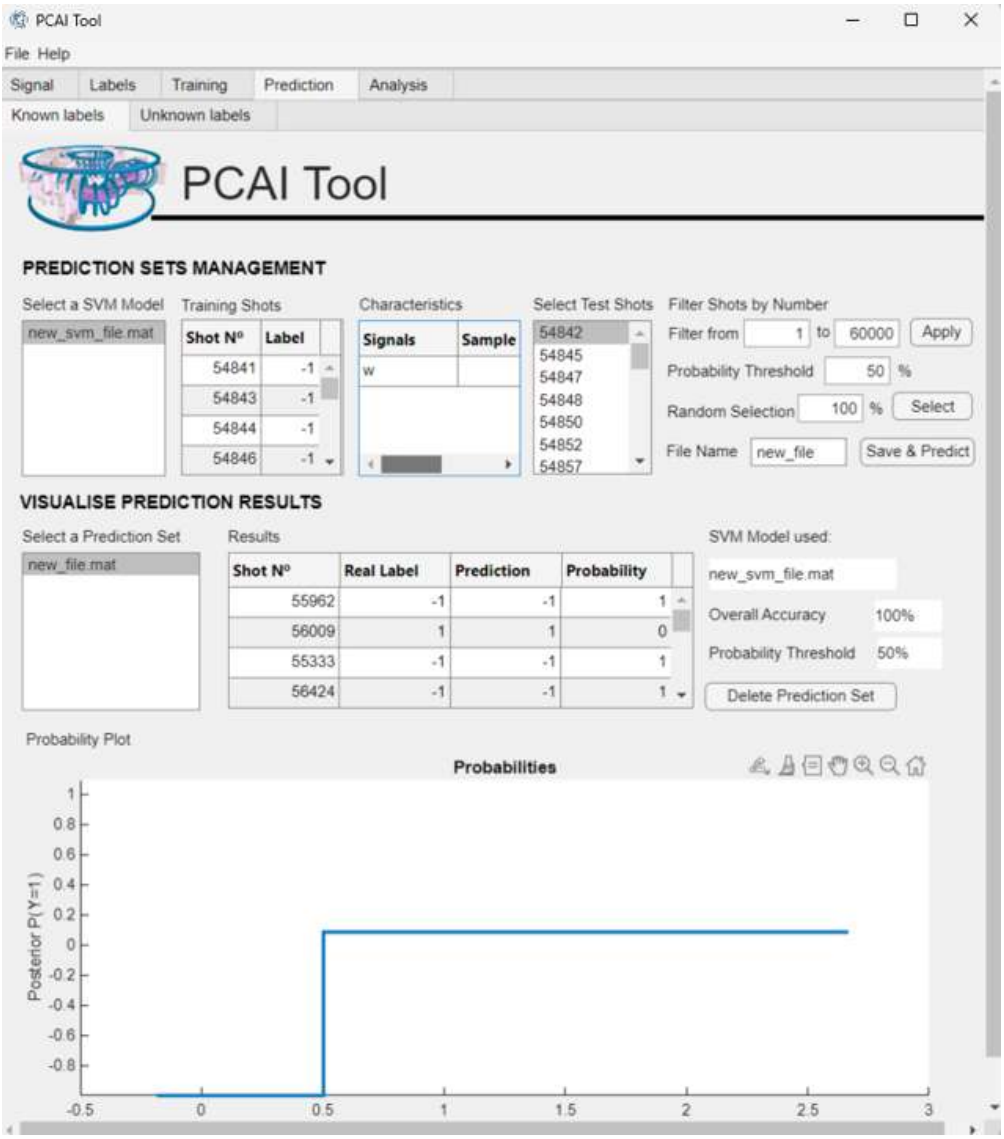

Figure A.33: Known Labels screen.

In Figure A.34, the user can select an SVM model in the list (1.), visualize the training shots used in the "Training Shots" table (2.) and check the characteristics vector employed in the "Characteristics" table (3.). To verify the accuracy of the SVM model, we have to predict the labels of already labeled shots available in the list (4.). A percentage can be selected in option (7.), and a probability threshold must be defined, so that the tool assigns a label depending on the percentage given by the SVM model prediction, i.e., the SVM returns the probability of a shot being classified as Collapsing (-1). Typical filtering options

are available (5.), and the prediction file can be named and saved by the user by editing the field "File Name" and pressing the "Save & Predict" button (8.).

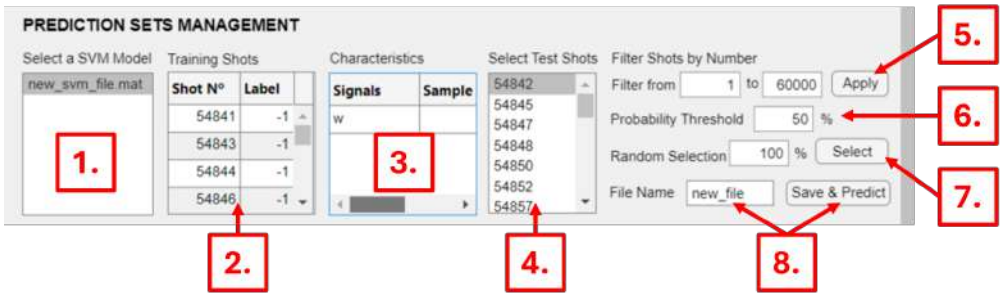

Figure A.34: Prediction sets management options.

In Figure A.35, results from model in the list (1.) can be visualized. The "Results" table (2.) shows for every shot its real label, the predicted one by the SVM model and the probability of being Collapsing (-1). The SVM used is shown in the field (3.), the configured threshold is displayed in the field (5.) and the "Overall Accuracy" (4.) is calculated taking into account the real and predicted labels. To delete a prediction set, press the "Delete Prediction Set" button (6.). The probability function, which in this case is a step one, is displayed in the plot (7.) that has the options shown in Table A.2.

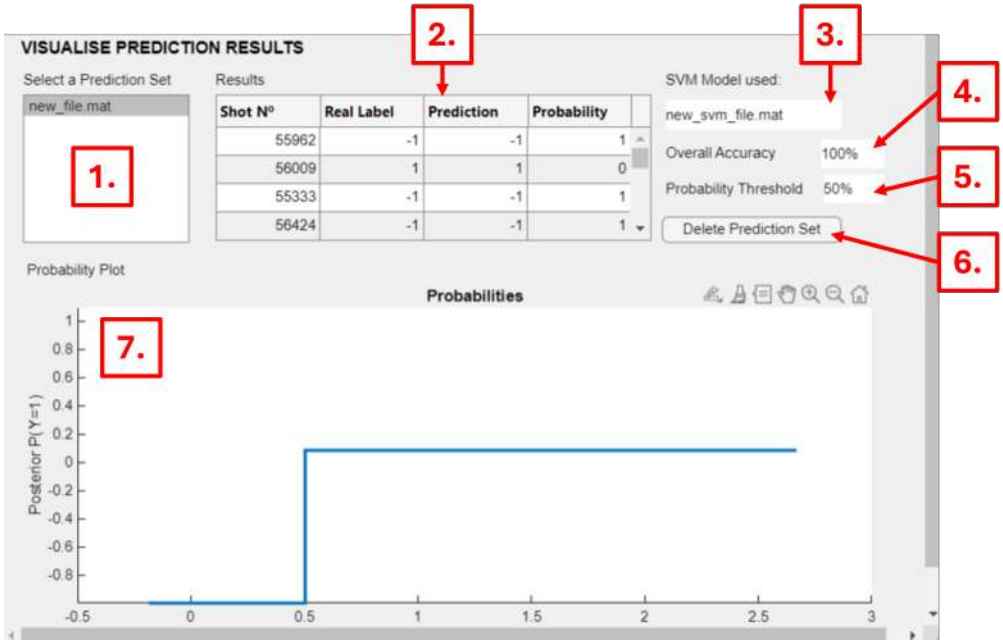

Figure A.35: Prediction results options.

A.2.7 Prediction/Unknown Labels

The "Unknown Labels" screen, shown in Figure A.36 is displayed when the "Prediction/Unknown Labels" tab is clicked. The options available in this tab allow the user to predict shots' labels with trained SVM models.

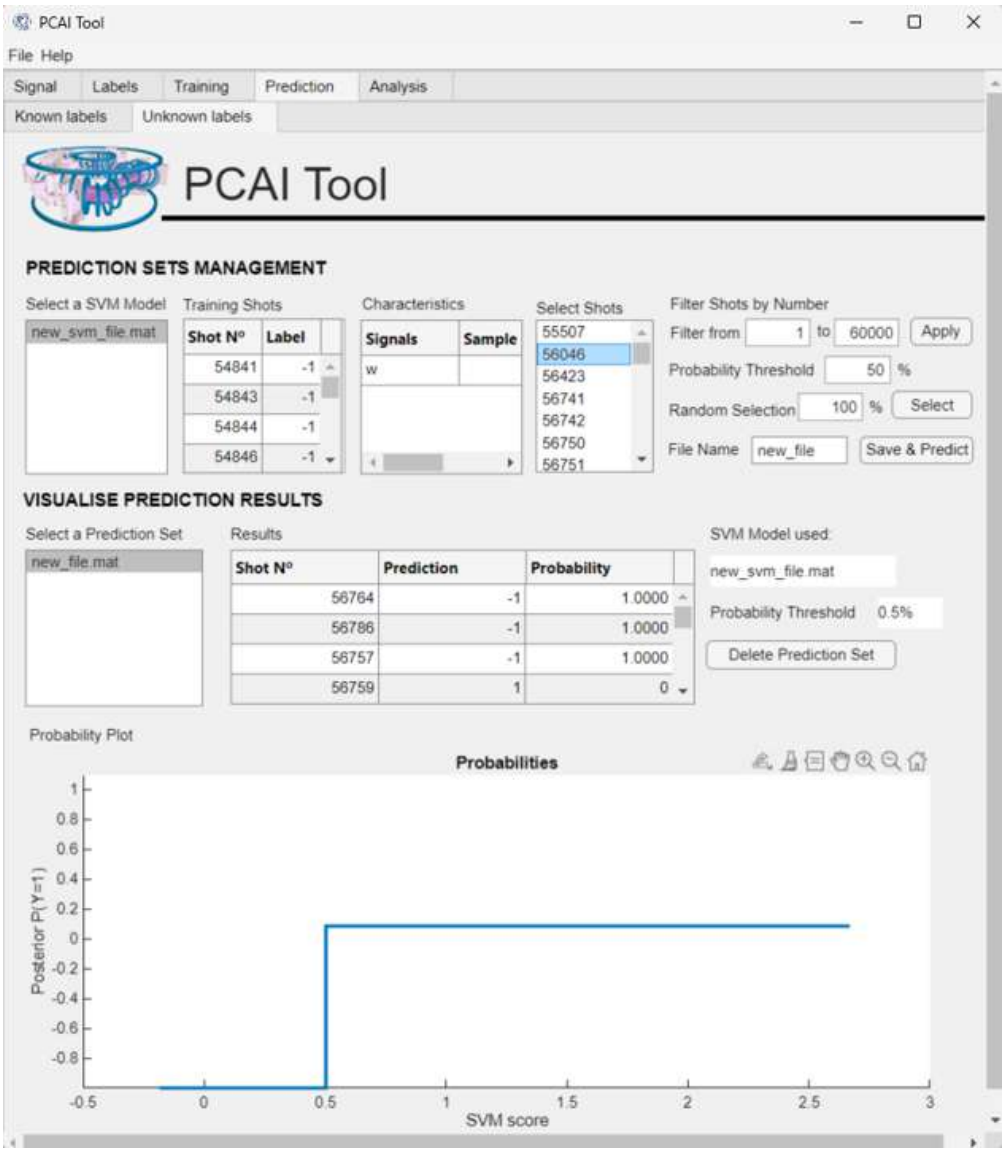

Figure A.36: Unknown Labels screen.

In Figure A.37, the user can select an SVM model in the list (1.), visualize the training shots used in the "Training Shots" table (2.) and check the characteristics vector employed in the "Characteristics" table (3.). The user can select the shots whose labels are going to be predicted in the list (4.). A percentage can be selected in option (7.), and a probability threshold must be defined, so that the tool assigns a label depending on the percentage given by the SVM model prediction, i.e., the SVM returns the probability of a shot being classified as Collapsing (-1). Typical filtering options are available (5.), and the prediction file can

be named and saved by the user by editing the field "File Name" and pressing the "Save & Predict" button (8.).

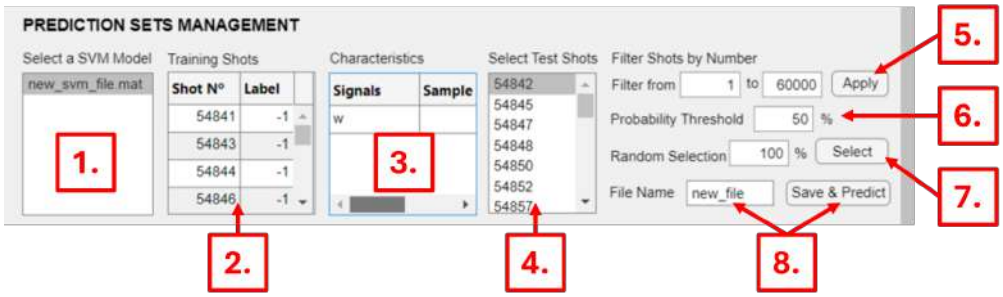

Figure A.37: Prediction sets management options.

In Figure A.38, results from model in the list (1.) can be visualized. The "Results" table (2.) shows for every shot the predicted label by the SVM model and the probability of being Collapsing (-1). The SVM used is shown in the field (3.), the configured threshold is displayed in the field (4.). To delete a prediction set, press the "Delete Prediction Set" button (5.). The probability function, which in this case is a step one, is displayed in the plot (6.) that has the options shown in Table A.2.

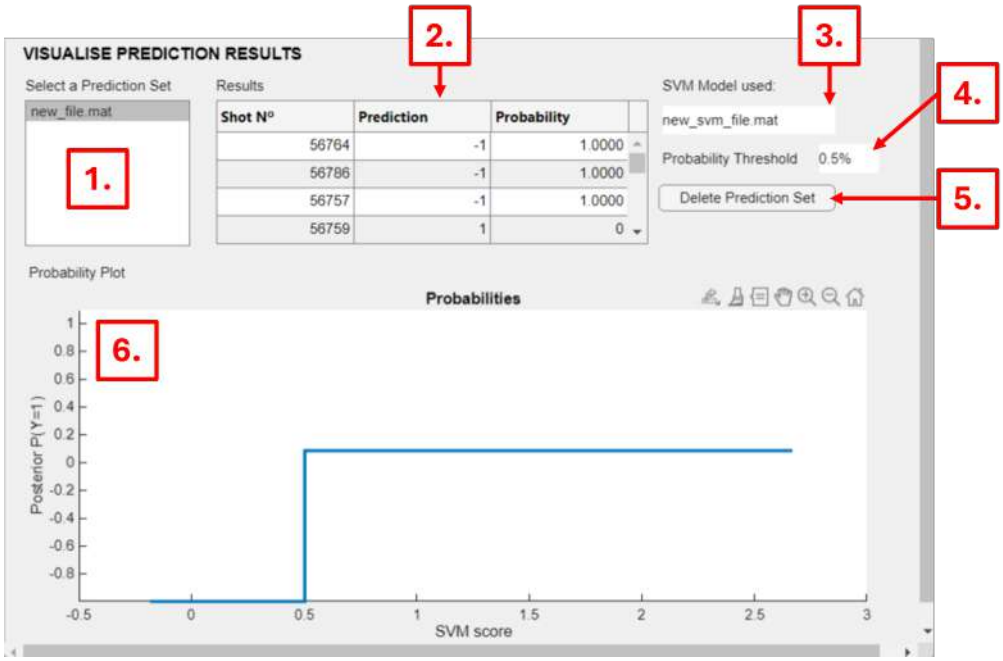

Figure A.38: Prediction results options.

A.2.8 Analysis/Density classification (manual)

The "Density classification (manual)" screen, shown in Figure A.39 is displayed when the "Analysis/Density classification (manual)" tab is clicked. The options available in this tab allow the user to display a density signal in the plot (4.) by selecting it in the menu "Shot Nº" (2.) and label it with security i.e., the label must be clear, thanks to the menu "Density Type" (3.). Labels could be (-1) Anomalous, (0) Fixed, (1) Regular or (2) Unchecked. Plot (4.) has the options shown in Table A.2. Results of manual classification are saved in "Density Classification Table" (5.). Finally, the user has the option to add to the "Eliminated Shots Table" (6.) shots that are not proper to operate with, to add one, press the "Add Shots to List" button (7.) and edit the cell of the "Shot Nº" column. To remove it, double-click on the "Remove" cell of the "Remove from this list" column.To reconstruct all the manually labeled anomalous signals, click on the "Reconstruct Anomalous" button (8.); then, those fixed signals will be tagged as (0) Fixed.

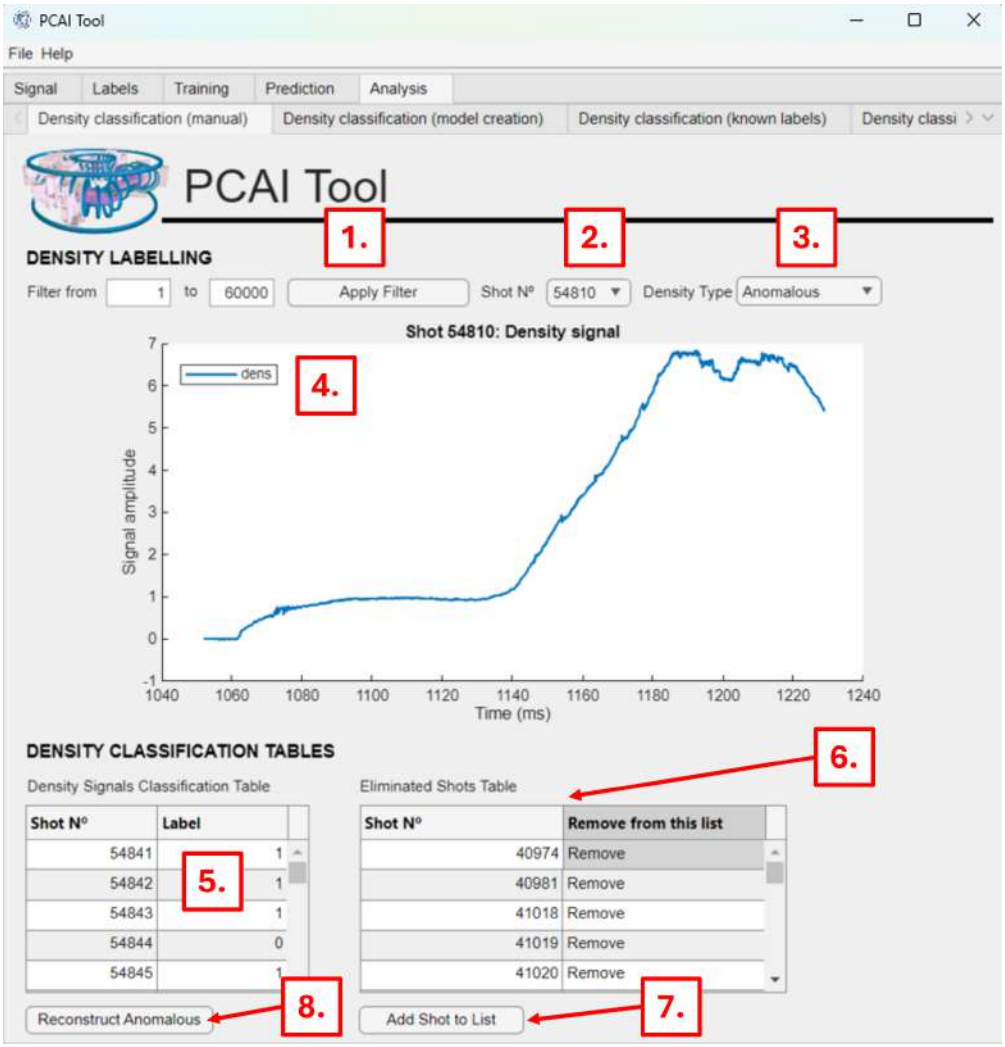

Figure A.39: Labels screen.

If the menu "Shot Nº" (2.) is empty, i.e., the "\data" folder is empty, the warning message shown in Figure A.40 appears, so that the user is advised to save one or several in the definitive folder.

Figure A.40: Selection warning.

A.2.9 Analysis/Density classification (model creation)

The "Density classification (model creation)" screen, shown in Figure A.41, appears when the "Analysis/Density classification (model creation)" tab is selected. This tab provides options for selecting a training dataset for the Support Vector Machine (SVM), which is used to classify density signals as Anomalous (-1) or Regular (1), and for training the model.

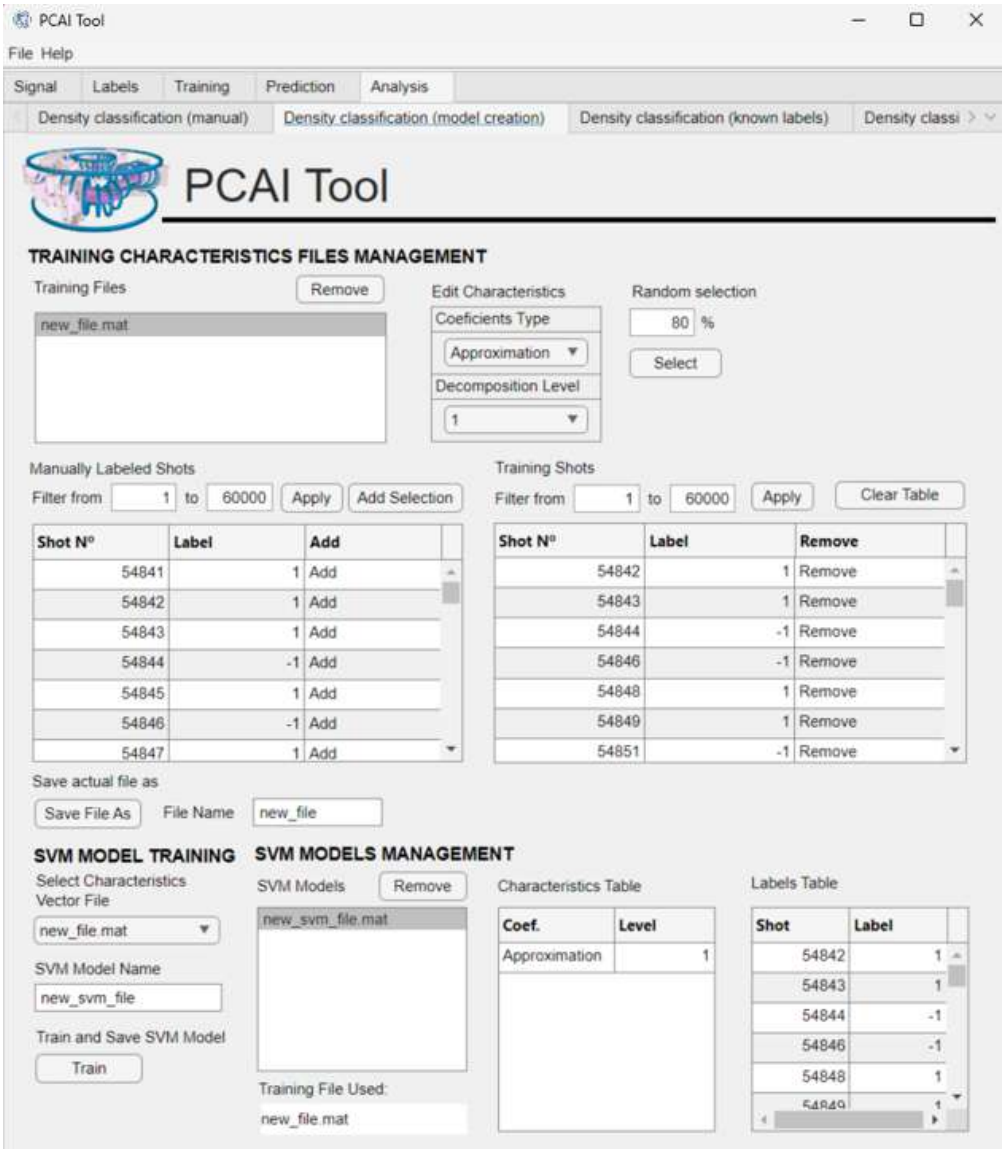

Figure A.41: Density classification (model creation) screen.

To train an SVM, a characteristics vector with its associated labels must be created, which is saved within a training file in ".mat" format. To do so, a characteristics vector must be configured, Figure A.42, so that the user can select what kinds of coefficients, approximation or detail, are used in the applied DTW to train the SVM (3.), and the decomposition level for the DWT applied to the signals (4.). Previously saved training files can be seen in the list (1.). By selecting one, the selected options for the training file appears in screen, the

selected file can be removed by pressing the "Remove" button (2.). To create the training file, the user can select a percentage of the manually labeled shots to be used in the edit field (5.) and press the "Select" button (6.).

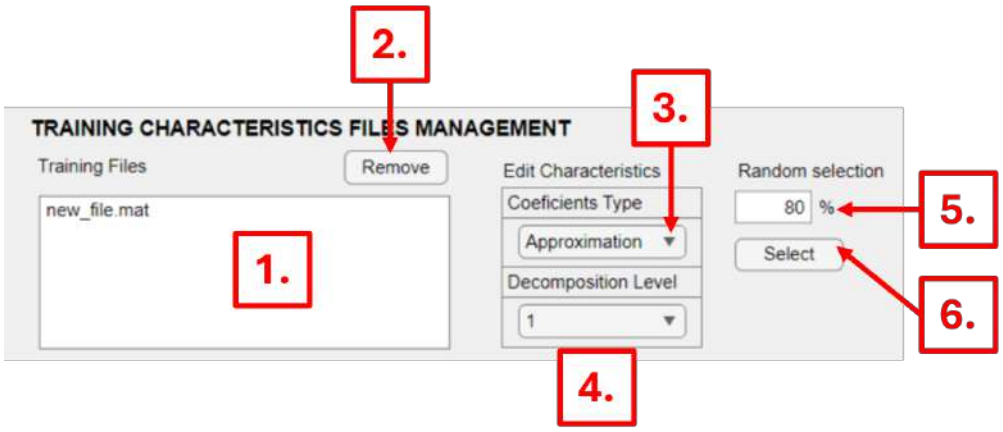

Figure A.42: Characteristics vector options.

When the "Select" button is pressed, the selected percentage of manually labeled shots' density signals appears in table (4.) in Figure A.43, those shots are the ones used to train the SVM. Available shots' density signals to train are displayed in the "Manually Labeled Shots" table (1.), which can be individually added by clicking the "Add" cell. All labeled shots' density signals can be added at once by pressing the "Add Selection" button. Typical filtering options for each table can be found in (2.) and (5.). The "Training Shots" table can be cleared by clicking the "Clear Table" button (6.). Finally, the training file can be created by pressing the button "Save File As" button (8.) with the name written in the field (7.). Created files appear in the "Training Files" list, mentioned previously.

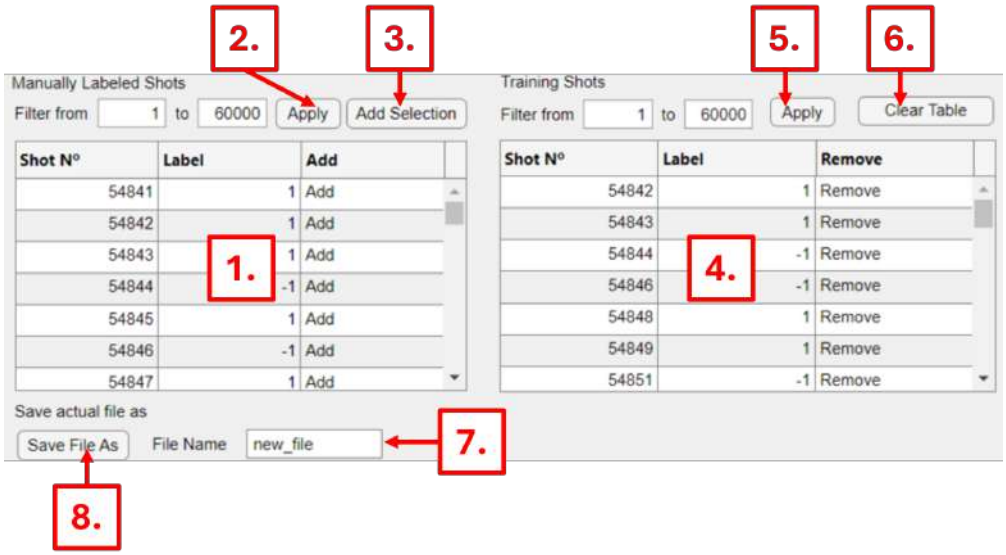

Figure A.43: Training options.

Finally, in Figure A.44 the user can select a characteristics vector file with the menu (1.), edit the name of the SVM file with the menu (2.) and train it by pressing the "Train" button (3.). Trained SVM files will appear in "SVM Models" list (4.). By selecting a model of the list, the training file will appear on screen (5.) together with the characteristics used (7.) and its labels (8.). To remove a selected SVM model, the user can press the "Remove" button (6.).

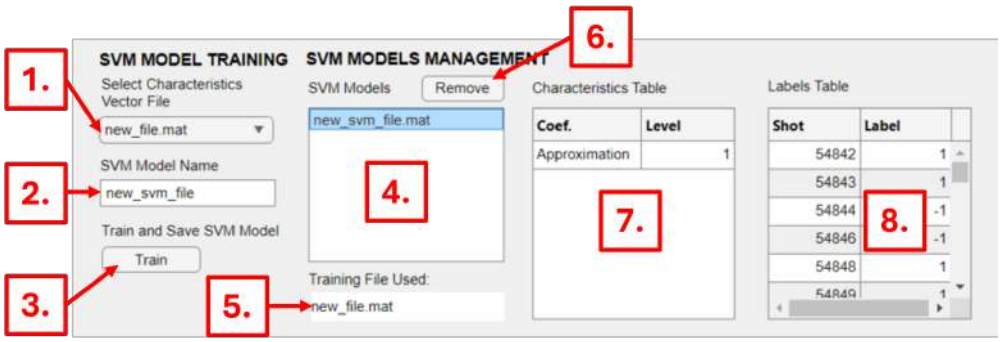

Figure A.44: SVM training options.

A.2.10 Analysis/Classification (known labels)

The "Classification (known labels)" screen, shown in Figure A.45 is displayed when the "Analysis/Classification (known labels)" tab is clicked. The options available in this tab allow the user to test trained SVM models.

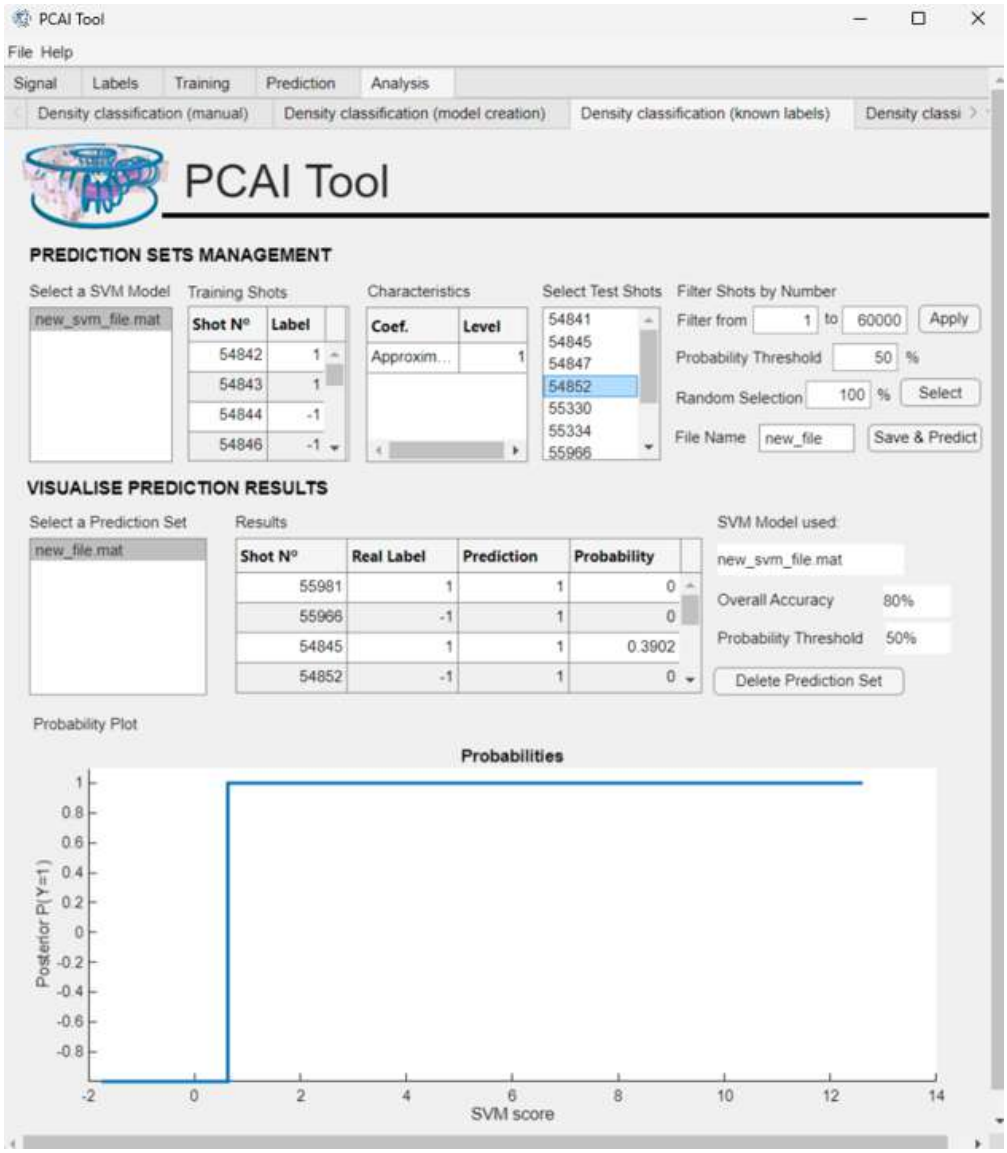

Figure A.45: Classification (known labels).

In Figure A.46, the user can select an SVM model in the list (1.), visualize the training shots used in the "Training Shots" table (2.) and check the characteristics vector employed in the "Characteristics" table (3.). To verify the accuracy of the SVM model, we have to predict the labels of already labeled density signals available in the list (4.). A percentage can be selected in option (7.), and a probability threshold must be defined, so that the tool assigns a label depending on the percentage given by the SVM model prediction, i.e., the SVM returns the probability of a density signal being classified as Anomalous (-1). Typical

filtering options are available (5.), and the prediction file can be named and saved by the user by editing the field "File Name" and pressing the "Save & Predict" button (8.).

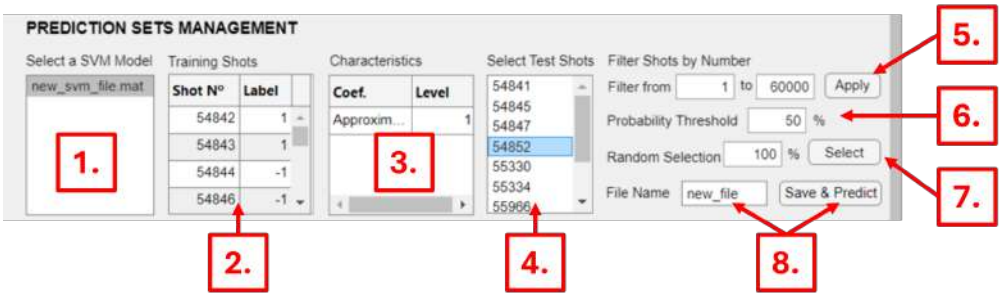

Figure A.46: Prediction sets management options.

In Figure A.47, results from model in the list (1.) can be visualized. The "Results" table (2.) shows for every shot its real label, the predicted one by the SVM model and the probability of being Anomalous (-1). The SVM used is shown in the field (3.), the configured threshold is displayed in the field (5.) and the "Overall Accuracy" (4.) is calculated taking into account the real and predicted labels. To delete a prediction set, press the "Delete Prediction Set" button (6.). The probability function, which in this case is a step one, is displayed in the plot (7.) that has the options shown in Table A.2.

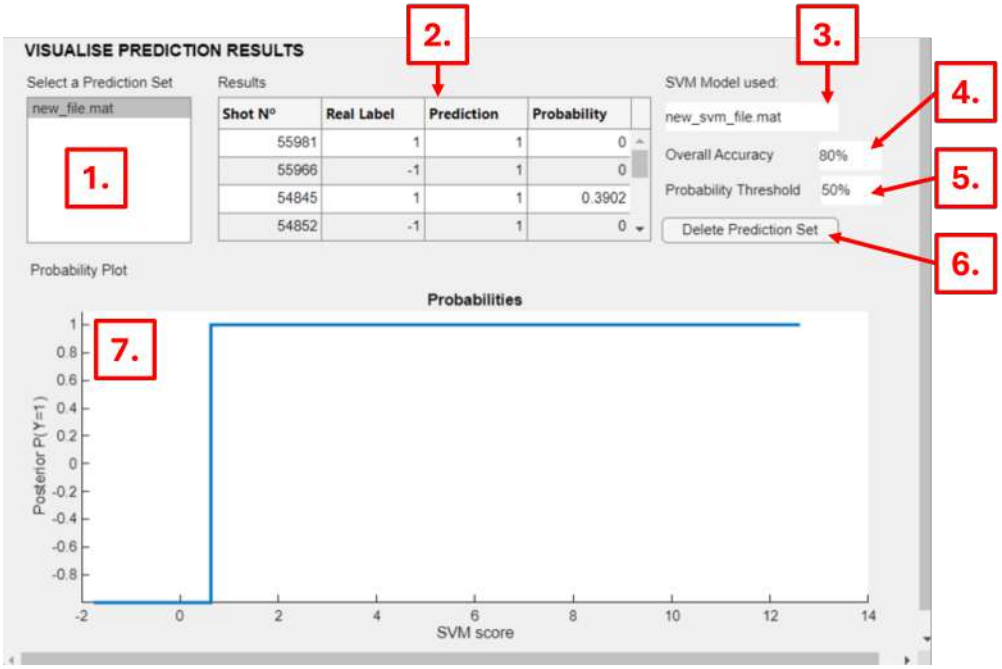

Figure A.47: Prediction results options.

A.2.11 Analysis/Classification (unknown labels)

The "Classification (unknown labels)" screen, shown in Figure A.48 is displayed when the "Analysis/Classification (unknown labels)" tab is clicked. The options available in this tab allow the user to predict density signals' labels with trained SVM models.

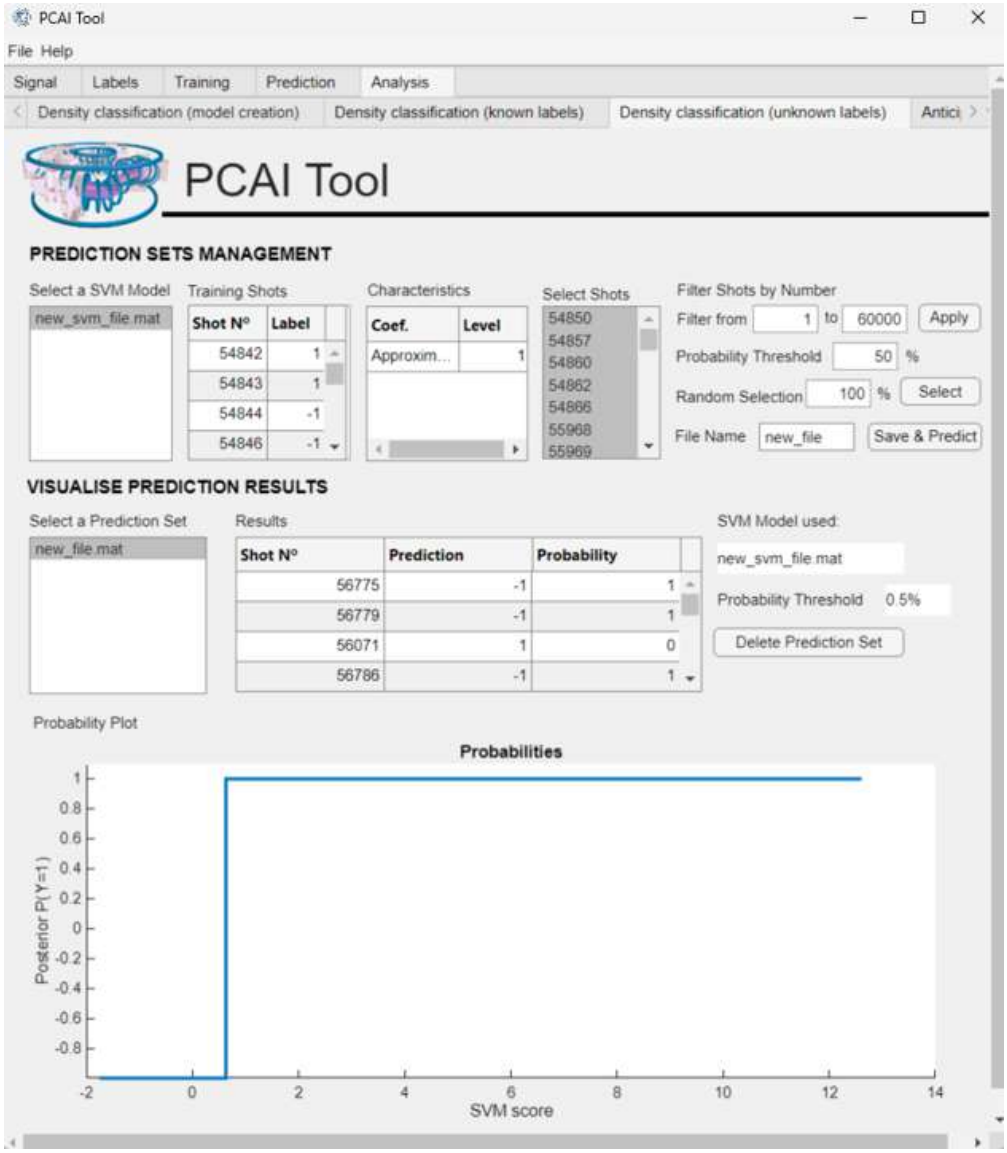

Figure A.48: Classification (unknown labels) screen.

In Figure A.49, the user can select an SVM model in the list (1.), visualize the training shots used in the "Training Shots" table (2.) and check the characteristics vector employed in the "Characteristics" table (3.). The user can select the shots whose labels are going to be predicted in the list (4.). A percentage can be selected in option (7.), and a probability threshold must be defined, so that the tool assigns a label depending on the percentage given by the SVM model prediction, i.e., the SVM returns the probability of a density signal being classified as Anomalous (-1). Typical filtering options are available (5.), and the prediction

file can be named and saved by the user by editing the field "File Name" and pressing the "Save & Predict" button (8.).

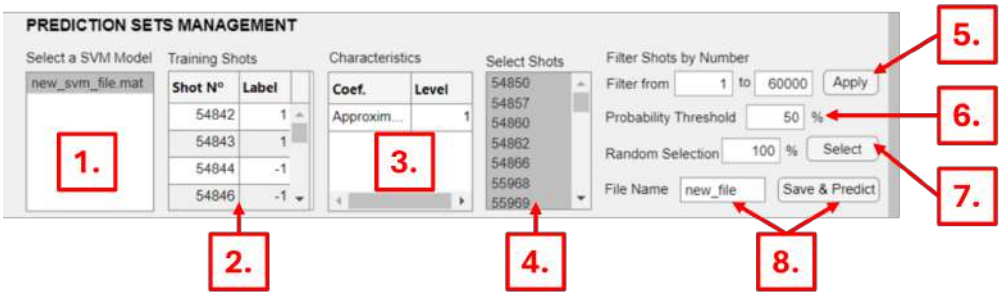

Figure A.49: Prediction sets management options.

In Figure A.50, results from model in the list (1.) can be visualized. The "Results" table (2.) shows for every shot the predicted label by the SVM model and the probability of being Anomalous (-1). The SVM used is shown in the field (3.), the configured threshold is displayed in the field (4.). To delete a prediction set, press the "Delete Prediction Set" button (5.). The probability function, which in this case is a step one, is displayed in the plot (6.) that has the options shown in Table A.2.

Figure A.50: Prediction results options.

A.2.12 Analysis/Anticipation time

The "Anticipation time" screen, shown in Figure A.51 is displayed when the "Analysis/Anticipation time" tab is clicked. The options available in this tab allow the user to create a histogram that reflects the time differences between the start of the decay of the energy signal and the shot end time. To do so, one or several prediction models can be added by double-clicking the "Add" cell of the "Prediction Models" table (1.) to the "Model to be used" table (2.). To remove a model from the table, double-click on the "Remove" cell. All predicted collapsing shots from those models will appear in the "Collapsing Shots" table (3.), together with the labeled ones by the user. Typical filtering options (4.) can be applied to that table. Finally, to create a histogram, the user must define a bin width (5.) and press the "Create Histogram" button (6.). The histogram will appear in the plot (7.), which has the options shown in Table A.2.

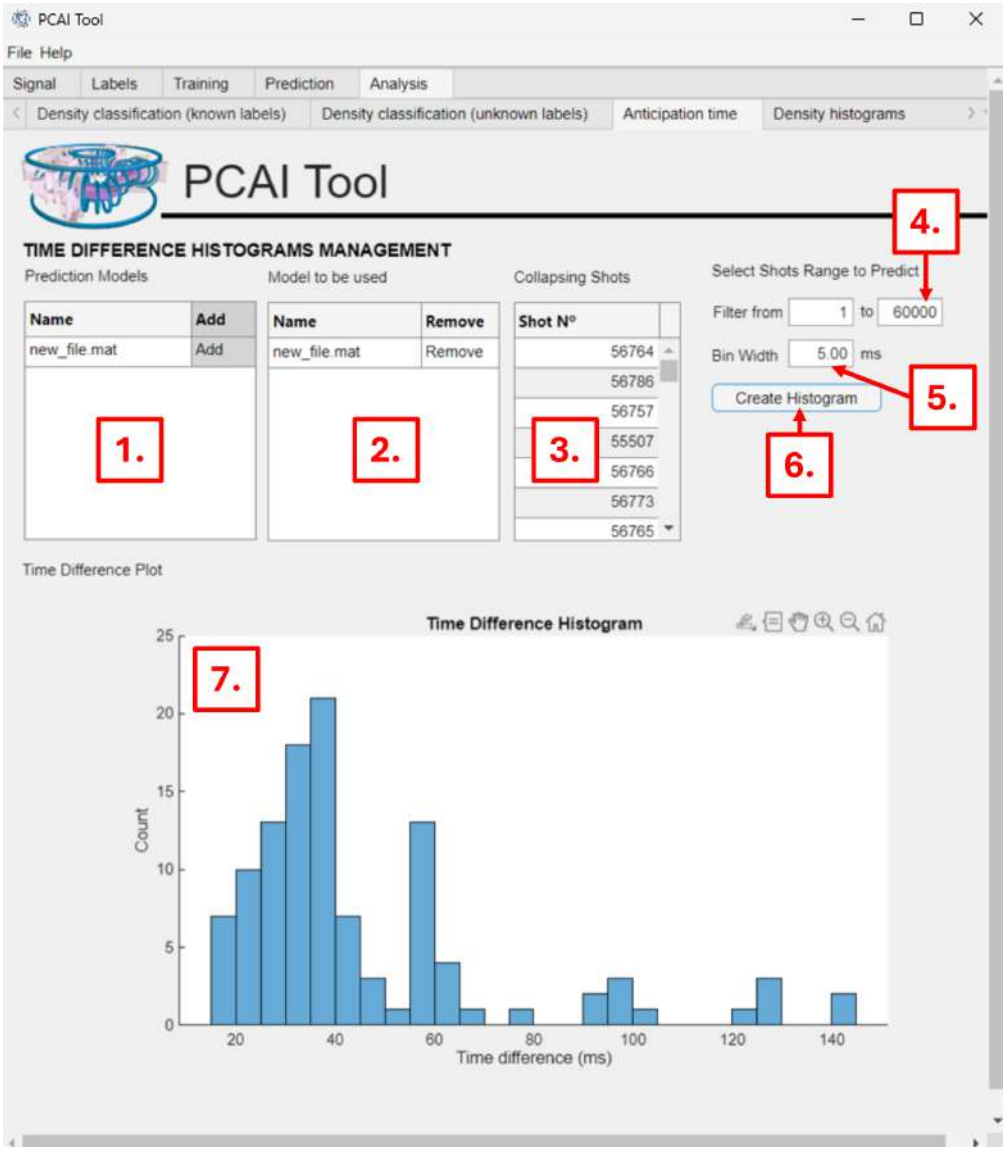

Figure A.51: Anticipation Time Screen.

A.2.13 Analysis/Density histograms

The "Density histograms" screen, shown in Figure A.52 is displayed when the "Analysis/-Density histograms" tab is clicked. The options available in this tab allow the user to create a histogram that reflects the values of the density signals at the start time of the decay of the energy signal. To do so, one or several prediction models can be added by double-clicking the "Add" cell of the "Prediction Models" table (1.) to the "Model to be used" table (2.). To remove a model from the table, double-click on the "Remove" cell. All shots from those models will appear in the "Collapsing Shots" table (3.), together with the labeled ones by the user. Typical filtering options (4.) can be applied to that table. Finally, to create a histogram, the user must define a bin width (5.) and press the "Create Histogram" button (6.). The histogram for the collapsing discharges will appear in the plot (7.) and for the non-collapsing discharges in the plot (8.), which has the options shown in Table A.2.

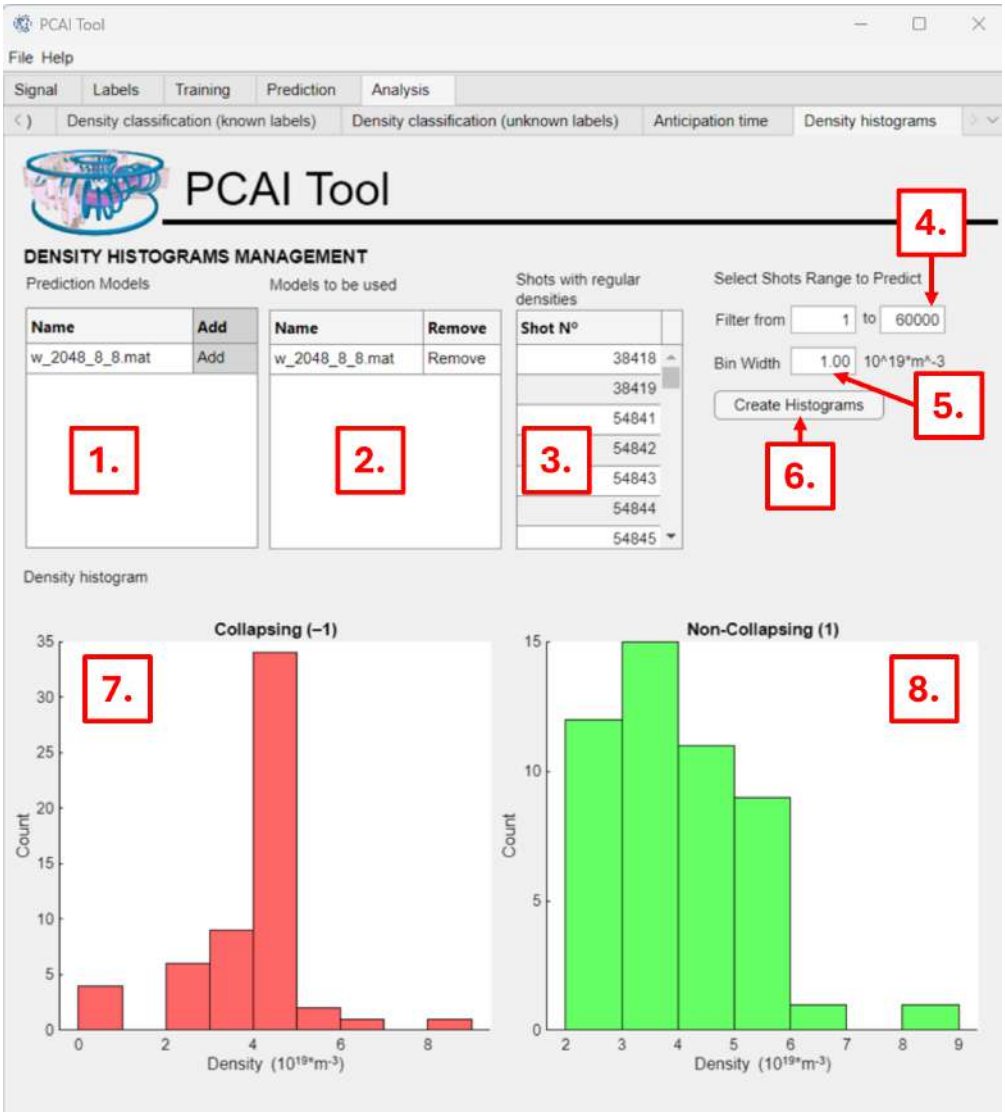

Figure A.52: Density histograms screen.