

# POLITECNICO DI TORINO

Master's Degree in Ingegneria Energetica e Nucleare



**Politecnico  
di Torino**

Master's Degree Thesis

## Simulation-based exploration of the model of a Molten Salt Fast Reactor for the identification and classification of abnormal operating conditions

Supervisors

Prof. Nicola PEDRONI

Prof. Sandra DULLA

Prof. Stefano LORENZI

Dott. Nicolò ABRATE

Candidate

Nicolò CARUSO

March 2023

## Abstract

The deployment of the Gen-IV Molten Salt Fast Reactor (MSFR) requires the demonstration of its enhanced safety features with respect to other traditional reactor concepts. To this aim, a power plant simulator including the primary system, the secondary system and the balance-of-plant has been developed in the framework of the SAMOFAR EU project. This simulator allows to describe the plant (transient) response to a variety of normal and abnormal operating conditions.

Within this framework, the objectives of the present thesis are: 1) to propose a simulation-based method to characterize the system behaviour with respect to variations in physical and operational parameters, by means of a thorough exploration of the MSFR power plant state space; 2) to develop a data-driven algorithm for the efficient detection and classification of incidents, relying on a k-Nearest Neighbors (kNN) classifier.

The proposed approach comprises the following steps. First, a set of physical (input) parameters that are found to strongly influence the behaviour of the plant simulator (e.g., the fuel and intermediate salt mass flow rates and the gas flow rate) is selected, together with their ranges of variation. Second, several possible combinations of physical parameters values are generated by random sampling and the corresponding time-varying (transient) behaviour of the MSFR is simulated. Finally, the time evolution of some relevant (output) plant parameters (e.g., the fuel and intermediate molten salt temperatures) is analysed in detail to: (i) identify normal and abnormal system (output) configurations; (ii) train, validate and test the kNN incident detection and classification model; (iii) retrieve those combinations of the reactor physical (input) variables (e.g., circulation pumps failures) that are responsible for the abnormal system states (namely, fault diagnosis). The proposed method has shown a satisfactory performance: in particular, the incident detection and classification accuracy ranges between 89% and 99%.





# Acknowledgements

Vorrei esprimere i miei più sentiti ringraziamenti al Prof. Nicola Pedroni per avermi seguito, guidato e supportato durante questo percorso di tesi. Vorrei ringraziare la Prof.ssa Sandra Dulla per avermi seguito durante il percorso e per tutti i consigli e il supporto fornitomi alla conferenza YMSR di Lecco. Un sentito ringraziamento al Prof. Stefano Lorenzi per il prezioso contributo sull'utilizzo del simulatore. Un caloroso ringraziamento al Dott. Nicolò Abrate per il suo aiuto, i consigli e la sua disponibilità sempre presente. Un sentito ringraziamento al gruppo NEMO per aver messo a disposizione le loro risorse computazionali e ai Proff. Antonio Froio e Roberto Bonifetto per avermi aiutato con l'utilizzo di Dymola.



# Table of Contents

<b>List of Tables</b>	VII
<b>List of Figures</b>	VIII
<b>1 Introduction</b>	1
1.1 Background . . . . .	1
1.2 Molten Salt Reactors . . . . .	2
1.3 The Molten Salt Fast Reactor . . . . .	3
1.4 The MSFR simulation model . . . . .	4
1.5 Thesis objectives and outline . . . . .	6
<b>2 Preliminary analysis of the MSFR model</b>	8
2.1 Initial simulations . . . . .	8
2.2 Selection of control and controlled variables . . . . .	14
2.3 Definition of the simulation status . . . . .	18
2.4 Range definition of control variables . . . . .	20
2.5 Simulation settings . . . . .	20
2.6 Categorization rules . . . . .	23
<b>3 Methodology</b>	24
3.1 Building a $k$ NN classifier . . . . .	27
3.1.1 Feature extraction for dimensionality reduction . . . . .	29
3.1.2 Normalization . . . . .	30
3.1.3 Hyperparameters optimization and cross-validation . . . . .	31
3.1.4 Testing and evaluation . . . . .	33
3.2 Inference on inputs . . . . .	34
<b>4 Results on coarse sampled data set</b>	35
4.1 Training and test set ground truth maps . . . . .	35
4.2 Classifiers construction . . . . .	39
4.3 Prediction Maps . . . . .	41

<b>5</b>	<b>Results on fine sampled data set</b>	48
5.1	Training and test set ground truth maps . . . . .	48
5.2	Test set transients . . . . .	53
5.3	Classifiers construction . . . . .	56
5.4	Prediction and correct or misclassified maps . . . . .	58
5.5	Confusion matrix charts . . . . .	65
5.6	Inference on inputs (diagnosis on pumps failure intensities) . . . . .	69
<b>6</b>	<b>Conclusions</b>	73
6.1	Future developments . . . . .	74
	<b>Bibliography</b>	76

# List of Tables

3.1	Type of Data Sets used during the analysis . . . . .	24
4.1	Feature reduction using DS1 . . . . .	39
4.2	Hyperparameters selections and accuracy on test of different classifiers using DS1A . . . . .	40
4.3	Hyperparameters selections and accuracy on test of different classifiers using DS1B . . . . .	40
5.1	Feature reduction using DS2 . . . . .	56
5.2	Hyperparameters selections and accuracy on test of different classifiers using DS2A . . . . .	57
5.3	Hyperparameters selections and accuracy on test of different classifiers using DS2B . . . . .	57

# List of Figures

1.1	Schematic views of the MSFR [9] . . . . .	3
1.2	Schematic view of the BoP [9] . . . . .	4
1.3	Object-oriented Modelica models and conceptual scheme [10] . . . . .	5
2.1	Input signals for the transient simulation caused by a 20% linear reduction of the FC flow rate . . . . .	10
2.2	Output from the transient simulation caused by a 20% linear reduction of the FC flow rate . . . . .	11
2.3	Input signals for the transient simulation caused by a 90% exponential reduction of the GC and RH flow rates . . . . .	12
2.4	Output from the transient simulation caused by a 90% exponential reduction of the GC and RH flow rates: a) Thermal power b) Mechanical power c) FC temperatures d) IC temperatures . . . . .	13
2.5	Behaviour of controlled variables due to different linear FC flow rate reductions . . . . .	15
2.6	Behaviour of controlled variables due to different exponential GC flow rate reductions . . . . .	16
2.7	a) Numerical Failure b) $LT_{IC}$ failure . . . . .	19
2.8	Signal extension until the end simulation time . . . . .	19
2.9	Time evolution of the controlled variables using different time steps.	21
2.10	Quality of the results for different time steps . . . . .	21
2.11	File size and simulation time reductions relative to benchmark simulation with finest time step. . . . .	22
3.1	Flow chart representation of the analysis steps . . . . .	25
3.2	SVD quality . . . . .	30
3.3	Hyperparameters optimization in cross-validation routine . . . . .	32
4.1	Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data which contains NF simulations (DS1A). . . . .	37

4.2	Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data which is NF free (DS1B). . . . .	38
4.3	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is not normalized. . . . .	42
4.4	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is normalized once before SVD. . . . .	43
4.5	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is normalized twice before and after SVD. . . . .	44
4.6	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is not normalized. . . . .	45
4.7	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is normalized once before SVD. . . . .	46
4.8	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is normalized twice before and after SVD. . . . .	47
5.1	Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data, which contains NF simulations (DS2A). . . . .	50
5.2	Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data, which is NF free (DS2B). . . . .	51
5.3	Scatter Plot Matrix of the inputs of the training set according to the different classification rules using data, which is NF-free (DS2B). . . . .	52
5.4	Transient evolution of the controlled variables, using C1 rules and data with NF presence. . . . .	54
5.5	. . . . .	55

5.6	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is not normalized. . . . .	59
5.7	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is normalized once before SVD. . . . .	60
5.8	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is normalized twice before and after SVD. . . . .	61
5.9	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is not normalized. . . . .	62
5.10	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is normalized once before SVD. . . . .	63
5.11	Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is normalized twice before and after SVD. . . . .	64
5.12	Confusion matrices using DS2A data (left figures) or DS2B data (right figures) according to the different classification rules. Feature Matrix is not normalized. . . . .	66
5.13	Confusion matrices using DS2A data (left figures) or DS2B data (right figures) according to the different classification rules. Feature Matrix is normalized once before SVD. . . . .	67
5.14	Confusion matrices using DS2A data (left most figures) or DS2B data (right figures) according to the different classification rules. Feature Matrix is normalized before and after SVD. . . . .	68
5.15	Number of observations (left figures) and conditional probability density functions (right figures) for different failure intensities typologies. Using DS2B and categorization rules of C3. . . . .	71
5.16	Probability density functions characterizing four different classified test points. . . . .	72

# Chapter 1

## Introduction

### 1.1 Background

The process of constructing the infrastructures needed in a modern and rapidly developing economy till now has been very energy and emissions intensive [1]. Moreover it is foreseen that the majority of the two billion people expected to be added to the world's population by 2050 are going to be people of rapidly developing economy [1]. In the framework of the Paris Agreement for fighting against climate change, green-house gases emissions in the atmosphere need to be consistently reduced in order to not overcome the 2°C increase. Missing the objective will lead to well known catastrophic scenarios for the environment and life in general [2].

In this framework, low carbon energy sources play a key role. In particular nuclear energy can provide clean, very reliable electricity and high-thermal energy, at competing cost [3]. In response to the public and political debate on nuclear energy about wastes, costs, safety and proliferation, the Generation IV International Forum (GIF) was formed, in 2001, with the aim of addressing these issues. The GIF, during its activities has selected six promising reactor technologies for further research and development [4]. One of these reactor technology is the one of the the Molten Salt Reactor (MSR). In this thesis a safety study of the Molten Salt Fast Reactor (MSFR) is carried out.

The remaining part of this chapter is structured as follows. In section 1.2 a brief history of MSR, their challenges and advantages are presented; section 1.3 gives details about the specific configuration of the MSFRs, which is the subject of this work; section 1.4 introduces the computational model of the MSFR used in this work while section 1.5 depicts the thesis objectives and outline.

## 1.2 Molten Salt Reactors

The history of MSR starts in the US in 1954, with the Aircraft Reactor Experiment (ARE) [5]. This research project then shifted to the Molten Salt Reactor Experiment (MSRE) 1964-1969 [6]. In the years 1970-1976 the design of a Molten Salt Fast Breeder Reactor (MSFBR) was carried out but never built [7]. In the '90s the US studies on MSR were recovered by the French Alternative Energies and Atomic Energy Commission (CEA), addressing the main issues of the MSFBR: null to positive feedback coefficients, positive void coefficient, unrealistic reprocessing and problems related to the graphite moderator. After that, the selection of the MSR as one of the six designs of generation IV new reactors by the GIF, relaunched the global research on this technology. The MSR technology offers many design possibilities like working in thermal or fast spectrum, with solid or liquid fuel and adopting different fuel cycles. In Europe, starting from the '90s french work, the idea of a liquid fuel MSR operated in the fast spectrum named Molten Salt Fast Reactor (MSFR) arose.

The choice of a MSR design gives many benefits but provides also some challenges [8]. Some of the benefits are:

- the possibility to operate near atmospheric pressure;
- the radioisotope retention of the molten salt compared to the high mobility for the most volatile fission products in solid fuels;
- being chemically inert and not flammable; the absence of fuel failure due to irradiation;
- excellent neutron economy and flexible fuel cycle;
- the possibility to operate at high temperatures, reaching high thermal efficiencies;
- effective load following for a correct integration in modern grids;
- high resources utilization.

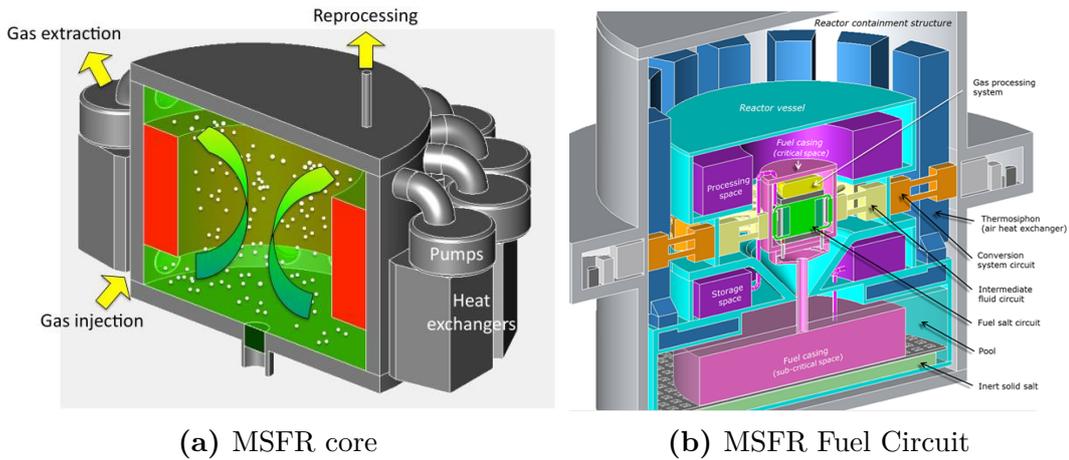
Some of the main technical challenges are:

- finding suitable combinations of salts composition and structural materials for containment;
- the characterization of the salt physical and chemical properties and the behaviour of the fission products in it;
- the lack of multi-physics simulations software which can be acceptable for a regulator body for licensing purposes.

### 1.3 The Molten Salt Fast Reactor

The MSFR is a further development of the '60s studies on the MSBR. The current MSFR, conceived during the EU projects SAMOFAR and SAMOSAFER, is a 3000  $MW_{th}$  reactor consisting of a cylindrical vessel with diameter and height of 2.25 m. The vessel, made of a nickel-based alloy, is filled with  $\approx 18 m^3$  of liquid salt, which acts both as fuel and as a coolant. The molten salt is operated under ambient pressure with a mean temperature of 750 °C and is pumped around the primary circuit in upward direction through the central core zone and in downward direction through the heat exchangers located circumferentially around the core. Between the core and the heat exchangers a container filled with a fertile blanket containing a thorium salt is present to increase the breeding gain. A dedicated heat removal system is needed to remove the power generated in the fertile blanket [9]. The MSFR plant is composed by three main circuits involved in power generation and extraction: the Fuel Circuit (FC), the Intermediate Circuit (IC) and the power conversion circuit/Gas Circuit (GC).

Figure 1.1 shows a detail of the core and of the FC. Figure 1.2 depicts a schematic view of the Balance of Plant (BoP).



**Figure 1.1:** Schematic views of the MSFR [9]

The fuel composition is being investigated based on different aspects involving chemistry, neutronics, burning capabilities, safety features and deployment capabilities. The optimal fuel salt composition, selected in the EU projects, is a binary fluoride salt, composed of LiF and a heavy nuclides (HN) mixture initially composed of fertile thorium and fissile matter. The MSFR can be operated with different fuel compositions, thanks to its online fuel control and flexible fuel processing: its initial fissile inventory can be composed of enriched natural uranium or transuranic (TRU) elements, currently produced by pressurized water reactors (PWRs), mainly

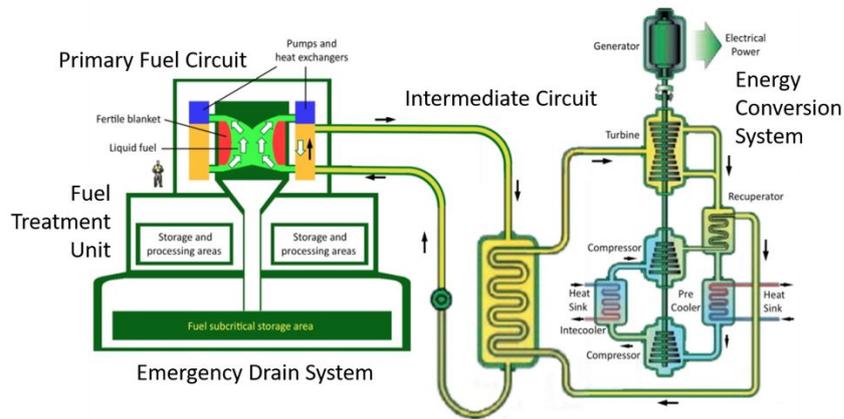


Figure 1.2: Schematic view of the BoP [9]

Plutonium and other minor actinides. The fission fragments produced in the salt are then removed in a salt treatment unit, with the main purpose of controlling thermophysical properties as neutron economy is weakly affected by fission products as the reactor is operated in the fast spectrum.

## 1.4 The MSFR simulation model

In this thesis, a model of the MSFR is used to investigate the system time-dependent response. The model consists in a power plant simulator developed by Politecnico di Milano during the SAMOFAR project [10]. The simulator is a control-oriented plant-dynamics tool, developed using the open-source, object-oriented Modelica language [11]. A one-dimensional modelling approach was used for thermal-hydraulics and heat transfer. For various plant components, standard and validated thermal-hydraulic Modelica libraries were used. On top of this an *ad hoc* neutronic MSR library was developed in order to model the peculiar behaviour of the motion of delayed precursors, which are extremely important in the reactor kinetics, and the circulating fuel.

Figure 1.3 depicts the power plant simulator in the Modelica environment and a conceptual scheme of the system.

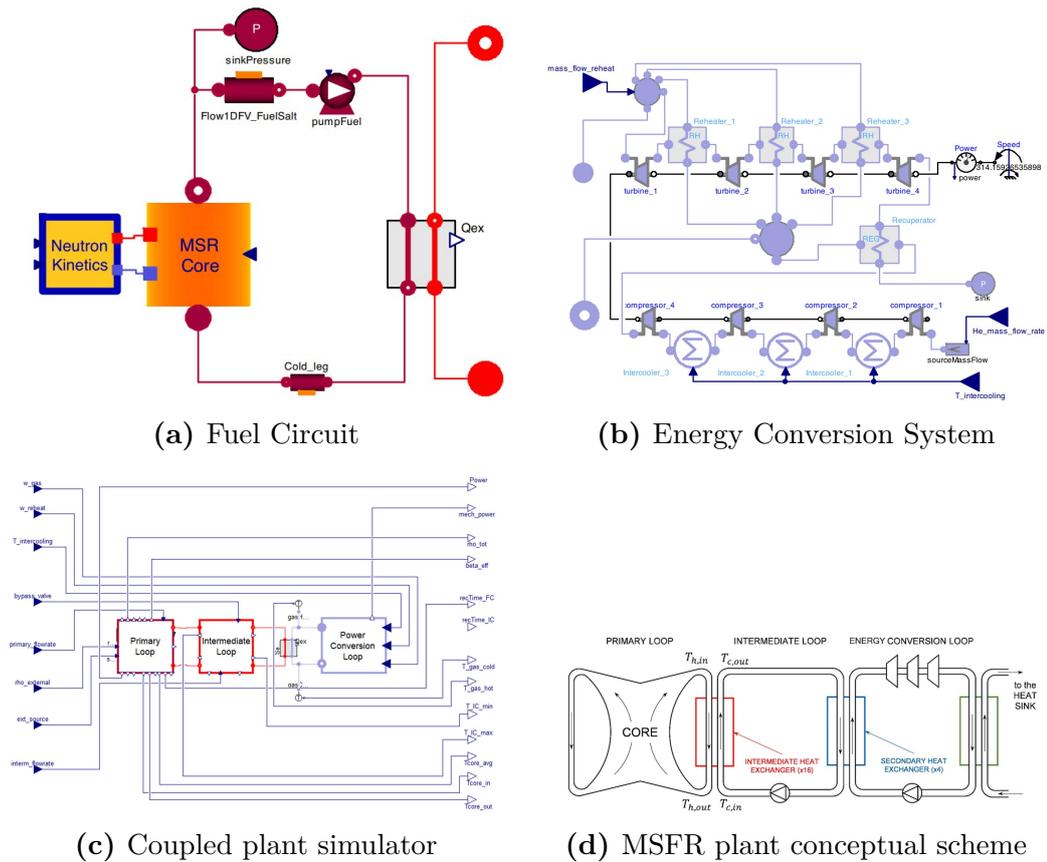


Figure 1.3: Object-oriented Modelica models and conceptual scheme [10]

## 1.5 Thesis objectives and outline

The first part of this section describes the goals of this thesis while the second part of the section summarizes the content of the next chapters of the thesis.

The deployment of the Gen-IV Molten Salt Fast Reactor (MSFRs) requires the demonstration of its enhanced safety features with respect to other reactor concepts. A relevant amount of work has been performed with this objective and is available in literature. Some works focus on modelling, design and analysis of the MSFR [12], [13]; other works focus on the safety issues of the MSFR [14], [15]. A number of data-driven works on the safety issues of nuclear reactor are available in literature [16], [17]. Nevertheless, to the best of our knowledge, this work is one of the first data-driven accident detection and classification approaches to the MSFR.

To this aim, the previously presented power plant simulator is used to describe the plant transient response to a variety of normal and abnormal operating conditions. Within this framework, the objectives of the present thesis are:

1. to propose a simulation-based method to characterize the system behaviour with respect to variations in physical and operational parameters, by means of a thorough exploration of the MSFR power plant state space;
2. to develop a data-driven algorithm for the efficient detection and classification of incidents, relying on a k-Nearest-Neighbor (kNN) classifier.

The proposed approach consists of the following steps:

- selection of a set of physical input parameters that are found to strongly influence the behaviour of the plant simulator, together with their ranges of variation;
- generation of several possible combinations of physical parameters values by random sampling and the corresponding time-varying (transient) behaviour of the MSFR is simulated. Finally, the time evolution of some relevant output plant parameters is analysed in detail;
- identification of normal and abnormal system output configurations;
- train, validate and test the kNN incident detection and classification model;
- retrieve those combinations of the reactor physical input variables that are responsible for the abnormal system states (namely, fault diagnosis).

The content of next chapters can be summarized as follows. Chapter 2 presents the rationale behind the use of the inputs (control variables) and outputs (controlled variables) adopted during the work; introduces the possible abnormal system status;

provides some information on the simulation settings and presents the different categorization rules adopted for classification purposes.

Chapter 3 describes all the steps of the analysis from a procedural point of view such as: the data generation; the production of exploration maps which act also as correctly labelled maps (namely, Ground Truth (GT)) in the classification part of the work; the procedures for building, validating, testing and evaluating the data-driven kNN algorithm; the fault diagnosis.

Chapters 4 and 5 provide the results coming from the application to two different Data Sets (DS) of the procedures described in chapter 3.

Finally, chapter 6 summarizes the work performed and its main outcomes, providing some conclusions and giving some future perspectives.

This thesis has been carried out in the framework of the SAMOSAFER project [9] and a portion of the work has been presented at the Young Molten Salt Reactor (YMSR) conference in Lecco, in June 2022 [18].

## Chapter 2

# Preliminary analysis of the MSFR model

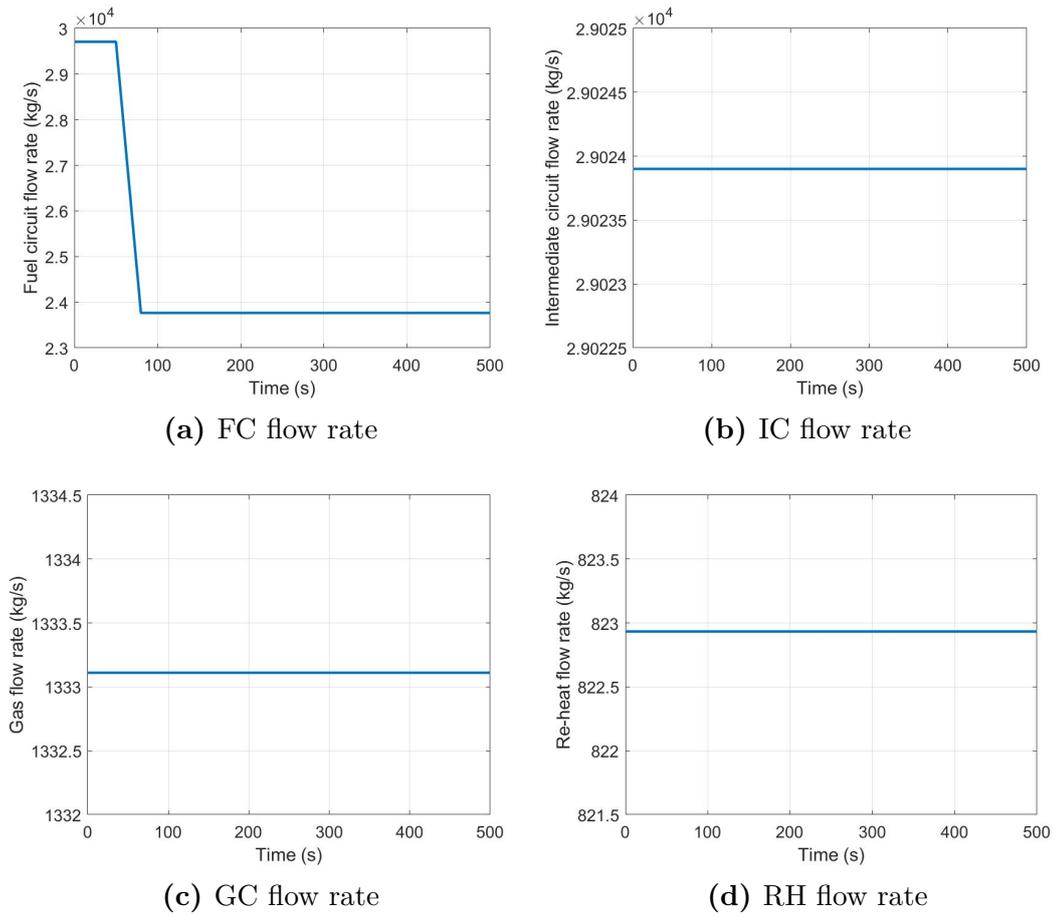
This chapter provides the information regarding the preliminary simulations and the final setup of the work. In section 2.1 the first step of this part of the work is discussed. Some initial simulations are run using the Modelica Model of the MSFR, in order to examine the behaviour of the system under inputs of different intensity and type. In section 2.2, the second step, which consisted in identifying some possible control and controlled variables suitable for the analysis, is shown. To support such choices, some additional simulations were run. In section 2.3 the definition of the possible states of a simulation are described. Section 2.4 shows the selection of a suitable range in which sampling the control variables for building the inputs of the simulations in the exploration phase of the work. Section 2.5 describes the simulation settings adopted while section 2.6 presents the categorization rules used during the classification part of the work.

### 2.1 Initial simulations

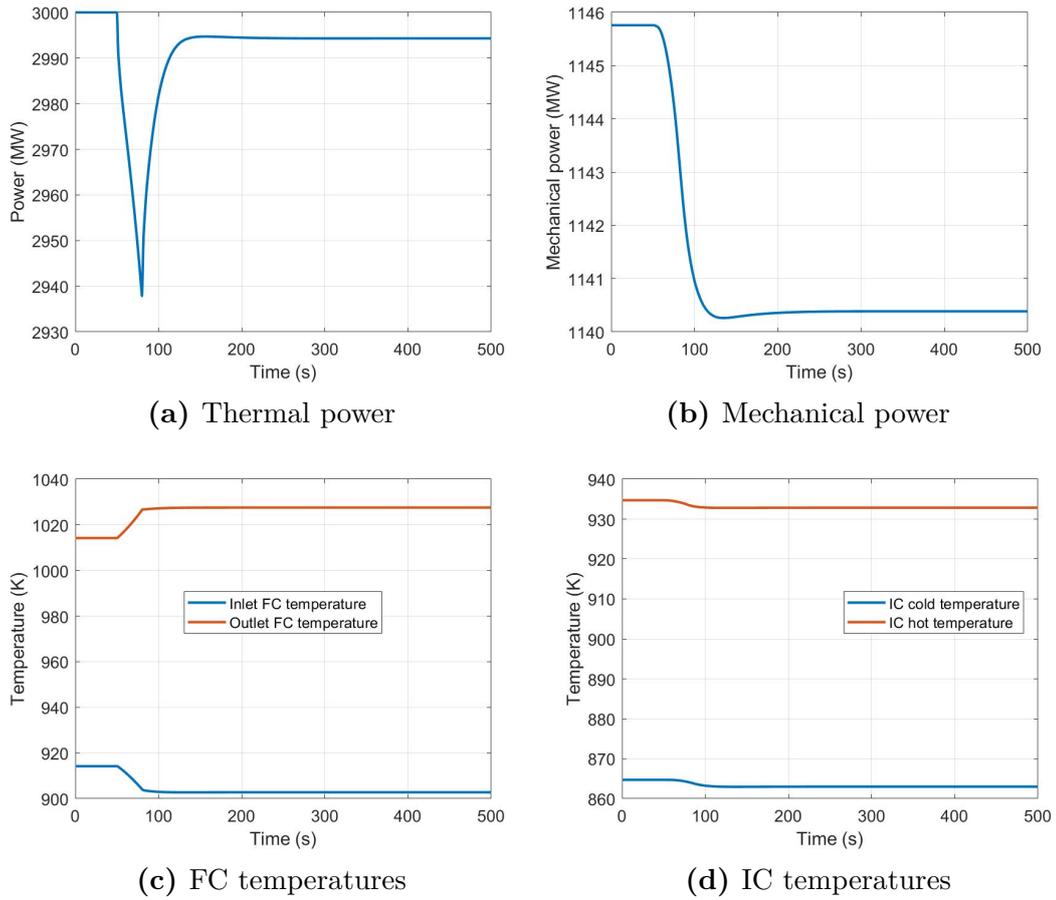
In order to test the behaviour of the Modelica Model of the MSFR, some preliminary simulations were run. Two simulations were run, characterized by mass flow rates variation. The mass flow rate variations are in the first simulation of linear type, in the second simulation of exponential type. The rationale for opting for the linear type of variation rests on its straightforward implementation and testing, while the exponential type is better suited for modeling the inertia of the circulation pumps. In the following the inputs and the outputs of two different sets of transient simulations are shown.

Concerning the first simulation a variation of the Fuel Circuit flow rate, figure 2.1, was used as input signal for the transient simulation of the system. The flow rate,

starting from its nominal value for the first seconds of the transient, is then subject to a linear variation with a reduction of 20% of its nominal value. In figure 2.2 the model outputs can be appreciated in terms of thermal and mechanical power and temperatures of the FC and IC. The transient can be described as follows: as soon as the FC mass flow starts decreasing, the core average temperature increases (as the salt slows down). This provides a prompt negative reactivity insertion, due to temperature feedback, that causes a reduction in the thermal power. The FC mass flow rate reduction leads to a decrease in the heat transfer rate in the intermediate HX, hence the intermediate salt outlet temperature starts to decrease 2.2d. The reduced intermediate salt temperature then causes a corresponding decrease in the temperature of the helium at the turbine admission, which leads to a reduction of the mechanical power output 2.2b. When the transient initiator ends, the increased fuel heating in the core can be removed in the intermediate HX by means of a larger temperature difference, leading to a new equilibrium [10].

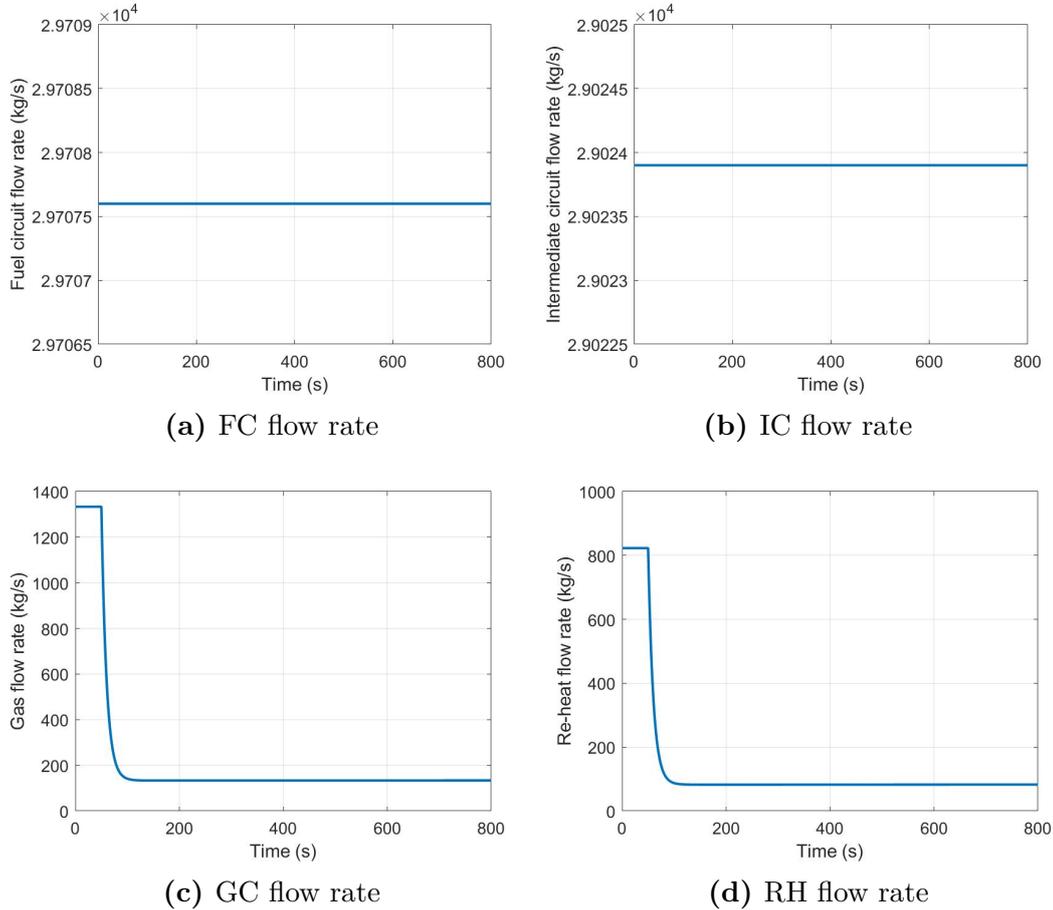


**Figure 2.1:** Input signals for the transient simulation caused by a 20% linear reduction of the FC flow rate

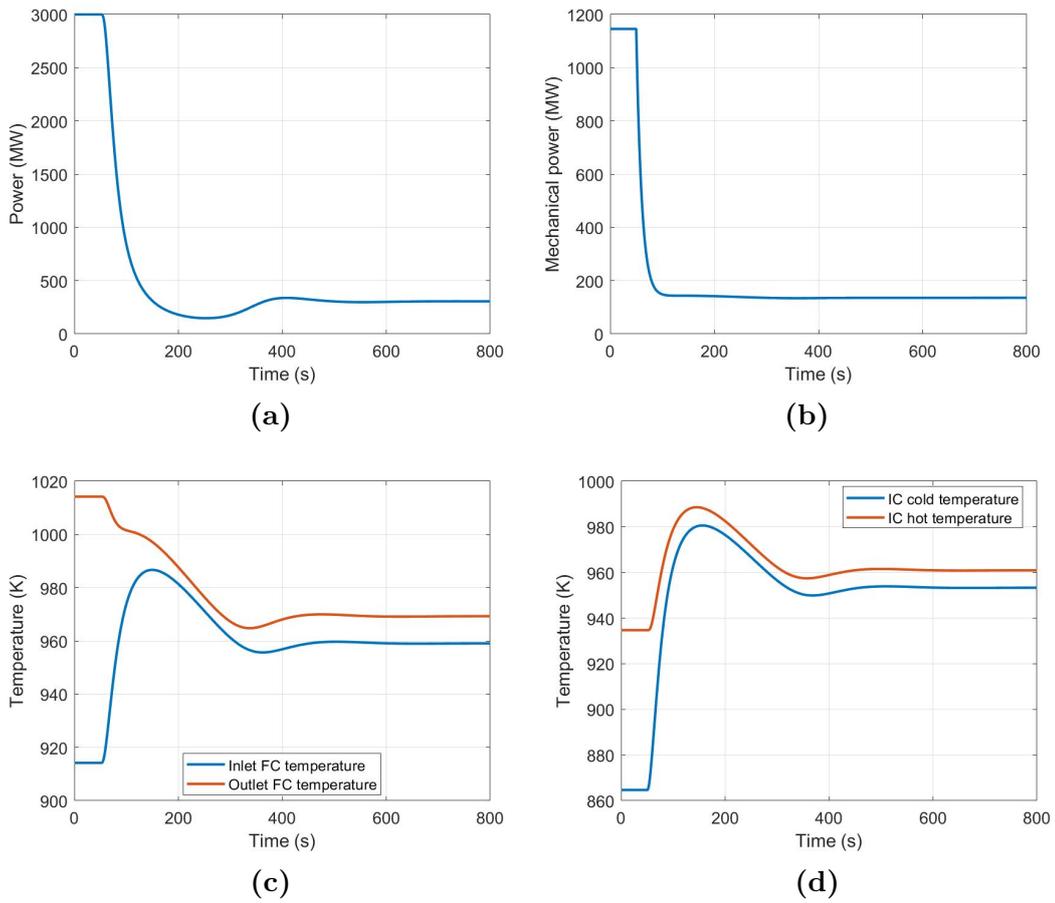


**Figure 2.2:** Output from the transient simulation caused by a 20% linear reduction of the FC flow rate

In the second simulation scenario an exponential variation of the GC flow rate and proportionally of the re-heat gas circuit, figure 2.3, was used as input signal for the transient simulation of the system. The GC flow rate, starting from its nominal value for the first seconds of the transient is subject to an exponential variation with a reduction of 90% of its nominal value. In figure 2.4 one can see the model outputs in terms of temperatures of the FC and IC. The transient can be described as follows: the reduction of the gas flow rate causes a decrease in the heat extraction from the IC. This causes an increase in the IC salt temperature that propagates the heat extraction reduction to the FC. Consequently the core average temperature rises causing a sharp reduction in the thermal power. The system then evolves to a new steady state with higher average temperatures and lower thermal power, with respect to nominal conditions.



**Figure 2.3:** Input signals for the transient simulation caused by a 90% exponential reduction of the GC and RH flow rates



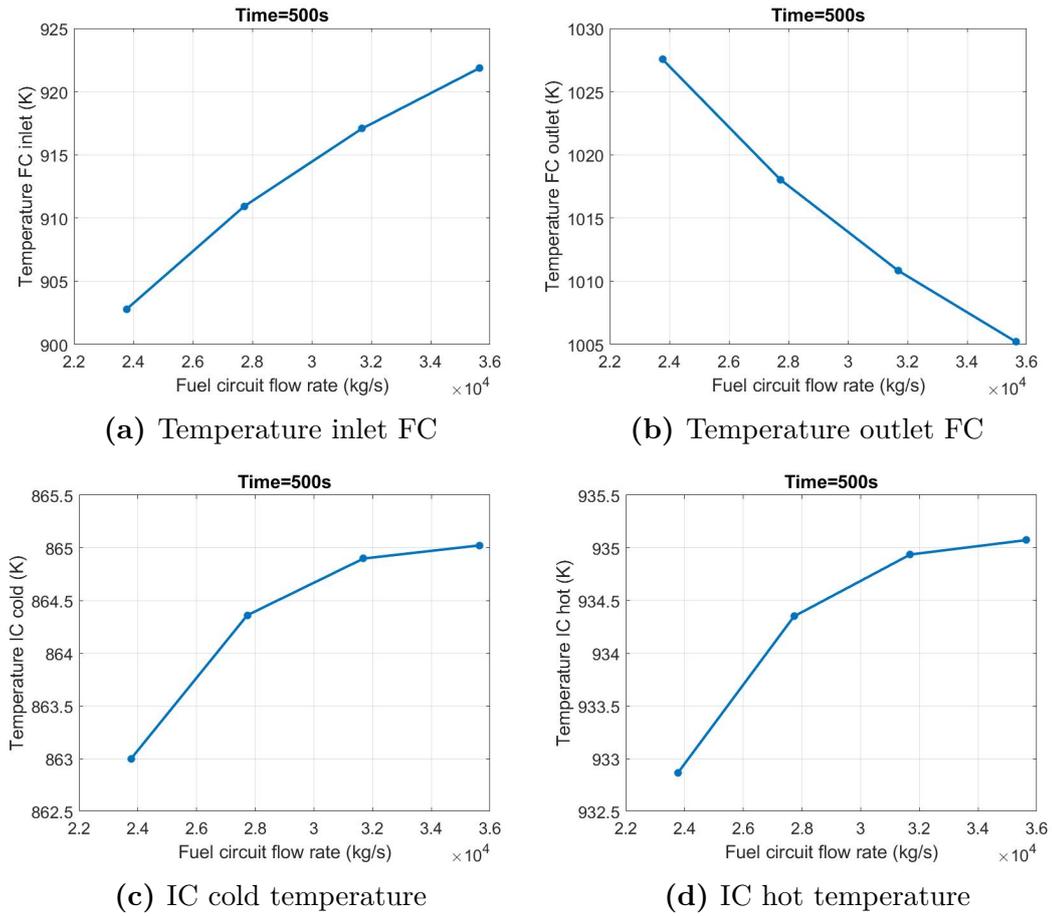
**Figure 2.4:** Output from the transient simulation caused by a 90% exponential reduction of the GC and RH flow rates: a) Thermal power b) Mechanical power c) FC temperatures d) IC temperatures

## 2.2 Selection of control and controlled variables

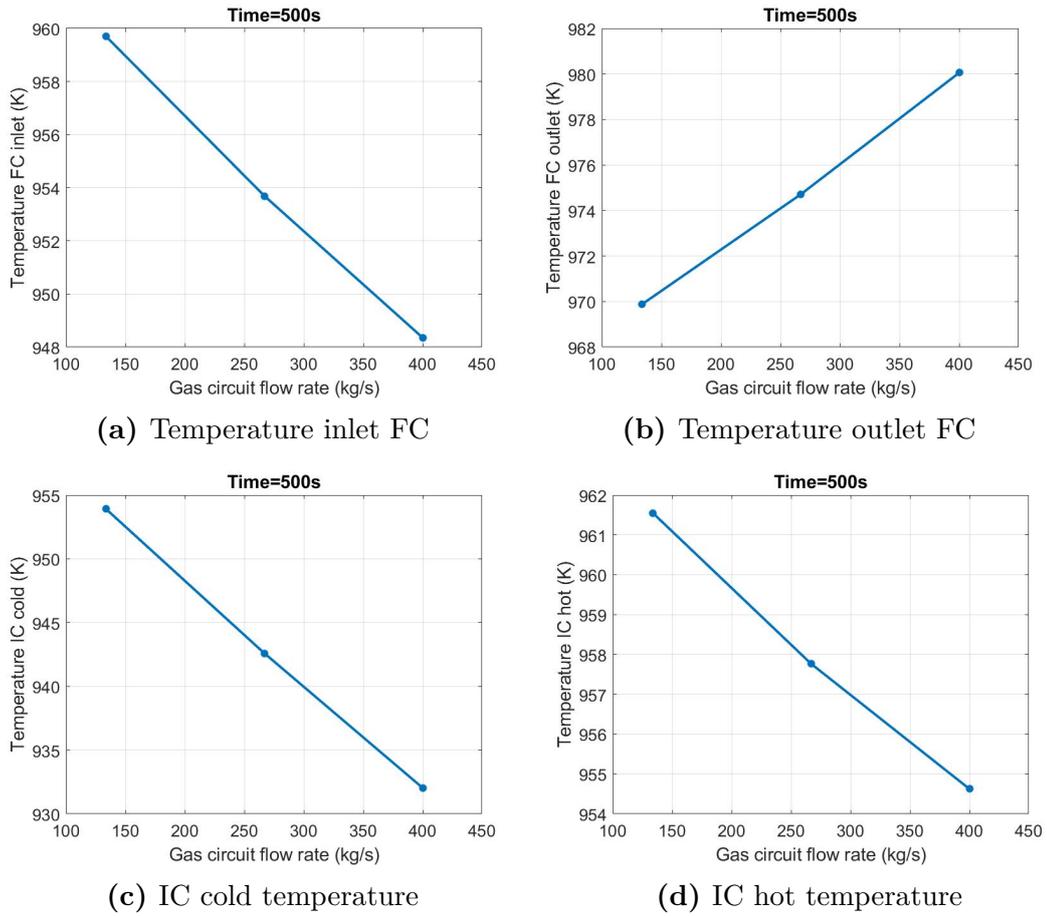
This section shows the investigation on which simulation inputs and which simulation outputs to use during the work. Regarding which control variables (inputs), the flow rates of the FC, IC and GC were selected, denoted as  $\dot{m}_{FC}$ ,  $\dot{m}_{IC}$ ,  $\dot{m}_{GC}$  respectively. By tuning the intensity variation of these variables, different inputs could be provided to the Modelica model for simulating different operational scenarios.

For what concerns the controlled variables (outputs), the FC (core) inlet and (core) outlet temperatures and the IC cold (heat-exchanger inlet) and hot (heat-exchanger outlet) temperatures were chosen. The choice comes from the fact that these signals can be measured in real time by instruments. The capability to measure in real time the output signals is fundamental for the final aim of the work, which is accident detection.

Before starting with the analysis, in order to see in which relationship the controlled variables are with the control variables, the following analysis was set up. All the control variables were kept constant, except one that was sampled in an investigation interval. Building in this way different inputs for the Modelica Model, different simulations were obtained. From each simulation, the value of all the controlled variables was extracted at the same time instant of  $t=500s$  (a time instant in which all the outputs are considered to have reached their steady state) and represented versus the control variable value at that time instant in each simulation. The results are shown in figure 2.5 and 2.6 respectively for FC linear variations of  $[-20, -16, +16, +20]\%$  of the nominal value and GC exponential variations of  $-[90, 80, 70]\%$  of the nominal value. These figures show that their relationship is not constant. On the contrary, being almost linear means that there is some input-output dependence, making the choice of control and controlled variables suitable for the continuation of the analysis.



**Figure 2.5:** Behaviour of controlled variables due to different linear FC flow rate reductions



**Figure 2.6:** Behaviour of controlled variables due to different exponential GC flow rate reductions

To summarize, the following control variables have been selected:

- **fuel circuit molten salt mass flow rate:**  $\dot{m}_{FC}$  ;
- **intermediate circuit molten salt mass flow rate:**  $\dot{m}_{IC}$  ;
- **gas circuit (and proportionally the re-heat gas circuit) mass flow rate:**  $\dot{m}_{GC}$  .

For what concerns the outputs, the following controlled variables have been chosen:

- **inlet core temperature:**  $T_{FC\ inlet}$ ;
- **outlet core temperature:**  $T_{FC\ outlet}$ ;
- **IC cold temperature (inlet HX):**  $T_{IC\ cold}$ ;
- **IC hot temperature (outlet HX):**  $T_{IC\ hot}$ .

## 2.3 Definition of the simulation status

In order to correctly classify each simulation there is the need for defining its status. This section outlines the criteria for determining if the system has failed its mission (i.e. producing power within the safety margins of the plant) during the transient simulation or if it remains in a successful state.

A particular type of simulation failure is indicated in this work as "Numerical Failure" (NF), when the simulation terminates prior to reaching its intended end time. This failure can occur when one or more variables exceed the acceptable limits during the simulation's transient phase, leading to errors in the property evaluation process of the molten salt.

A second type of failure, more interesting for the objective of this thesis, is the one due to physical constrain violation. In particular a maximum allowable temperature  $T_{max} = 1373K$ , which characterizes the temperature at which structural damages can occur, and a minimum allowable temperature  $T_{min} = 858K$ , which defines the salt freezing point in the FC, are introduced. These two thresholds constitute an upper and lower bound for the temperature value of the molten salt outside which the system can be considered failed. Given these definitions the following status are possible for a given simulation:

- **Success:** the simulation stops at the requested end simulation time; the molten salt temperature stays between the maximum and minimum allowable temperatures during the whole transient;
- **Numerical Failure:** the simulation stops before the requested end simulation time;
- **High Temperature Fuel Circuit ( $HT_{FC}$ ):** the molten salt temperature exceeds  $T_{max}$  in the FC;
- **Low Temperature Fuel Circuit ( $LT_{FC}$ ):** the molten salt temperature goes below  $T_{min}$  in the FC;
- **High Temperature Intermediate Circuit ( $HT_{IC}$ ):** the molten salt temperature exceeds  $T_{max}$  in the IC;
- **Low Temperature Intermediate Circuit ( $LT_{IC}$ ):** the molten salt temperature goes below  $T_{min}$  in the IC;

In figure 2.7 the temperatures evolution of two different simulations are shown. In figure 2.7a an example of transient which ends before the requested end simulation time is presented; on the right, 2.7b displays an example of  $LT_{IC}$  failure, a status in which the lower temperature constrain is violated during the simulation.

An important note about numerically failed simulation needs to be drawn to the

attention of the reader. Specifically, two different types of data were used in the analysis. The first type of data is numerical failure-free, meaning that all simulations that failed numerically were removed from the data set before any post-processing was attempted. The second type of data used contained simulations that failed numerically but were manipulated to be consistent with the rest of the data to allow their post-processed. The manipulation consisted of extending the value of the output signals at the time of failure, maintaining the value of the variable constant until the end of the simulation. Figure 2.8 depicts an example of numerically failed simulation in which the inlet FC temperature signal is manipulated. Specifically it can be seen that between 80 and 120 seconds the signal is characterized by rapid changes a characteristic peculiar to simulations that fail numerically. Around 120 seconds the point in which the signal would have ended is prolonged as a constant.

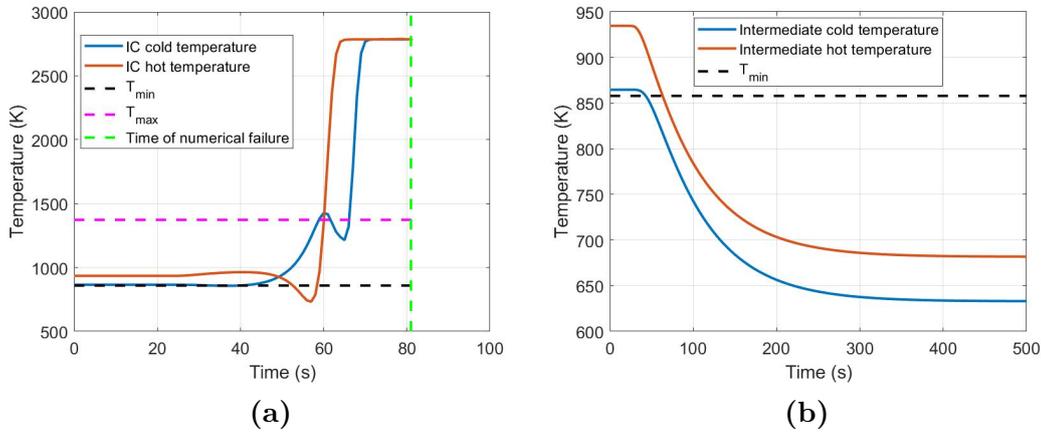


Figure 2.7: a) Numerical Failure b)  $LT_{IC}$  failure

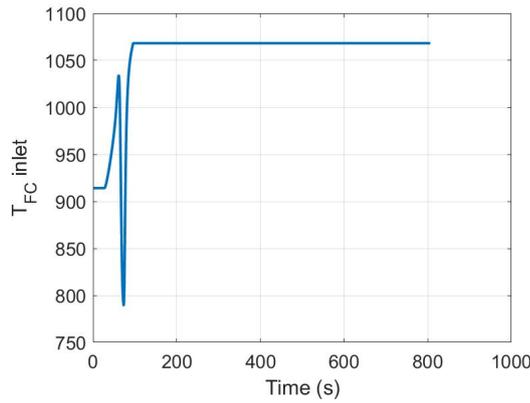


Figure 2.8: Signal extension until the end simulation time

## 2.4 Range definition of control variables

To determine an appropriate range for the control variables during the exploration phase, the individual ranges in which each control variable could vary independently were studied. The result was that standalone intensity variations of  $-98\%$ ,  $-80\%$ ,  $-94\%$  of FC flow rate, IC flow rate, GC flow rate respectively, were able to induce numerical failures in the simulator. On the other hand, in the following of the analysis, combination of different flow rates reductions are used as simulation inputs. To reduce the risk of producing too much data with numerical failures, a conservative range was selected for control variables sampling. This approach is based on the *a priori* assumption that the combined effects of the inputs would not provide beneficial compensation towards safer states. For these reasons, the range  $[-90; 0]\%$  reduction from nominal value was chosen for each control variable.

## 2.5 Simulation settings

The simulations were run using the Modelica model of the MSFR. All the simulations were run in free dynamics conditions. The study of the plant free dynamics is a fundamental step in order to understand the behavior of the reactor in response to different transient initiators [10]. The simulations started from steady state nominal conditions and at time  $t = 50\text{s}$  all three control variables, namely FC flow rate, IC flow rate, GC flow rate, started varying according to the specific input of each simulation. For each input a transient lasting 800 seconds was simulated. In order to be sure that the time step of the time grid used to solve the problem was not impacting the results some preliminary simulations were run. Figure 2.9 shows the four controlled variables evolution using different time steps. Figure 2.10a shows the controlled variables value at a specified time instant, for different time steps. Figure 2.10b shows the error, relative to a benchmark simulation using the finest time step of  $\Delta t = 0.01\text{s}$ , of using a coarser time step. From all these figures, it can be appreciated qualitatively 2.9 and quantitatively 2.10 how the choice of the time step is quite irrelevant to the quality of the solution. This is justified by the fact that the solver uses, to compute the solution of the problem, an adaptive time step when needed. For this reason, for optimizing the time step value, other metrics had to be used. Figure 2.11 depicts the file dimension (2.11a) and simulation time (2.11b) reductions obtained by using larger time steps with respect to a benchmark simulation using the finest time step. It can be seen that around a time step of one second, both curves flatten providing low to no benefits in choosing a coarser time step. For these reasons, a time step of 1 second was enough to provide time-step independent results for a reasonable computational time and output file dimension. Each simulation took about 90 seconds to run.

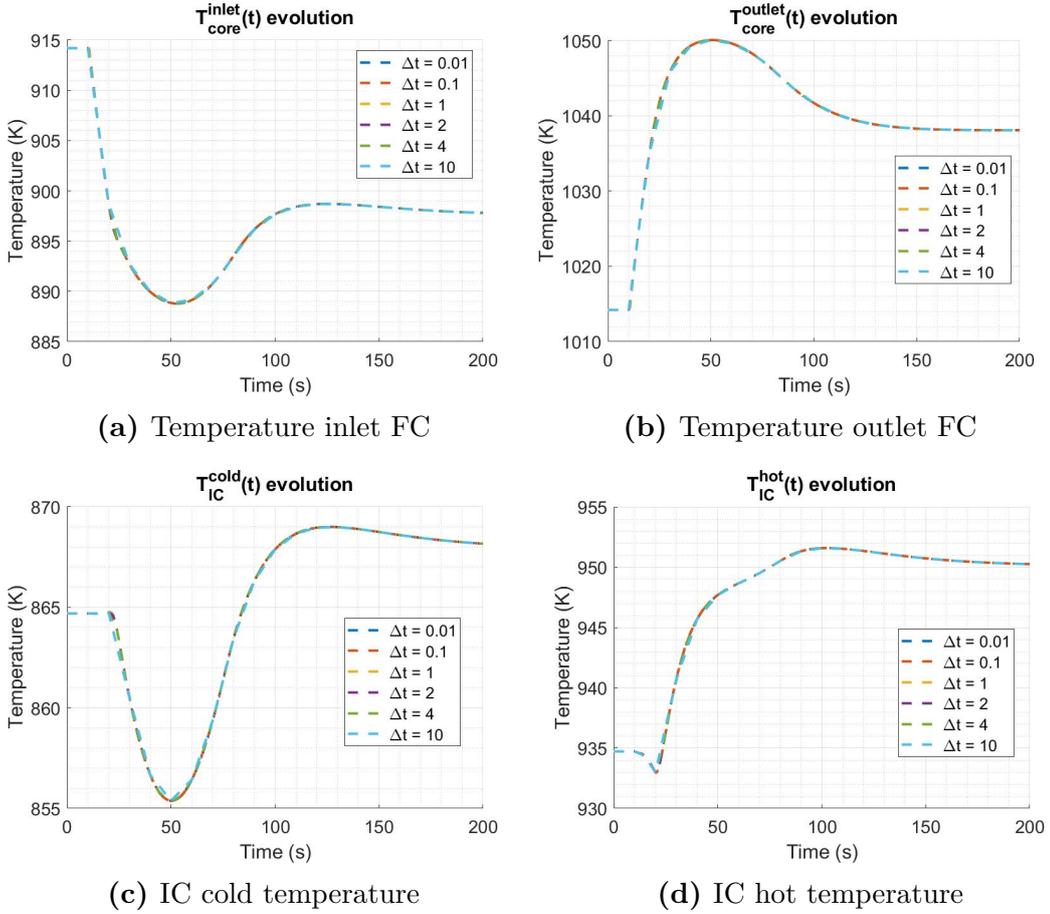
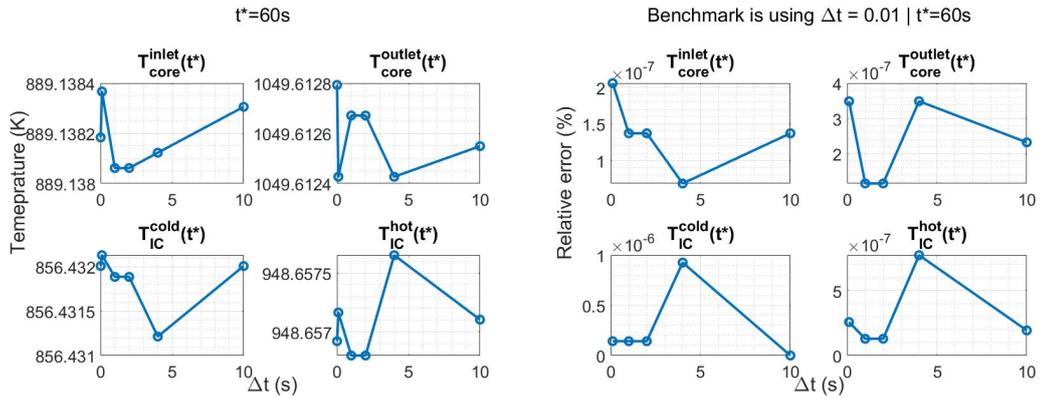
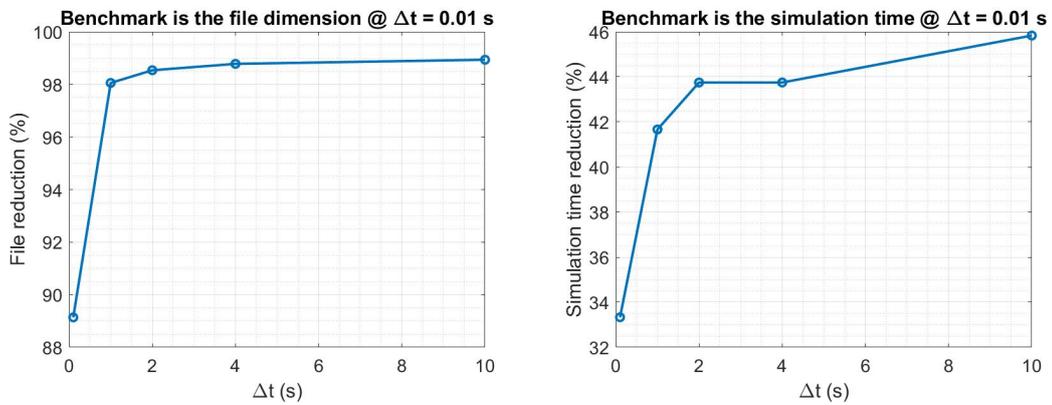


Figure 2.9: Time evolution of the controlled variables using different time steps.



(a) Temperature at  $t^* = 60$  s for different time steps (b) Error relative to benchmark simulation with finest time step

Figure 2.10: Quality of the results for different time steps



(a) File size reduction for different time steps (b) Simulation time reduction for different time steps

**Figure 2.11:** File size and simulation time reductions relative to benchmark simulation with finest time step.

## 2.6 Categorization rules

According to the definitions of the simulation status discussed in section 2.3, three different categorizations have been developed. Each category defines what are the possible classes and the rules for the attribution of the class name to a simulation. For this reason, a given simulation can be classified differently depending on the categorization rule used. The categories are called C1, C2 and C3 and are defined as follows:

- C1: the classes can be "Success", "Numerical Failure" (if present) or "Physical Failure" where the latter class is assigned if the simulation status is of  $HT_{FC}$  or  $LT_{FC}$  or  $HT_{IC}$  or  $LT_{IC}$ ;
- C2: the class name can either be "Success" or is determined by identifying the first type of failure to occur among the various controlled variables throughout the entire transient period;
- C3: the class name can either be "Success" or a list of the names of all the failures that occur during the simulation time.

It is evident that the categorization rules go from simpler to more challenging classification purposes going from C1 to C3. Furthermore the different categorization rules can be used for different purposes. For example C1 can give an idea of what is a safe operative zone; C2 is the more appropriate for accident identification purposes; C3 is for testing the developed data-driven classifier in a potential large multi-class problem.

# Chapter 3

## Methodology

This chapter provides an overview of the techniques used in the subsequent chapters without showing actual results. Since similar procedures were applied to different set of data throughout this work the reader can refer to this chapter for the methodology when looking at the different results presented in the following. Chapters 4 and 5 demonstrate the application of these techniques and present the corresponding results.

Throughout the work, four distinct data sets were employed, each varying in terms of the number of simulations used for both the Training Set (TRS) and Test Set (TES), as well as the inclusion or exclusion of NF simulations. Table 3.1 provides a summary of the characteristics of the different data sets.

Types of data used			
Data set name	TRS (# of sim)	TES (# of sim)	NF simulations
DS1A	1000	250	Included
DS1B	780	207	Removed
DS2A	5982	1747	Included
DS2B	5003	1478	Removed

**Table 3.1:** Type of Data Sets used during the analysis

Figure 3.1 depicts, in the form of a flow chart, all the steps of the analysis. In the flow chart,  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  represent the controlled variables fuel circuit inlet and outlet temperature, intermediate circuit cold and hot temperature, respectively. Being the controlled variables, outputs of simulations, they are described as vectors of points evaluated at different time instants as follows:

$$T(t) \quad \text{where} \quad t \in t_0, t_1, t_2, \dots, t_n$$

Here,  $T(t)$  is the generic controlled variable at time  $t$ , and  $t_0, t_1, t_2, \dots, t_n$  are the discrete time grid points where the signal is defined.

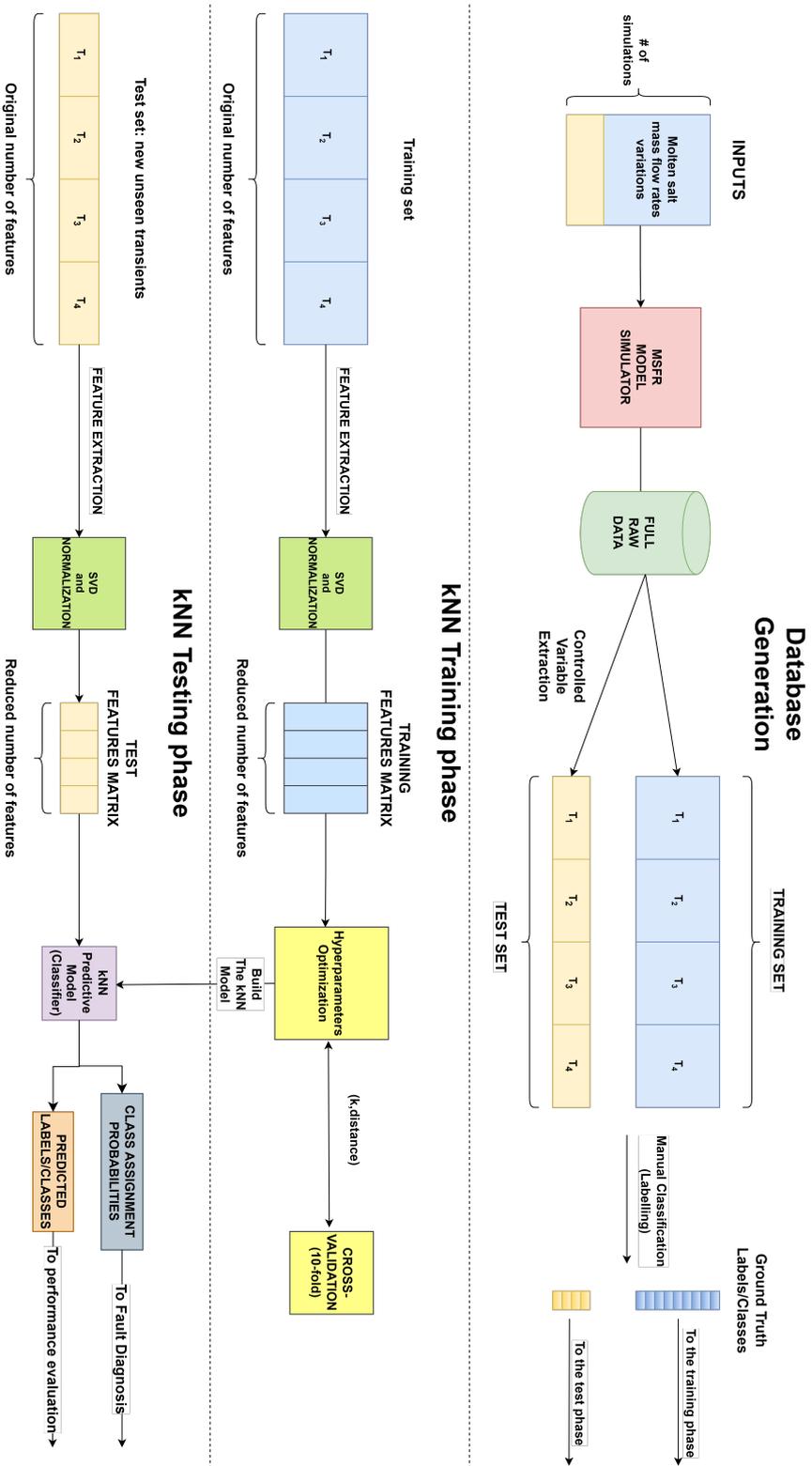


Figure 3.1: Flow chart representation of the analysis steps

### Data generation

The first step of the analysis consisted in generating the data by sampling points in the simulation input space, by varying the control variables  $\dot{m}_{FC}$ ,  $\dot{m}_{IC}$ ,  $\dot{m}_{GC}$  from nominal conditions. These sampled points are then used to generate the input signals for the simulations of the MSFR. All the control variables undergo exponential transitions (a good model for the description of the physical behaviour of circulation pumps due to inertia) when moving from nominal conditions to new values. All the simulations are performed in free dynamics (i.e., the plant response with no control actions). Once the simulations were performed running the four temperature signals designated as controlled variables (section 2.2) were extracted, using *ad hoc* MATLAB [19] scripts, and used for representing their simulation in the post-processing part.

### Exploration maps

The second step of the analysis consisted in making exploration maps of both the training and test set, used as ground truth maps in the following part of the work. The expression "ground truth" refers to the accurate data that is used as a reference/benchmark against which other data (the classifier predictions) can be compared. Given a data set, six different ground truth maps were produced, three for the training set and three for the test set. The reason for which there are more than one ground truth map for a given training or test set is due to the fact that different classification rules were used as discussed in 2.6.

### Time evolution of controlled variables

The third step of the analysis consisted in plotting the time transients of the controlled variables (output). This step allows the visualization of the behaviour of the output space.

### 3.1 Building a $k$ NN classifier

The fourth step was the central part of the analysis and consisted in building a  $k$ -nearest neighbor ( $k$ NN) classifier [20].

In this work the  $k$ NN algorithm is used for classification purposes. In order to use a  $k$ NN algorithm a list of examples with their features needs to be used as a training set. The definitions of these terms are:

- example/observation/sample: it refers to a single instance of data in the training or test set, that is used to train a model or evaluate its performance. An example is typically represented as a vector of features [21];
- features: it refers to the attributes or characteristics of the data points being used to train the model or being classified [21];

A  $k$ NN classifier is a type of non-parametric supervised machine learning algorithm that can be used for both classification and regression tasks. The  $k$ NN algorithm is based on the idea of feature similarity, which assumes that examples that are similar in terms of their feature values are likely to belong to the same class. One of the key advantages of the  $k$ NN algorithm is its simplicity, as it requires minimal training. However, the algorithm can be computationally expensive, as it requires the calculation of the distance between a test example and all training examples for each prediction. Additionally, the algorithm can be sensitive to the choice of  $k$  and the distance metric used, which can affect the performance of the classifier. For this reason an hyperparameters optimization routine was implemented during the cross-validation step as described in subsection 3.1.3.

The data analysis pipeline used in this work can be described as follows:

- sampling the intensity variations of the control variables and generation of the simulation input;
- system transient simulation;
- output extraction of the controlled variables from simulation;
- feature selection (reduction) from the raw data. In order to reduce the number of features of each example the Singular Value Decomposition (SVD) technique was used;
- normalization: it is common practice to normalize data before any training or classification is attempted;
- cross validation: the training set is subject to a cross validation process for the optimization of the  $k$  parameter and the distance function;

- application to the test set: the newly built, cross validated and optimized  $k$ NN classifier is used to make predictions on the test set, which is independent from the training set.

In order to make predictions on the test set, the MATLAB function *predict* [22] was used. The function classifies by minimizing the expected misclassification cost:

$$\hat{y} = \arg \min_{y=1,\dots,n_c} \sum_{j=1}^{n_c} \hat{P}(j|x)C(y|j) \quad (3.1)$$

where:

- $\hat{y}$  is the predicted classification;
- $n_c$  is the number of classes;
- $\hat{P}(j|x)$  is the posterior probability of class  $j$  for observation  $x$ ;
- $C(y|j)$  is the cost of classifying an observation as  $y$  when its true class is  $j$ .

The posterior probability is defined as follows:

$$\hat{P}(j|x) = \frac{\sum_{i \in nbd} W(i) \cdot 1_{Y(X(i))=j}}{\sum_{i \in nbd} W(i)} \quad (3.2)$$

where:

- Where  $W(i)$  is the weight associated to the  $i$ -th data point;
- $Y(X(i))$  is the true class of the feature vector  $X(i)$ ;
- $1_{Y(X(i))=j}$  is 1 when  $Y(X(i)) = j$  and 0 otherwise.

The posterior probability is used in sections 3.2 and 5.6 to perform fault diagnosis. In the next sections, a more in depth description of the steps performed for building the classifier are presented.

### 3.1.1 Feature extraction for dimensionality reduction

This step consists in reducing the number of features of a training or test observation in order to improve the performances of the  $k$ NN classifier while reducing the computational resources.

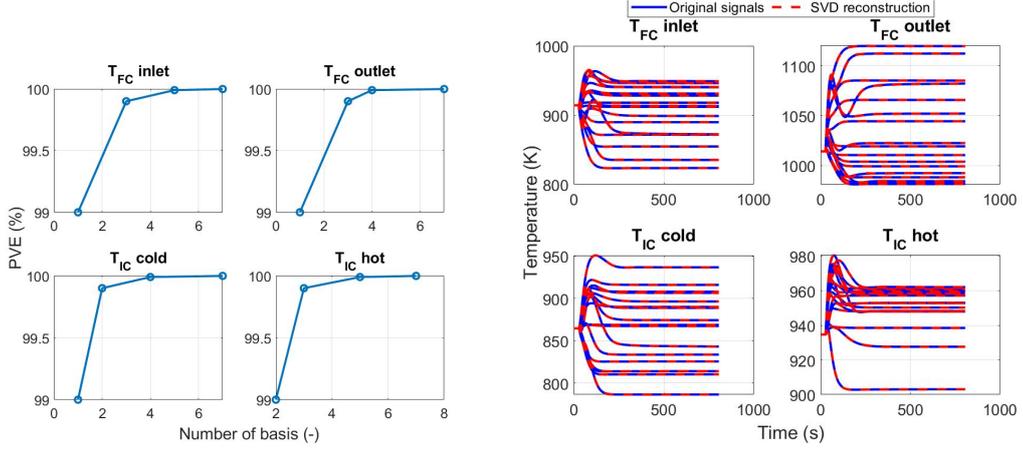
Let us assume that no treatment of the data is performed yet. Then the feature matrix, i.e. the list of all the features (columns) for each example (rows), would be generated as follows. Starting from a simulation, the four controlled variables are extracted as four different vectors. Note that each controlled variable is a vector, because it contains the values of the physical quantities of interest at the different time instants of the transient. The four vectors of the controlled variables are then concatenated forming a single row vector of dimension  $(1, 4n_t)$  where  $n_t$  is the number of time-steps in which the transient is discretized. This is done for each simulation at disposal leading to a matrix  $(m, 4n_t)$  where  $m$  is the number of samples. In this way the  $m$ -th row of the matrix is a sample and in the corresponding columns all the features associated are listed.

In order to reduce the number of features while not losing important data the Singular Value Decomposition (SVD) [23] technique can be used. If this step is used the procedure to build the feature matrix is changed as follows. Four matrices, each containing all the transients of a controlled variable are built. Each matrix is then subject to the SVD procedure and the feature matrix is obtained by concatenating the four SVD-reduced matrices. The resulting size of the feature matrix in this way is  $(m, p)$  with  $p \ll 4n_t$ .

It is important to notice that, for each matrix-signal, the SVD was applied firstly to the training set, obtaining the SVD coefficients and a set of basis that can describe that data. Secondly, the corresponding test set signals were cast in a matrix form and projected obtaining the corresponding SVD coefficients.

Figure 3.2 depicts how the SVD quality is assessed, using training set data. Figure 3.2a shows how many basis are needed for the SVD decomposition as the Percentage of Variance Explained (PVE) requested is increased. It can be noted that a modest number of basis is associated to high levels of PVE requested. This can be explained as the data under analysis come from simulations and are noise free.

Figure 3.2b shows the original signals and the reconstructed version using seven basis for each signal (corresponding to a request on the PVE of 99.999%). Figure 3.2a depicts that a relatively low number of basis are needed to describe such a large request on the PVE. This is justified by the fact that the data set is of numerical simulation nature and hence free of experimental noises. Figure 3.2b provides a qualitative measure of how well the original training set signals have been reconstructed using seven basis for each signal.



(a) Number of basis vs requested PVE (b) Original signals and SVD reconstruction

Figure 3.2: SVD quality

### 3.1.2 Normalization

In machine learning, normalization refers to the process of transforming the features of a data set to a standardized scale. This is done to ensure that all features have equal importance in the analysis and that the model can learn from the data effectively. There are a few reasons why normalization is a common practice in machine learning:

- different features may have different units of measurement, ranges, or scales;
- normalization ensures that all features are on the same scale so that they can be compared and weighted equally;
- normalization can help to prevent the model from being biased towards certain features. For example, if one feature has a much larger range than the others, it may dominate the model's predictions, even if it is not actually the most important feature.

Equation (3.3) describes the normalization routine used in this work

$$\hat{\underline{\underline{M}}} = \min_{new} + \frac{(\max_{new} - \min_{new}) \cdot (\underline{\underline{M}} - \min_{old})}{(\max_{old} - \min_{old})} \quad (3.3)$$

where  $\min_{new}$ ,  $\max_{new}$  represent the minimum and maximum of the new interval range;  $\min_{old}$ ,  $\max_{old}$  represent the minimum and maximum of the old interval range;  $\hat{\underline{\underline{M}}}$ ,  $\underline{\underline{M}}$  represent the newly normalized matrix and the matrix to normalize. Two different normalization approaches were used. In the first one the data is

normalized, then the SVD is applied and the feature matrices derived; in the second one, the normalization is carried again after the SVD.

In this specific application all the features come from temperature signals thus having values of the same units of measurements and ranging in similar intervals. Therefore also a non normalized configuration was investigated. In this way, 3 different group of data are used in the following of the work: one in which the data is not normalized, one in which the data is normalized once before the SVD and another in which data is normalized twice: before and after the SVD.

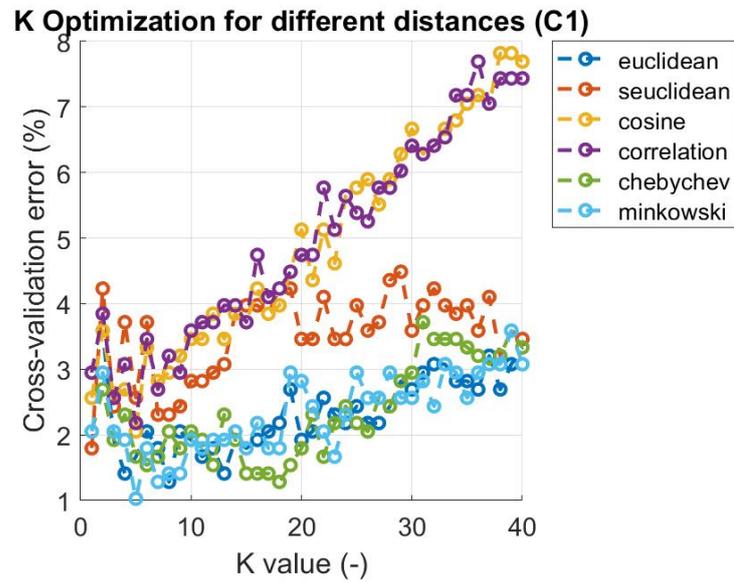
It is important to notice that for each matrix-signal, the normalization was applied firstly to the training set storing the maximum and the minimum values. Secondly using the previously computed maximum and minimum, the normalized test set matrix-signal was computed.

### 3.1.3 Hyperparameters optimization and cross-validation

In the context of Machine Learning (ML), the term parameters is used to refer to something that can be learned by the algorithm during training while the term hyperparameters refers to something that is passed to the algorithm. The  $k$ NN is a type of non-parametric ML algorithm, which means it does not make any assumptions about the underlying distribution of the data. The aim of this routine was to find the best combination of the hyperparameters  $k$  (the number of nearest neighbors) and distance function, which could minimize the cross-validation error of the  $k$ NN classifier.

The optimal number of neighbors depends on the size of the data set in terms of both observations and features and predicting an optimum value is impossible. The choice of the number of neighbors strongly impacts the model's behavior. A lower value of  $k$  can overfit the data, whereas an higher value of  $k$  tends to "smooth out" the prediction by averaging the values over a greater neighborhood. However, if the value of  $k$  is too high, then the model can underfit the data [24]. In order to be sure to find the best value of  $k$  a large exploration interval from 1 to 40 was used. The optimization routine involved first choosing a distance function and then iterating through the various values of  $k$  in the selected range. For each combination of  $k$  and distance function, the  $k$ NN classifier was trained, cross-validated, and the miss-classification error was stored. After these steps, the  $k$  values were tested for a given distance function and the  $k$  value that was associated with the minimum miss-classification error was stored. The whole process was then repeated for each distance function in the list of the possible candidates. At the end of the routine, the analyst had a vector of the best  $k$  values for each distance function used. The final step consisted, by using this vector, to choose the optimal combination of hyperparameters ( $k$  and distance function) that provided the smallest miss-classification error.

Figure 3.3 allows to visualize the output of the hyperparameters optimization routine. In this case the couple of hyperparameters which is associated to the minimum cross-validation error is  $k=6$  using the minkowski distance.



**Figure 3.3:** Hyperparameters optimization in cross-validation routine

### 3.1.4 Testing and evaluation

The purpose of the testing and evaluation phase was to assess the accuracy of the newly trained  $k$ NN classifier on unseen data which is the test set. This phase was crucial to determine whether the model could generalize well to new, unseen data. To evaluate the model's performance, several types of tools were adopted, including prediction maps, correct or miss-classified maps, and confusion matrix charts.

- **Prediction maps** were generated to visualize the spatial distribution of the predicted classes in the simulations input space.
- **Correct or misclassified maps** are created, which highlight the areas, in the simulations input space, where the model performed well or poorly in terms of classification accuracy. These maps helped identifying the regions in which the classifier performed well or poorly.
- **Confusion matrix charts** to provide a summary of the model's classification performance, were used. A confusion matrix can be used to analyze the performances of the classifier. Along the rows the correct labels are listed while along the columns the classifier predictions. Hence the diagonal elements denote correctly classified outcomes. The misclassified outcomes are the elements of the off diagonals of the confusion matrix. For this reason the best classifier will have a confusion matrix with only diagonal elements and no off diagonals elements.

## 3.2 Inference on inputs

The last step of the work consisted in using the classifier outputs for performing inference on the simulations inputs. The idea was to retrieve those combinations of the reactor physical input variables (e.g., circulation pumps failures) that were responsible for the abnormal system states. The process can be described in more details as follows: a triplet of pump failure intensity generates a transient simulation from which the usual features are extracted generating a query point (a feature vector for which the output label is unknown). The query point is then classified by the model which predicts the probabilities that the query belongs to the different classes. The aim of this part of the analysis consisted in the following: given an example which is classified by the model with a given label, going back to what were the inputs that generated that transient from which the features for building that example were extracted. In this way, when the classifier outputs a class, one could know the causes and intervene, trying to avoid them if possible or simply take actions accordingly. In more detail this part of the analysis consisted in the following steps:

- construction of histograms showing the number of observations in the training set that belong to each class, for each pump failure intensity range (3 histograms); these represent the probability distributions of the pumps failure intensities (inputs) conditional on the different transients (output) classes;
- normalization of the histograms to get the conditional Probability Distribution Functions (PDFs) that belong to each class, for each pump failure intensity range (3 conditional PDFs), which possibly represent a "new" transient coming from an hypothetical real MSFR plant. ;
- construction of the PDFs of some examples of the test set. This was done by using the previously built conditional PDFs and the probabilities to belong in each class for the example (these probabilities are provided by the classifier). With these instruments and by using the total probability formula the PDFs of the test example were obtained.
- use of the newly built PDFs of the test set example for identifying the causes of the transients in terms of pumps failures. This simulates an hypothetical real application of the classifier-based inference: a new transient coming from the "real" MSFR is first detected and classified (in terms of failure type); then, the combination of pump failure intensities that were most likely responsible of the plant anomalous behavior are identified (and possibly inspected and maintained).

## Chapter 4

# Results on coarse sampled data set

This chapter presents the results obtained using DS1A and DS1B. The characteristics of these data sets can be found in table 3.1.

### 4.1 Training and test set ground truth maps

In this section the exploration maps for both the training and test set are shown. These maps act also as Ground Truth (GT) for the classification part of the work. In particular these maps show how the different classes are distributed in the input space of the simulation.

Figure 4.1 shows the ground truth maps for the training and test sets in the case of NF data, using the DS1A data set. It is possible to see that each categorization rules contains different number of classes, having less in C1 and more in C3 as expected since the latter categorization rule is more general. By looking at figure 4.1a one can see that the majority of the numerically failed simulations are found in correspondence of large values of IC flow rate reductions ( $\Delta IC$  flow rate  $> 70\%$  are present). A lower, but still relevant, amount of numerically failed simulations but still relevant can be observed for  $\Delta FC$  flow rate  $> 80\%$ .

Figure 4.1c shows the training set simulations classified according to C2 rules. This classification allows to see how, in most of the simulations that failed, numerically or physically, the most frequent first failure is  $LT_{IC}$ . Only a few  $HT_{IC}$  failures are present, previously labelled, with C1 rules, as Numerical Failure. There is only one instance in which  $HT_{FC}$  is the first cause of failure.

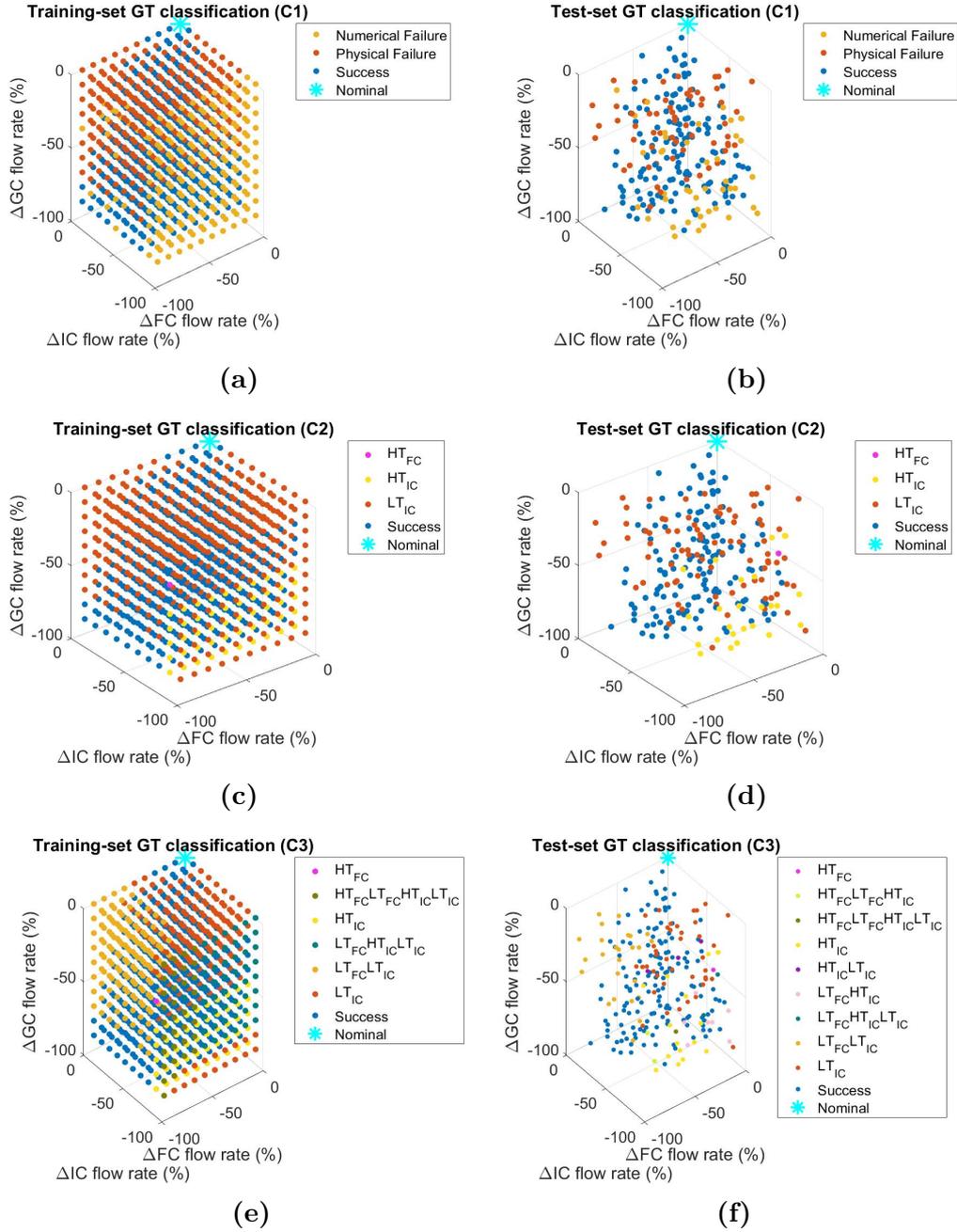
Figure 4.1e shows the most general classification used in the work (C3). In this case, the labels of the classes related to failures are composed by the sequence of all the failures that happen during the transient. Many instances of multi-failure

simulations are visible. Moreover, for  $\Delta IC$  flow rate  $> 70\%$ , most of the simulations in which the first failure was of  $LT_{IC}$  happened to have a second failure of low temperature in the FC, hence according to C3 rules the  $LT_{FC}LT_{IC}$  class was associated.

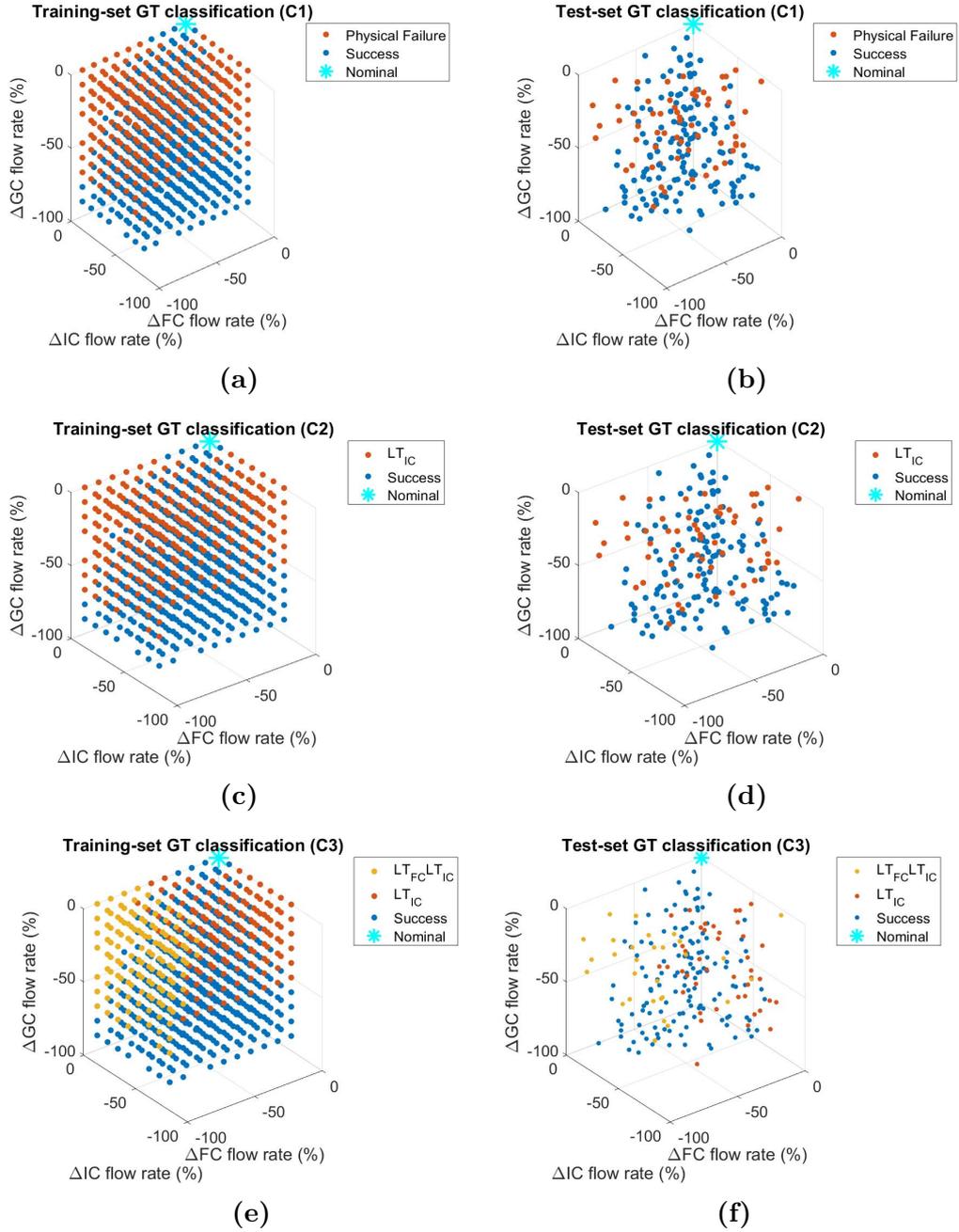
The ground truth maps for the training and test sets in the case of NF free data, i.e. the DS1B data set, are presented in figure 4.2. It is evident that the possible number of classes for this data set is smaller, especially using C3 rules, as most of the heterogeneity is contained in the NF data which are excluded.

By looking at either the group of figures of 4.1 or 4.2 it can be seen that the "safe zone" of operation, denoted by the blue dots, tends to be strongly influenced by the values of the GC flow rate reductions. In particular, for values of GC flow rate near the nominal values, there is a smaller area of safe operation with respect to larger GC flow rate reductions. From the inspection of the maps the following comments can be drawn:

- large reductions in the IC flow rate are the major cause for Numerical Failure accidents, with a minor cause to be imputed to reductions in the FC flow rate;
- large reductions of the GC flow rate are actually beneficial for maintaining the system in a successful state.



**Figure 4.1:** Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data which contains NF simulations (DS1A).



**Figure 4.2:** Ground Truth maps of the training set (left figures) and test set (right figures) according to the different training classification rules using data which is NF free (DS1B).

## 4.2 Classifiers construction

In this section some comprehensive tables regarding the classifiers are shown. In particular table 4.1 summarizes the relevant outcomes of the feature reduction step. This step was performed with a request on the Percentage of Variance Explained (PVE) of 99.999% for each controlled variable matrix. The request of an high value for the PVE is justified by the fact that there is no noise in the signals and their evolution is quite smooth. The table reports: the number of features selected (where  $N_f^1, N_f^2, N_f^3, N_f^4$  are the number of features extracted from  $T_{FC,in}$ ,  $T_{FC,out}$ ,  $T_{IC,in}$ ,  $T_{IC,out}$  respectively), the total number of features used in the analysis ( $N_f^{TOT}$ ), and the Feature Reduction (FR) percentage with respect to using the concatenation of the full signals as an example (3224 features).

PVE = 99.999 %						
Type of data set	$N_f^1$	$N_f^2$	$N_f^3$	$N_f^4$	$N_f^{TOT}$	FR (%)
DS1A	18	20	9	11	58	98.2
DS1B	6	6	6	6	24	99.3

**Table 4.1:** Feature reduction using DS1

Tables 4.2 and 4.3 presents a summary, for DS1A and DS1B data sets respectively, of the hyperparameters selected and of the performances of the different classifiers. Depending on the type of normalization procedure and classification rule selected, different classifiers are built using different hyperparameters and showing different performances.

As expected the performances on test data are better when trying to classify NF-free data. The overall, accuracy on test data is between 69% and 94% showing that the procedure is promising despite the need for some more data may improve the quality of the classification.

DS1A				
Pre-processing	Hyperparameters			Metrics
Normalization	CR	k	Distance	Accuracy on test (%)
none	C1	1	minkowski	83.2
none	C2	5	euclidean	73.2
none	C3	3	euclidean	78
once	C1	1	euclidean	82.4
once	C2	1	euclidean	74.8
once	C3	5	chebychev	78.8
twice	C1	1	minkowski	82.8
twice	C2	3	chebychev	73.6
twice	C3	1	correlation	78.8

**Table 4.2:** Hyperparameters selections and accuracy on test of different classifiers using DS1A

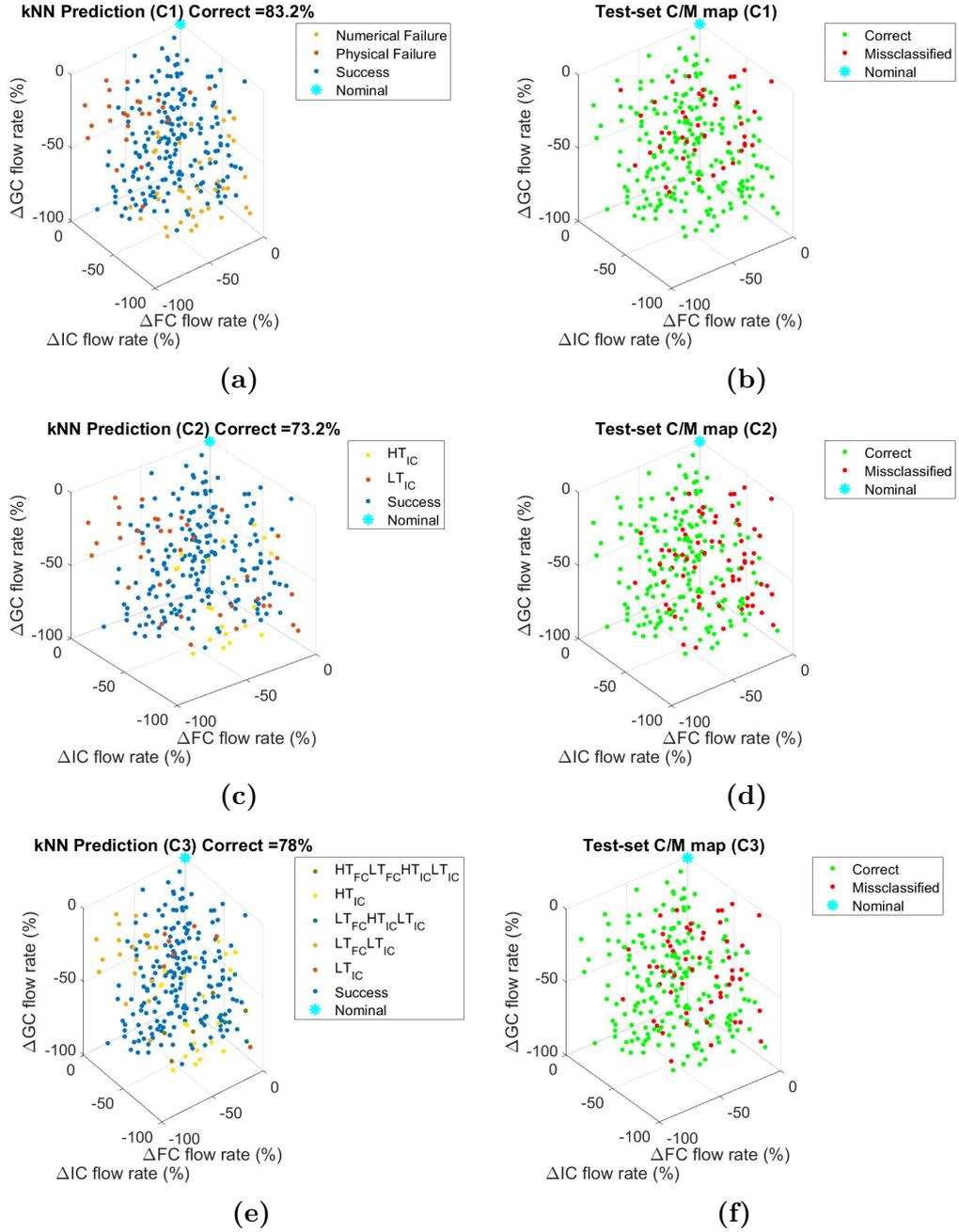
DS1B				
Pre-processing	Hyperparameters			Metrics
Normalization	CR	k	Distance	Accuracy on test (%)
none	C1	19	chebychev	94.2
none	C2	20	chebychev	93.7
none	C3	21	chebychev	87.9
once	C1	5	minkowski	83.1
once	C2	11	chebychev	87.9
once	C3	6	euclidean	72.9
twice	C1	17	chebychev	87.9
twice	C2	22	chebychev	87.4
twice	C3	11	euclidean	69.6

**Table 4.3:** Hyperparameters selections and accuracy on test of different classifiers using DS1B

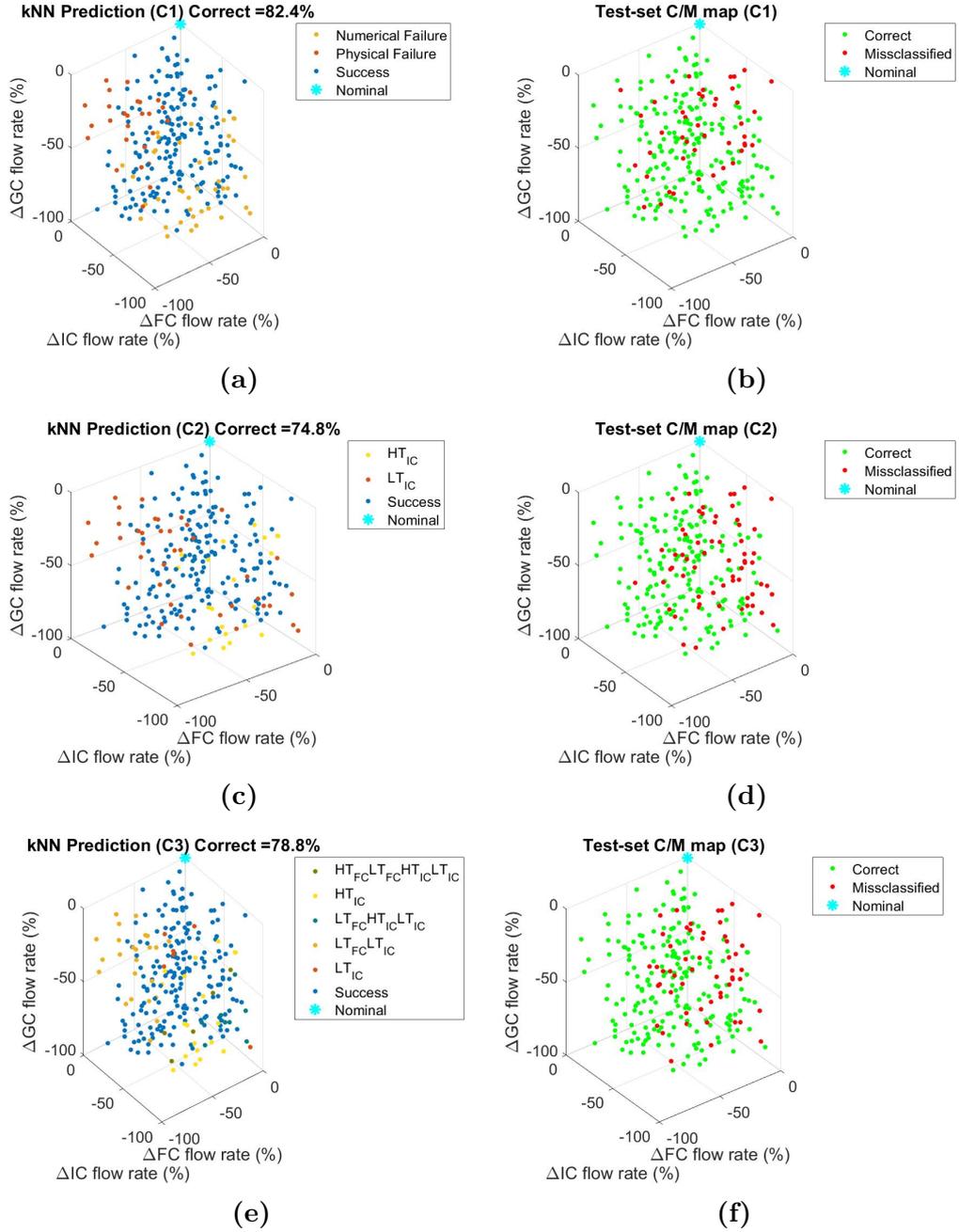
## 4.3 Prediction Maps

This section provides a more visual support for the interpretation of the performances of the different classifiers. The first three figures (4.3, 4.4 and 4.5) refer to DS1A, while the last three (4.6, 4.7 and 4.8) refer to DS1B. All the figures present prediction maps (the left column figures) and "correct" or misclassified maps (the right column figures) for the following cases: not normalized, normalized once before SVD, normalized twice before and after SVD, in order of appearance of each DS.

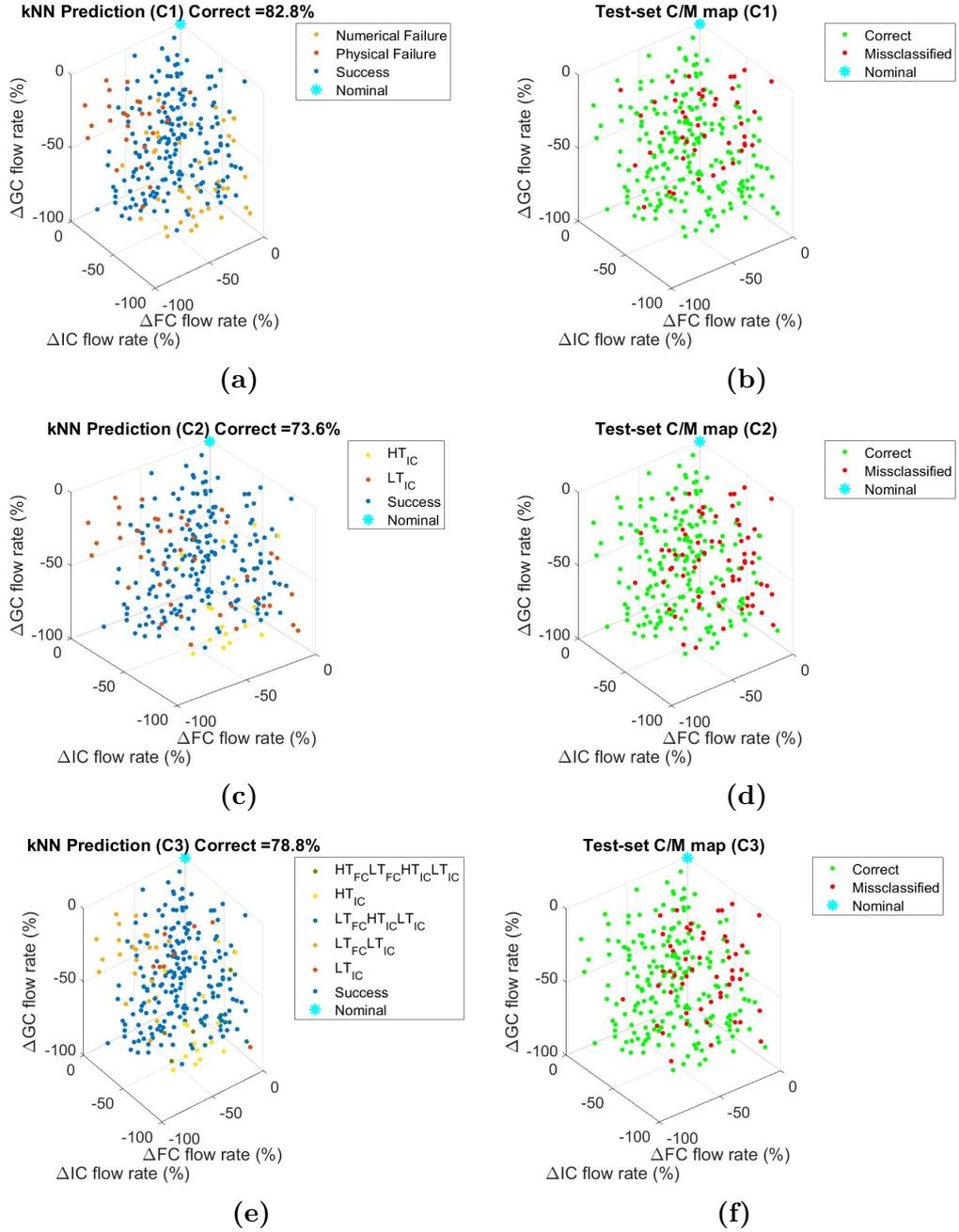
The results of the classifiers are overall good and allow, with the support of new data, to further investigate the problem. These new results are more deeply investigated and presented in the next chapter.



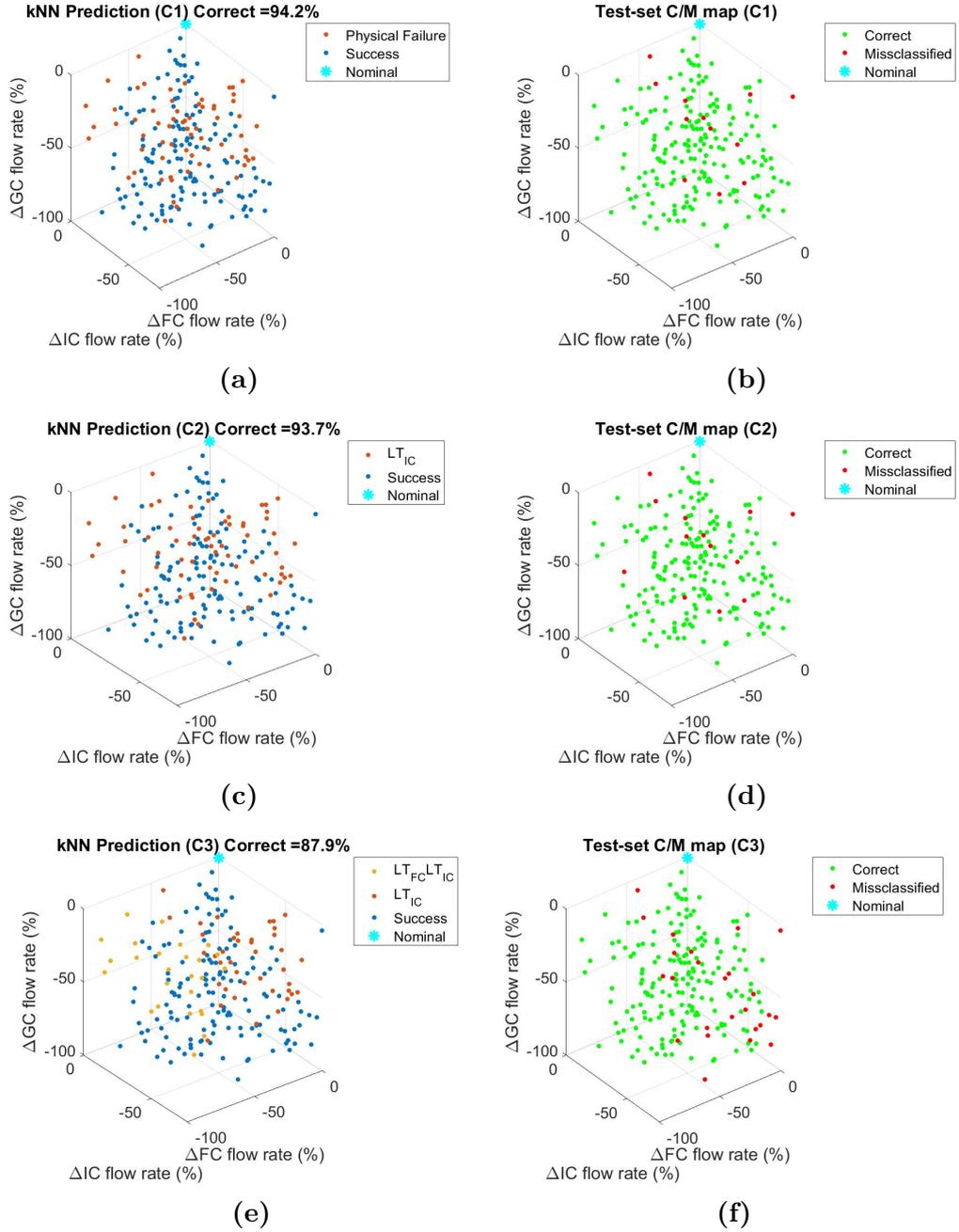
**Figure 4.3:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is not normalized.



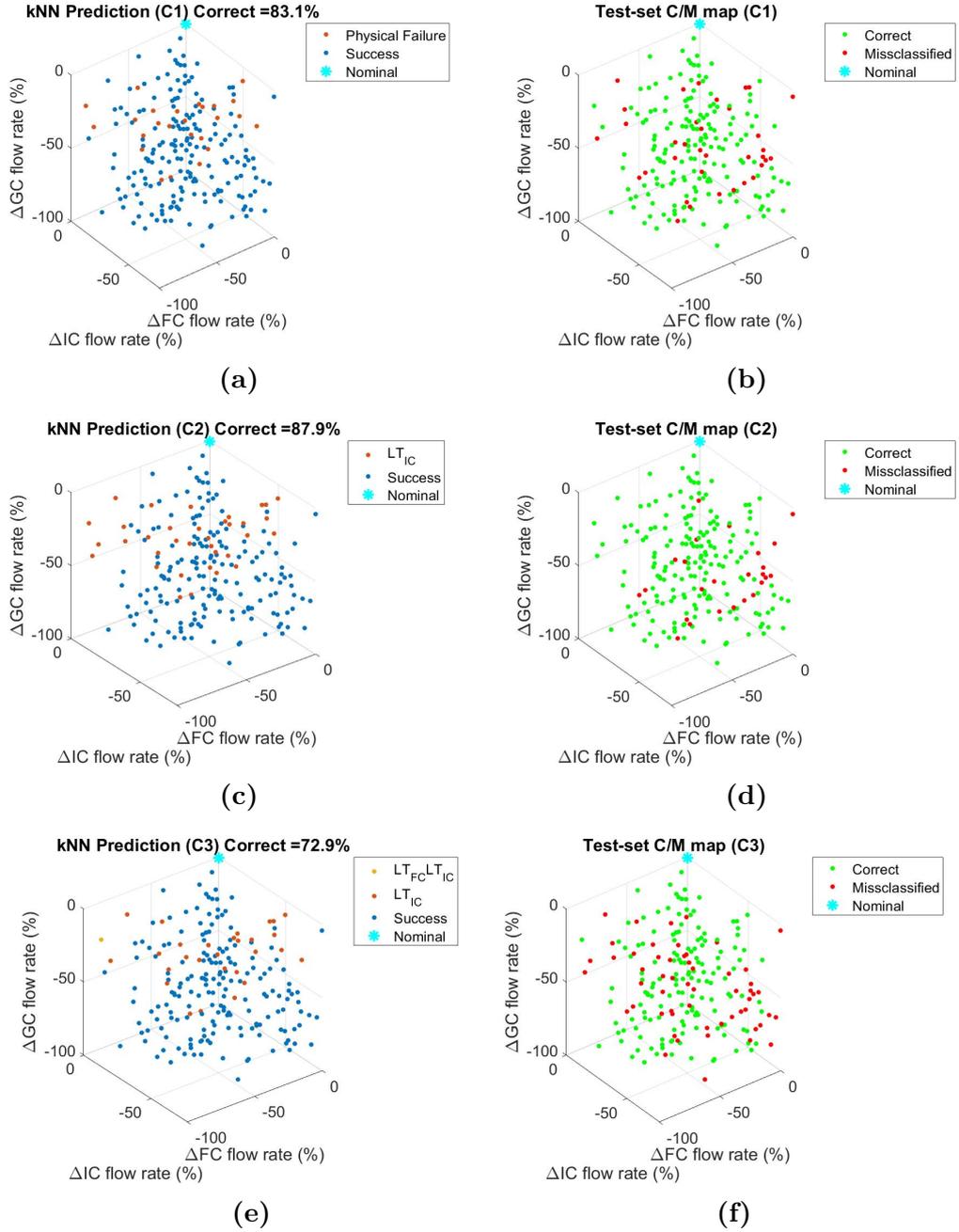
**Figure 4.4:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is normalized once before SVD.



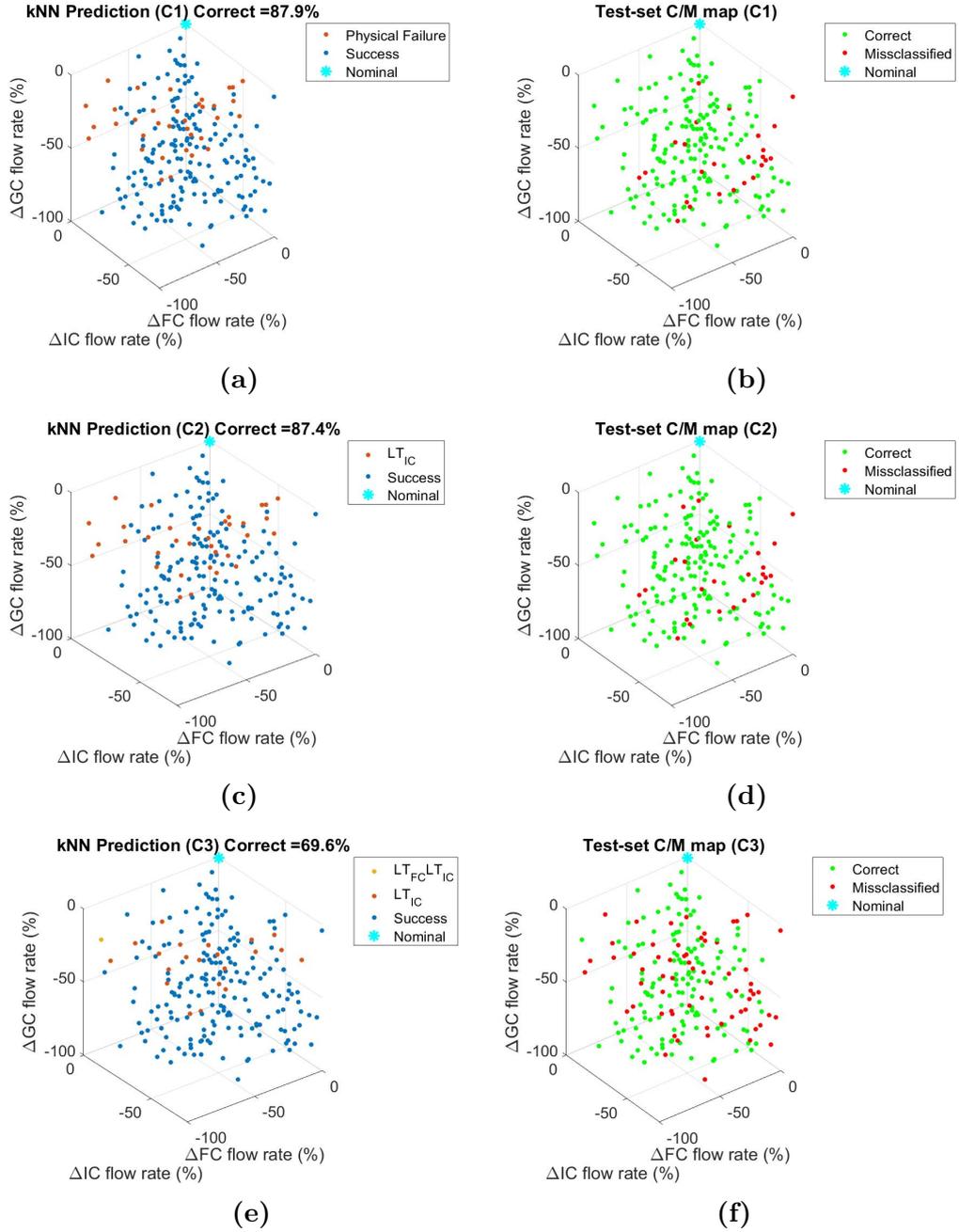
**Figure 4.5:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1A). The Feature Matrix is normalized twice before and after SVD.



**Figure 4.6:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is not normalized.



**Figure 4.7:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is normalized once before SVD.



**Figure 4.8:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS1B). The Feature Matrix is normalized twice before and after SVD.

# Chapter 5

## Results on fine sampled data set

This chapter presents the results obtained using DS2A and DS2B. The characteristics of these data sets can be found in table 3.1. This chapter continues the analysis presented in the previous chapter. In the first part some of the already shown results are integrated by means of more data. In the second part newly post-processed results are provided.

### 5.1 Training and test set ground truth maps

In this section, the exploration maps for both the training and test set are shown. These maps act also as Ground Truth (GT) for the classification part of the work. In particular these maps show in the input space of the simulation how the different classes are distributed.

Figure 5.1 shows the ground truth maps for the training and test sets in the case of NF data, which i.e. the DS2A data set. It is possible to see that each categorization rules contains different number of classes, having less classes in C1 and more in C3 as expected since the latter categorization rule is more general.

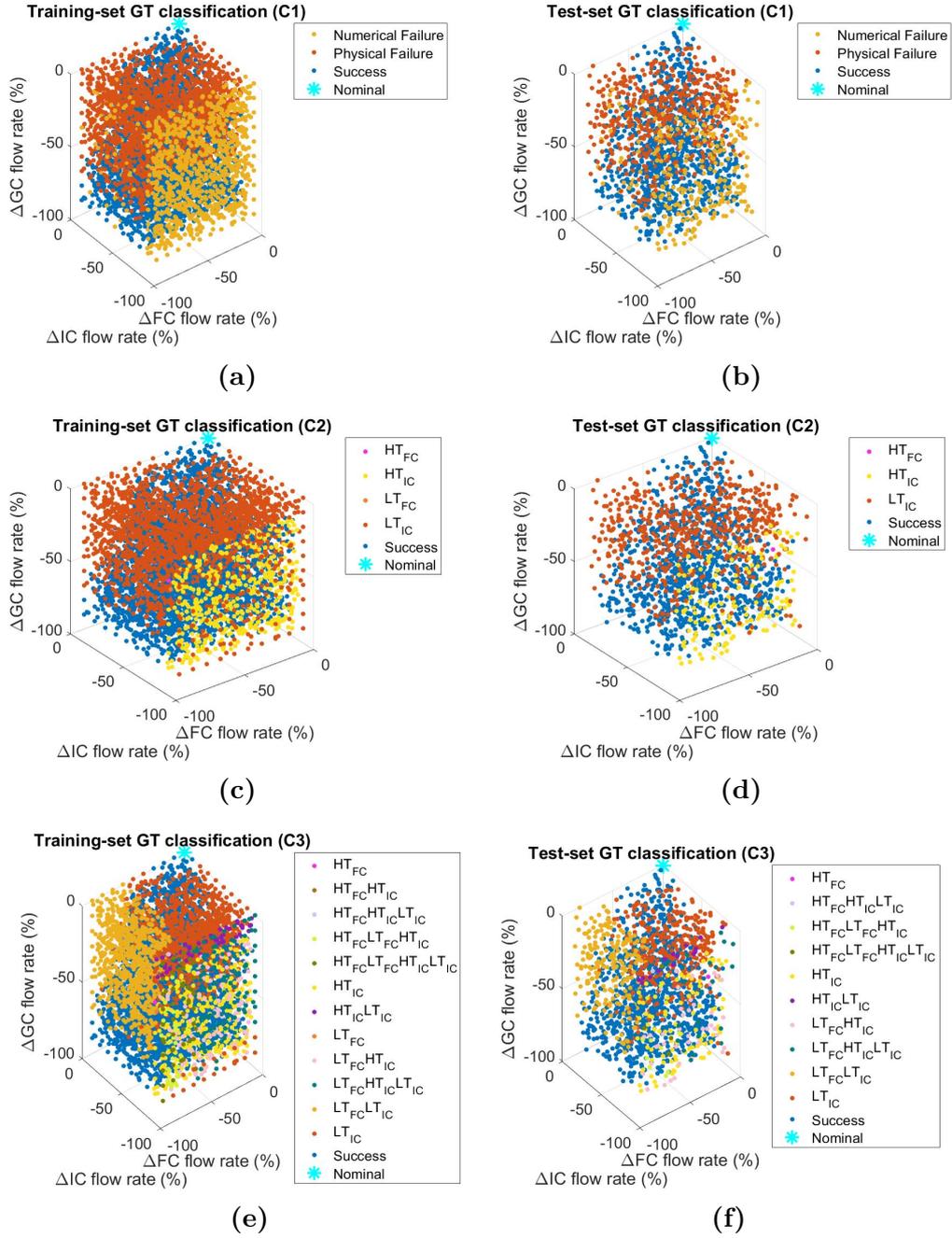
One difference with respect to the previous chapter is visible in figure 5.1c. The figure shows the training set classified according to C2 rules. In this case, most of the numerically failed simulations (according to C1 5.1a) are those in which the first failure to happen is because of high temperature in the IC ( $HT_{IC}$ ). This is a difference with respect to the same figure of the previous chapter, in which most of the numerically failed simulations happened to be classified using C2 rules as ( $LT_{IC}$ ). This can be justified by the fact that in the previous chapter, the DS1 data set was used. This data set contains an insufficient number of simulations for correctly mapping the details of these classes distributions. In this chapter, having

used a larger number of simulations, a better map of the model of the MSFR can be appreciated.

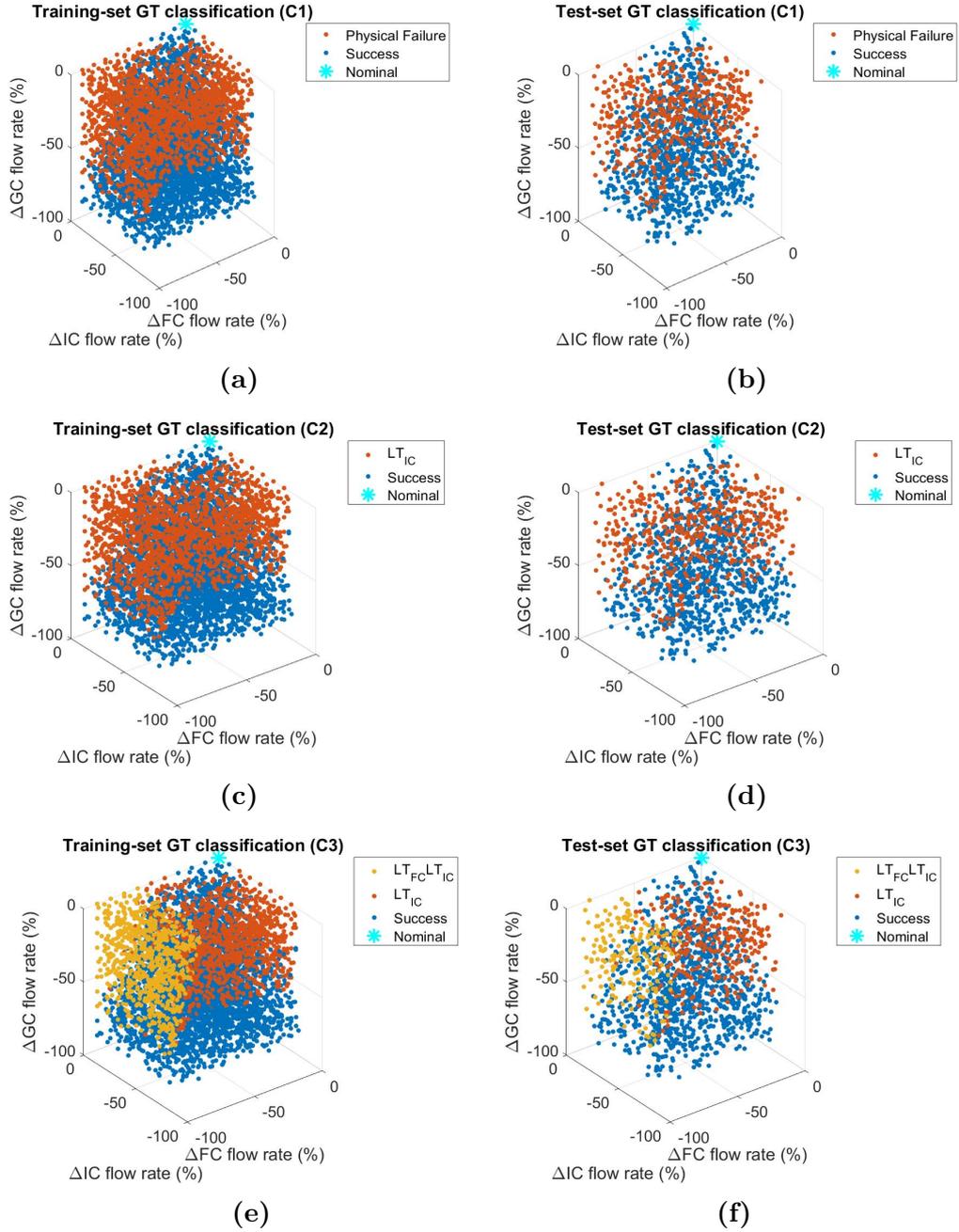
Figure 5.2 shows the ground truth maps for the training and test sets in the case of NF-free data, i.e. the DS2B data set. Also in this case, going from NF presence to NF-free data, shows a reduction in the number of possible classes in all the categorization rules. For example, concerning figure 5.2c one can see that all the accidents classes are only of the type "low temperature in the IC" ( $LT_{IC}$ ).

Figure 5.2e shows that for  $\Delta FC$  flow rate  $> 70\%$ , those simulations which physically failed due to ( $LT_{IC}$ ) at first, evolved and reached a low temperature failure in the FC also.

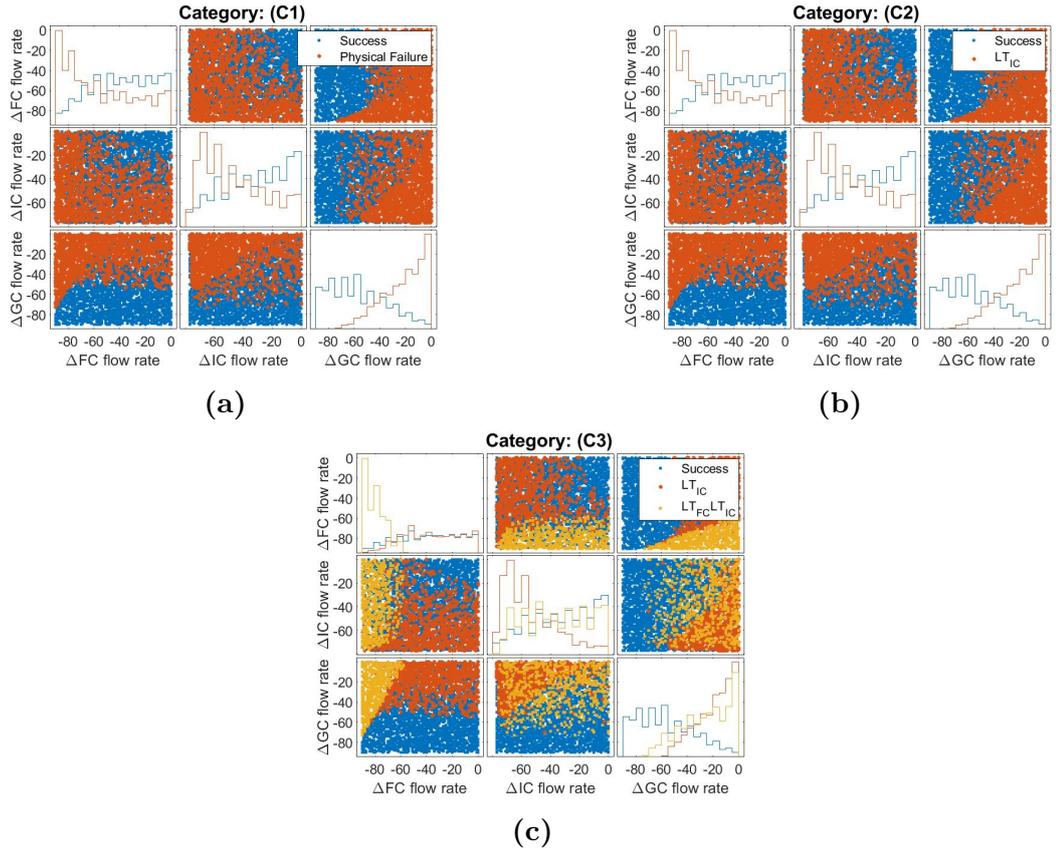
Figure 5.3 shows the Scatter Plot Matrix of the inputs of the simulations of the training set, for each categorization rule, using data without NF simulations. Figure 5.3a describes the impact of the different inputs on the safety operation area. Figure 5.3b delineates how the generic 'physical failure' is being fully represented by the ( $LT_{IC}$ ) class. Figure 5.3c characterizes in more detail the generic 'physical failure' class, showing how the ( $LT_{FC}LT_{IC}$ ) class is independent on the intensity variation of the IC flow rate. Overall, the presence of more data supports the two observations made in the previous chapter: the GC mass flow rate strongly influences the safety operation area; the IC and FC mass flow rate influence the Numerical Failure class.



**Figure 5.1:** Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data, which contains NF simulations (DS2A).



**Figure 5.2:** Ground Truth maps of the training set (left figures) and test set (right figures) according to the different classification rules using data, which is NF free (DS2B).



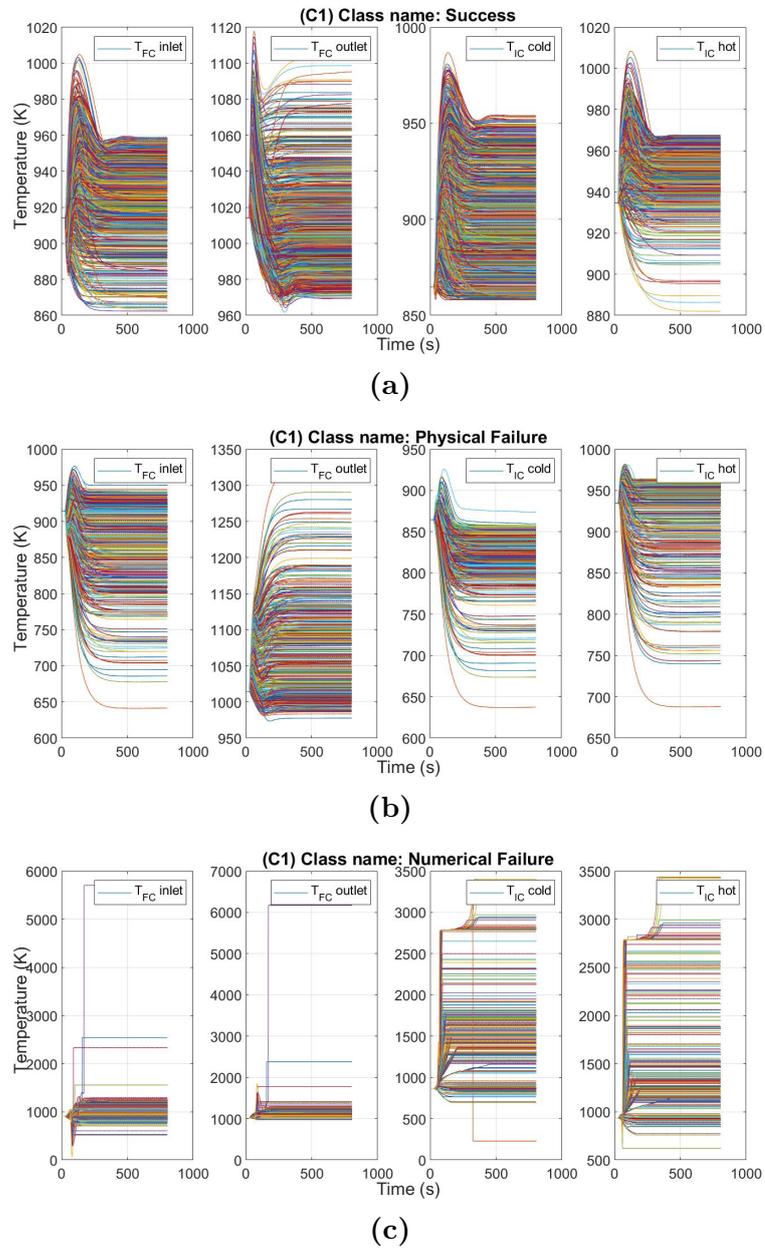
**Figure 5.3:** Scatter Plot Matrix of the inputs of the training set according to the different classification rules using data, which is NF-free (DS2B).

## 5.2 Test set transients

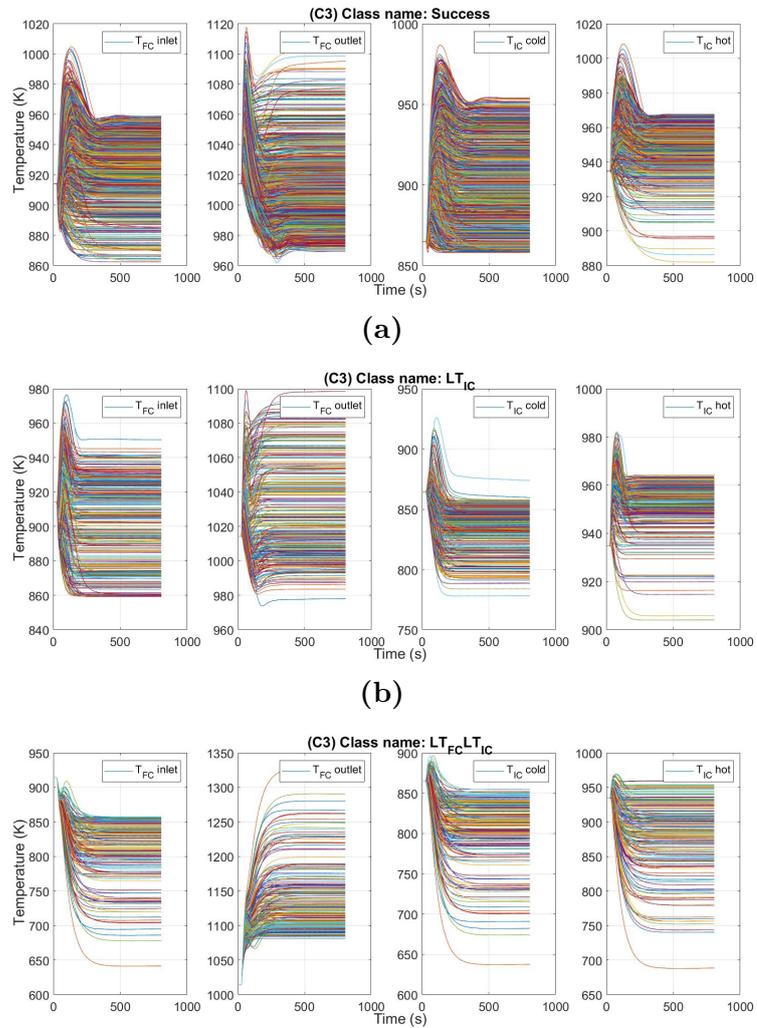
In this section the transient evolution of some simulations are shown. In particular, figure 5.4 shows the transients of the four controlled variables using simulations from the test set containing NF data. The figure contains three sub-figures, one for each class available with C1 rules. The transients are plotted accordingly.

Figure 5.4a shows the transients for those simulations classified as successful. Coherently the physical values of these variables never exceeds the threshold limits for low or high temperature failures. Figure 5.4c is another way of appreciating the heterogeneity of the class of numerically failed simulations, in which some transients reach very high temperatures while others are featured by very low temperatures. One common feature among all these signals is the sharp slope developed during the transient evolution, often caused by the sharp reduction in the circulating pumps flow rates.

Figure 5.5 shows the transients of the four controlled variables using simulations from the test set which do not contain NF simulations. The figure contains three sub-figures, one for each class available within C3 rules. The transients are represented accordingly.



**Figure 5.4:** Transient evolution of the controlled variables, using C1 rules and data with NF presence.



(c) Transient evolution of the controlled variables, using C3 rules and NF-free data .

Figure 5.5

### 5.3 Classifiers construction

In this section, some comprehensive tables regarding the classifiers are shown. In particular, table 5.1 summarizes the relevant outcomes of the feature reduction step. This step was performed with a request on the Percentage of Variance Explained (PVE) of 99.999% for each controlled variable matrix. The table reports: the number of features selected (where  $N_f^1, N_f^2, N_f^3, N_f^4$  are the number of features extracted from  $T_{FC,in}$ ,  $T_{FC,out}$ ,  $T_{IC,in}$ ,  $T_{IC,out}$ , respectively), the total number of features used in the analysis ( $N_f^{TOT}$ ), and the Feature Reduction (FR) percentage with respect to using the full concatenation of the signals as an example (3224 features).

PVE = 99.999 %						
Type of data set	$N_f^1$	$N_f^2$	$N_f^3$	$N_f^4$	$N_f^{TOT}$	FR (%)
DS2A	25	1	19	19	64	92.1
DS2B	7	7	7	7	28	96.5

**Table 5.1:** Feature reduction using DS2

Tables 5.2 and 5.3 presents a summary of the hyperparameters selected and of the performances on test data of the different classifier, for DS1A and DS1B data sets respectively. Depending on the type of normalization procedure and classification rule selected, different classifiers are built using different hyperparameters, showing different performances.

Accuracy on the test data is always above 89%, and goes up to 99% in certain cases. The only exception occurs when a classifier is built with DS2A and the data are not normalized, resulting in very poor performances. The reason behind this could be related to the fact that NF data are quite heterogeneous thus requiring mandatory normalization for training and classification purposes.

DS2A				
Pre-processing	Hyperparameters			Metrics
Normalization	CR	k	Distance	Accuracy on test (%)
none	C1	5	cosine	41.7
none	C2	3	cosine	32.9
none	C3	1	correlation	13.2
once	C1	5	euclidean	90.8
once	C2	1	euclidean	92.1
once	C3	1	euclidean	89.4
twice	C1	3	cosine	89.9
twice	C2	1	minkowski	92.1
twice	C3	1	euclidean	89.5

**Table 5.2:** Hyperparameters selections and accuracy on test of different classifiers using DS2A

DS2B				
Pre-processing	Hyperparameters			Metrics
Normalization	CR	k	Distance	Accuracy on test (%)
none	C1	7	euclidean	98.9
none	C2	8	minkowski	99.2
none	C3	7	euclidean	98.3
once	C1	9	minkowski	98.9
once	C2	8	minkowski	99.1
once	C3	3	chebychev	97.9
twice	C1	3	correlation	98.6
twice	C2	3	cosine	98.6
twice	C3	7	euclidean	98.3

**Table 5.3:** Hyperparameters selections and accuracy on test of different classifiers using DS2B

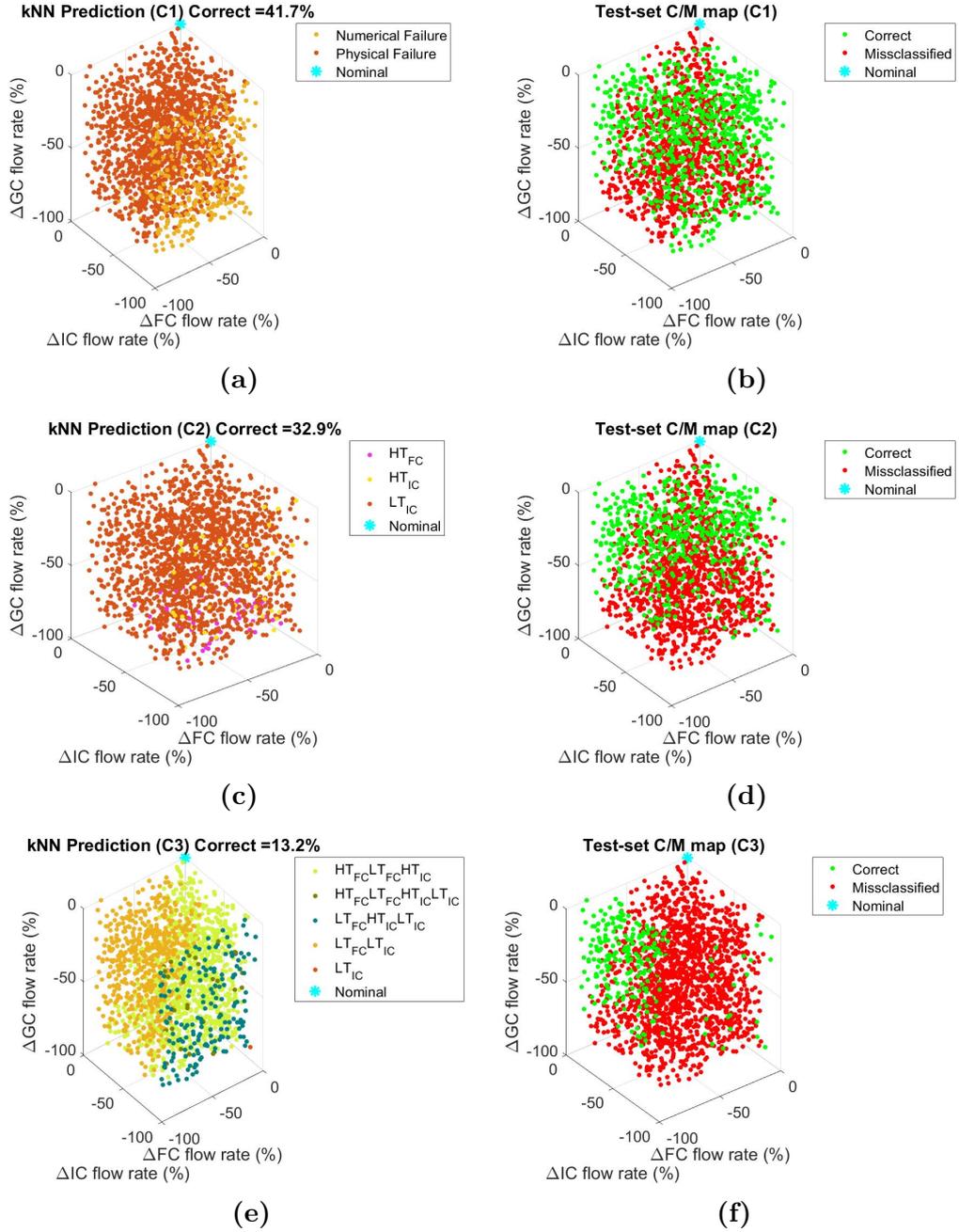
## 5.4 Prediction and correct or misclassified maps

This section provides a more visual support for the interpretation of the performances of the different classifiers. The first three figures (5.6, 5.7 and 5.8) refer to DS2A, while the last three (5.9, 5.10 and 5.11) refer to DS2B. All the figures present prediction maps (the left figures) and correct or misclassified maps using the following features: not normalized, normalized once before SVD, normalized twice before and after SVD.

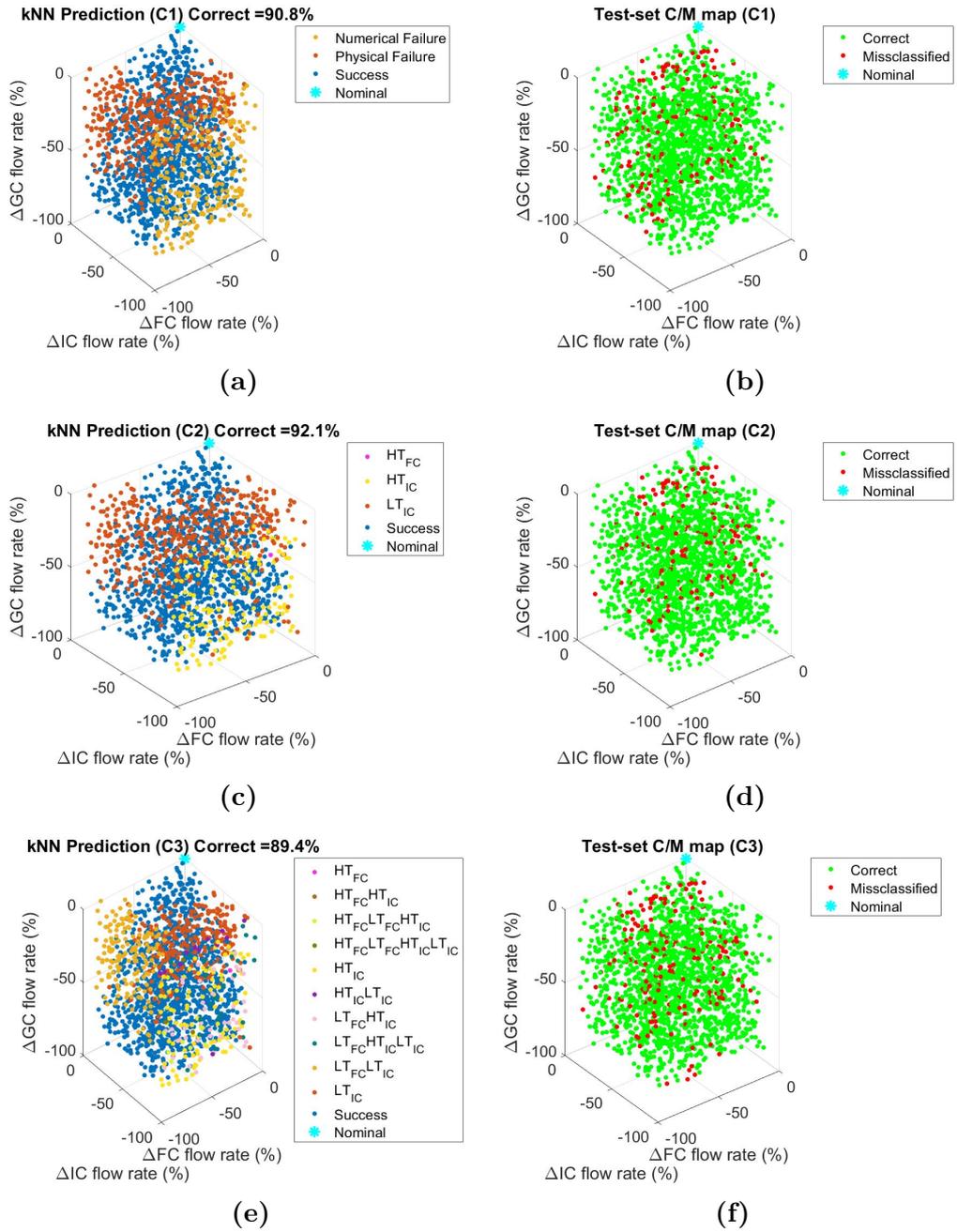
In particular, by looking at the left column of figure 5.6, one can see how the classifier model is unable to predict the class "Success", which implies a poor accuracy as a result. Looking at the right column of the same figures, one can appreciate how most of the predictions are actually wrong.

The results are much better in the case of normalized features, as shown in figures 5.7 and 5.8, in which the accuracy ranges between 89% and 92%.

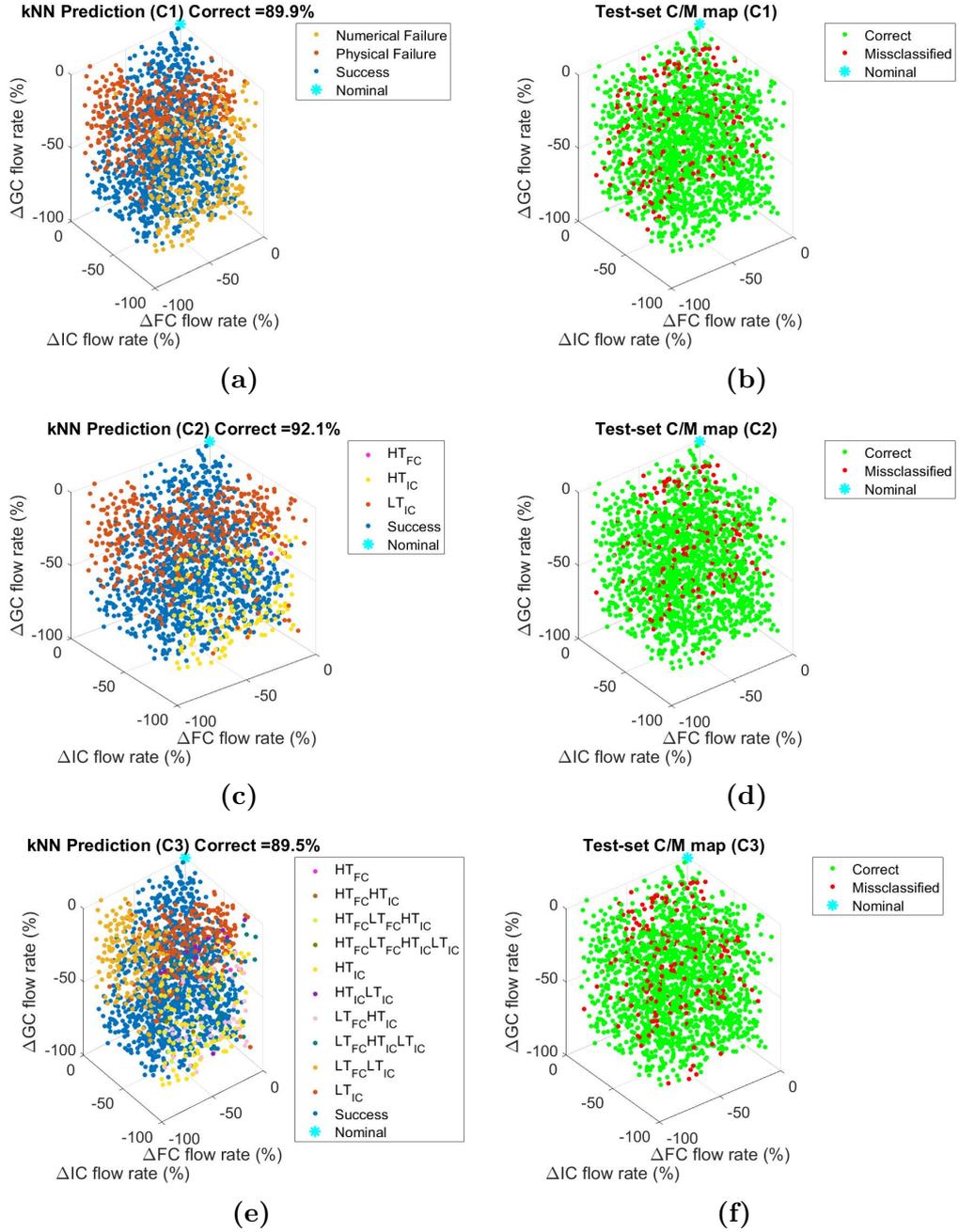
For what concerns figures related to DS2B, figures 5.9, 5.10 and 5.11, show results that are always satisfactory. This underlines how normalization is much less relevant in this application, where NF data is filtered out before. The accuracy, in this case, ranges from 97% to 99%.



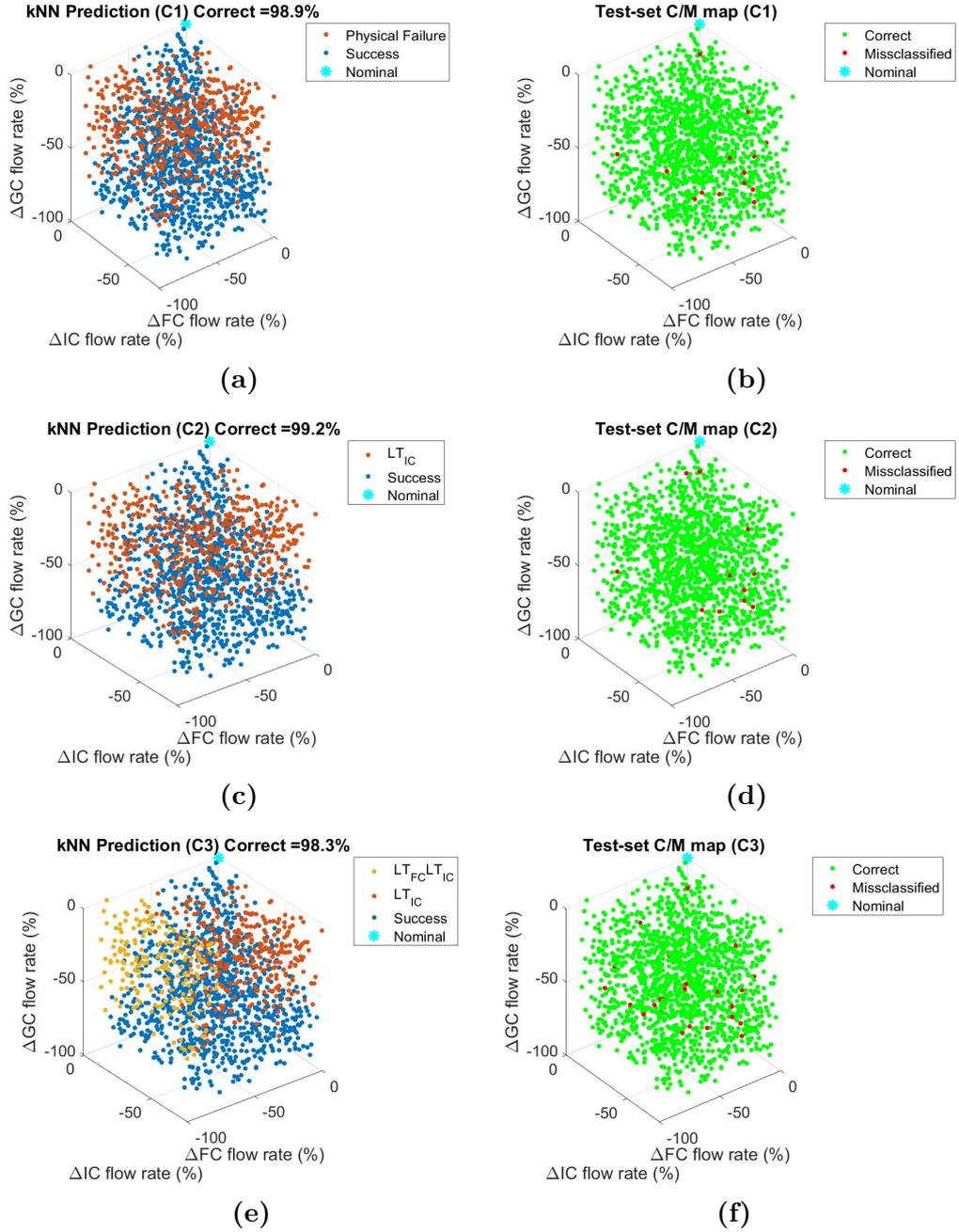
**Figure 5.6:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is not normalized.



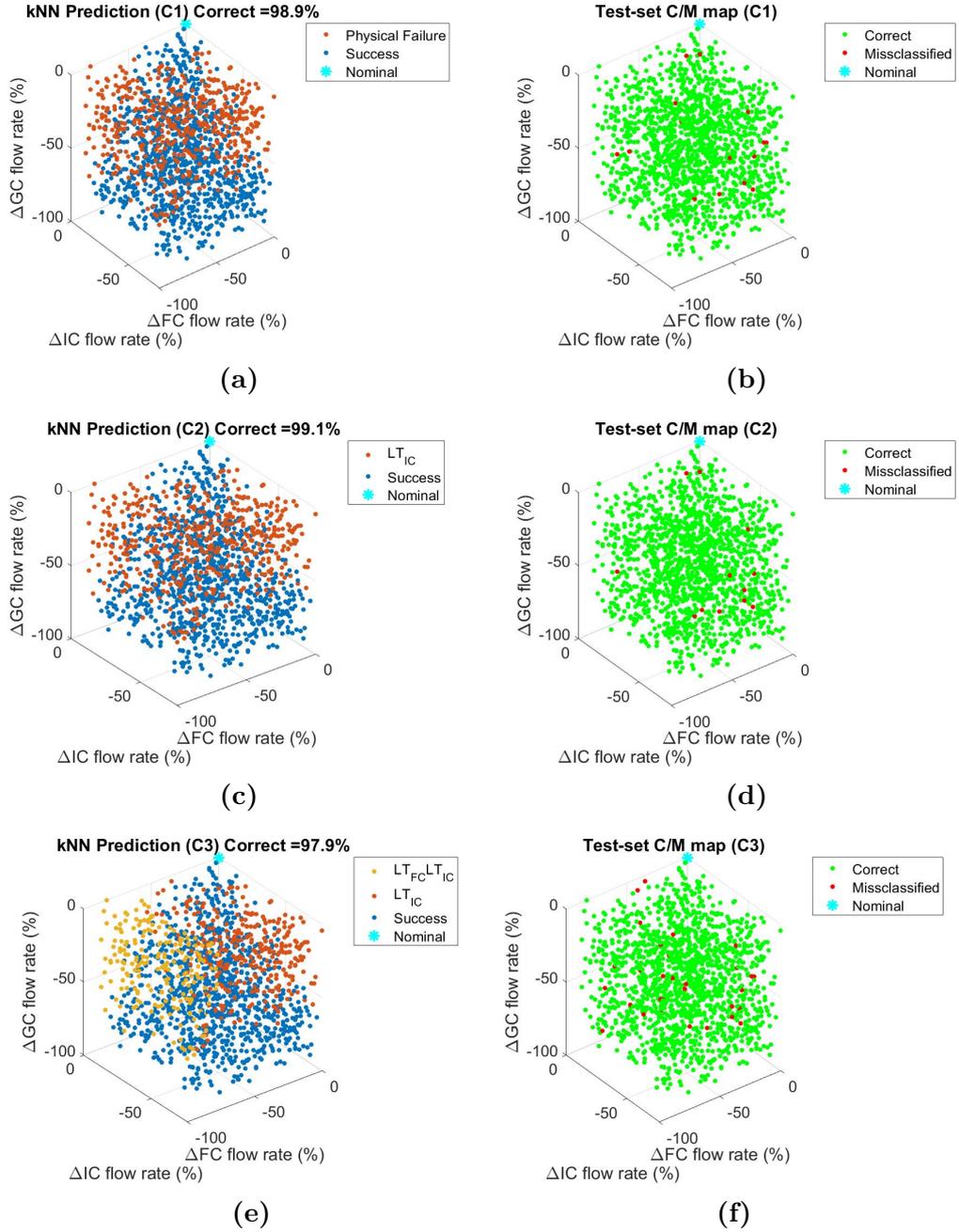
**Figure 5.7:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is normalized once before SVD.



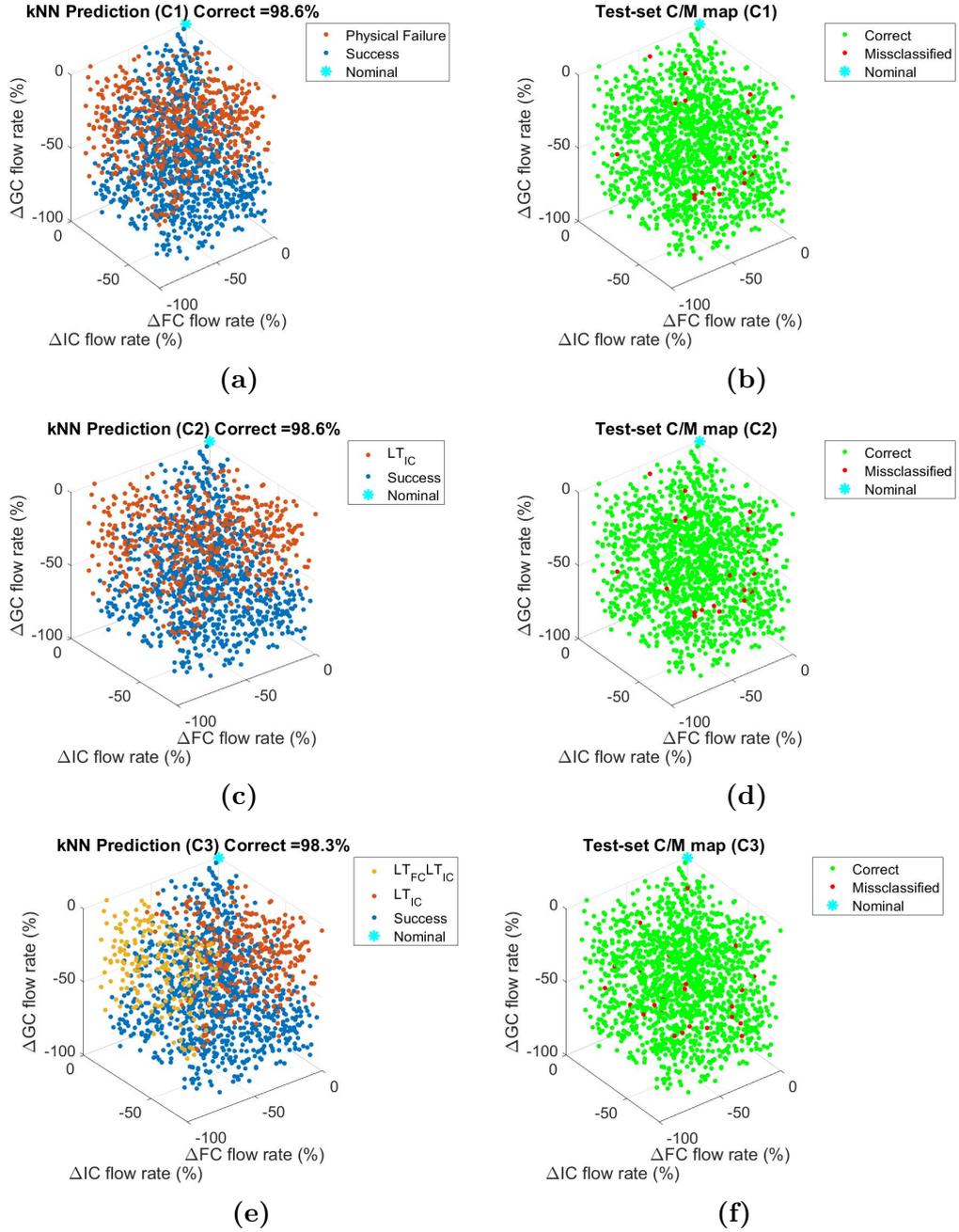
**Figure 5.8:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which contains NF simulations (DS2A). The Feature Matrix is normalized twice before and after SVD.



**Figure 5.9:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is not normalized.



**Figure 5.10:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is normalized once before SVD.



**Figure 5.11:** Model prediction maps (left figures) and correct or miss-classified maps (right figures) according to the different classification rules using data which does not contain NF simulations (DS2B). The Feature Matrix is normalized twice before and after SVD.

## 5.5 Confusion matrix charts

This section shows confusion matrix charts providing new metrics such as class-wise precision and recall. The classifier predictions can be categorized, for a given class, as follows:

- True Positive (TP): the actual class and the predicted class are the same;
- False Negative (FN): the actual class is positive and the prediction is negative;
- False Positive (FP): the actual class is negative and the prediction is positive;

Precision can be interpreted as what percentage is truly positive out of all the positive predictions and is defined as follows:

$$\frac{TP}{TP + FP} \quad (5.1)$$

Recall can be interpreted as what percentage is predicted positive out of the total positive and is defined as follows:

$$\frac{TP}{TP + FN} \quad (5.2)$$

Figures 5.12, 5.13 and 5.13 show confusion matrix charts for three cases of features used: not normalized, normalized once before SVD, normalized twice before and after SVD. In each figure, the left column refers to DS2A, while the right column to DS2B.

In particular, the left column of figure 5.12 shows the poor performances of the classifier under the perspective of new metrics like the class-wise recall and precision. On the other hand, in figures 5.13 and 5.14, one can see, on the left, the improvement on the classifier metrics when the features are normalized.

In all the three figures, the column on the right refers to DS2B. Each figure exhibits good performances in this application, independently of the normalization.







## 5.6 Inference on inputs (diagnosis on pumps failure intensities)

This section describes the last part of the work, which is of fault diagnosis. This part of the work is carried out by using the categorization rules C3 and data which belong to DS2B (numerical failure free). In particular the graphs of figure 5.15 are built on the training set data: they represent the empirical probability distributions (histograms) of the pumps failure intensities conditional on the different (output) transients classes, i.e., the distributions of the pump failure intensities that generated (i.e., that are responsible for) the plant failure behavior of a given class (e.g.,  $LT_{FC}$ ,  $LT_{FC}LT_{IC}$ ). On the other hand, the graphs of figure 5.16 are built for four points of the test set by using the PDFs of the pump failure intensities constructed on the training set and the probabilities of those test-examples to belong in the different classes, provided by the trained classifier (see section 3.1 for technical details).

In more detail, the left part of figure 5.15 shows some histograms that have on the horizontal axis different failure intensities of the recirculating pumps flow rates (one subplot for each of the three circuits), and as common vertical axes the number of observations in that class. The right column describes the normalized version of these histograms which are probability density functions.

The conditional distributions on the right column of figure 5.15 are weighed by the above mentioned assignment probabilities (Theorem of Total Probability) as follows:

$$f(\Delta\dot{m}_i) = \sum_j^{nc} f(\Delta\dot{m}_i|C_j) \cdot Pr(\tau(\Delta\dot{m}_i) \in C_j) \quad \text{where } i \in \{FC, IC, GC\} \quad (5.3)$$

where

- $nc$  is the total number of classes;
- $f(\Delta\dot{m}_i|C_j)$  is the  $i$ -th conditional (on class  $C_j$ ) distribution (depicted on the right column of figure 5.15);
- $Pr(\tau(\Delta\dot{m}_i) \in C_j)$  indicates the assignment probability provided by the classifier (the probability that transient  $\tau$ , caused by the pump failure  $\Delta\dot{m}_i$ , belongs in the  $j$ -th class ( $C_j$ )).

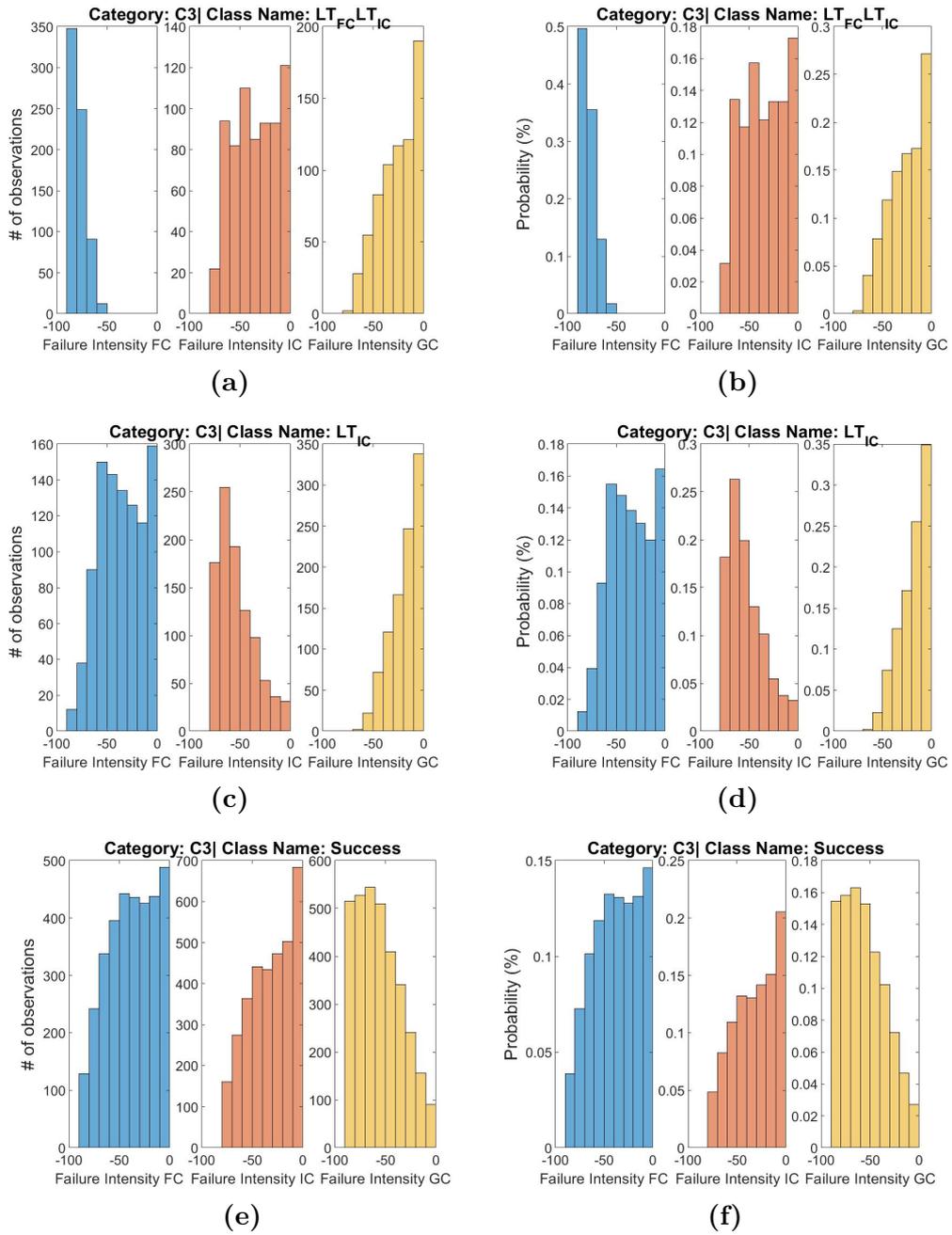
The unconditional probability distributions of the pump failure intensities, are obtained for each transient: these can be used to retrieve *a posteriori* the most likely root causes of the observed (output) system failure behaviour.

Figure 5.16 provides the inference results for four different test example points. The four test transients ( $\tau_1, \tau_2, \tau_3, \tau_4$ ) are generated with the following combinations of failure intensities respectively:  $-(66, 45, 23)\%$ ,  $-(80, 12, 54)\%$ ,  $-(30, 37, 19)\%$  and

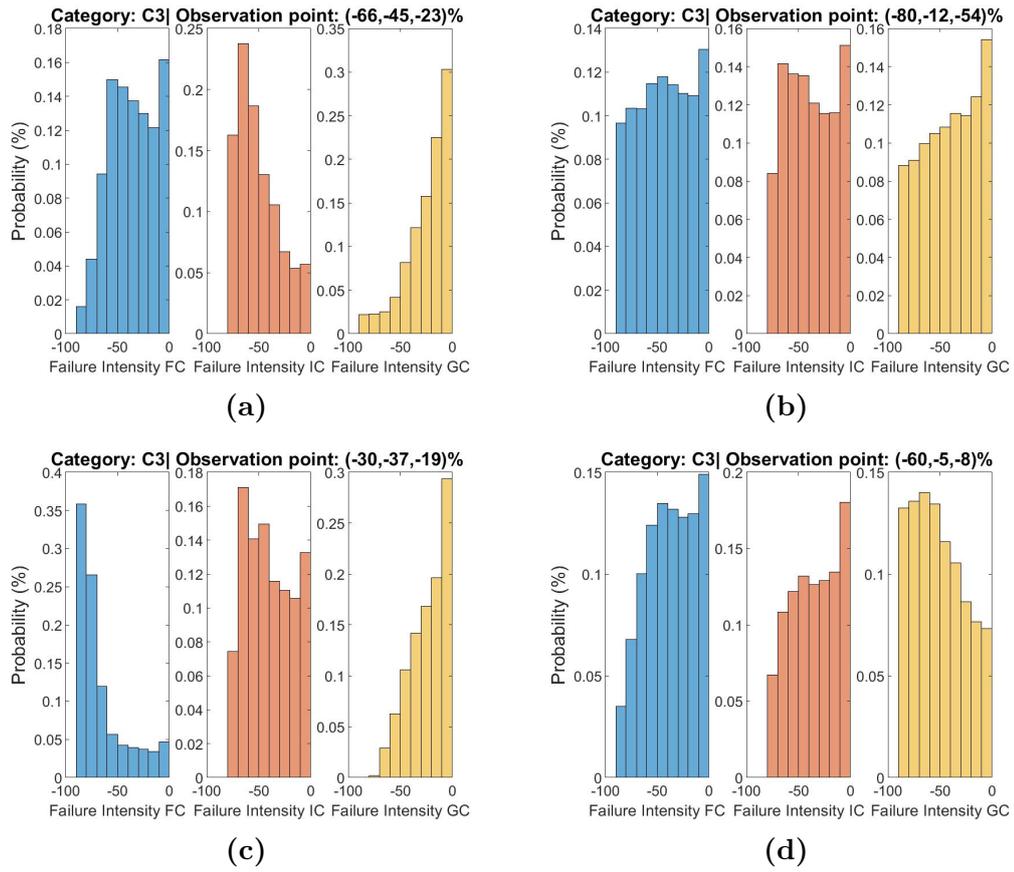
-(60, 5, 8)%. The trained classifier provides the following assignment probabilities, i.e., the probabilities that each test transient belongs to the available classes:

- $[\Pr(\tau_1 \in LT_{FC}LT_{IC}), \Pr(\tau_1 \in LT_{IC}), \Pr(\tau_1 \in \text{Success})] = [0; 85.7; 14.3]\%$
- $[\Pr(\tau_2 \in LT_{FC}LT_{IC}), \Pr(\tau_2 \in LT_{IC}), \Pr(\tau_2 \in \text{Success})] = [14.3; 28.6; 57.1]\%$
- $[\Pr(\tau_3 \in LT_{FC}LT_{IC}), \Pr(\tau_3 \in LT_{IC}), \Pr(\tau_3 \in \text{Success})] = [71.4; 28.6; 0]\%$
- $[\Pr(\tau_4 \in LT_{FC}LT_{IC}), \Pr(\tau_4 \in LT_{IC}), \Pr(\tau_4 \in \text{Success})] = [0; 14.3; 85.7]\%$

These assignment probabilities are used in equation 5.3 to retrieve the unconditional probability distributions of figure 5.16. Overall, the most likely causes (inferred inputs) coincide with the real inputs for all the tested points. Nevertheless, a specific routine assessing the confidence interval of the inference should be carried out.



**Figure 5.15:** Number of observations (left figures) and conditional probability density functions (right figures) for different failure intensities typologies. Using DS2B and categorization rules of C3.



**Figure 5.16:** Probability density functions characterizing four different classified test points.

# Chapter 6

## Conclusions

In the context of the deployment of the Gen-IV Molten Salt Fast Reactor (MSFRs), there is the need for the demonstration of the reactor enhanced safety features with respect to other reactor concepts. To this aim, with the support of a power plant simulator developed in the framework of the SAMOFAR EU project, this thesis was carried out with two different objectives: 1) to propose a simulation-based method to characterize the system behaviour with respect to variations in physical and operational parameters, by means of a thorough exploration of the MSFR power plant state space; 2) to develop a data-driven algorithm for the efficient detection and classification of incidents, relying on a k-Nearest Neighbors (kNN) classifier.

For what concerns the first objective, the input parameters that strongly influence the behaviour of the MSFR power plant have been identified as: fuel circuit, intermediate circuit and gas circuit salt mass flow rates.

The output parameters, for the system status representation, have been selected in the form of the FC (core) inlet and (core) outlet temperatures and the IC cold (heat-exchanger inlet) and hot (heat-exchanger outlet) temperatures, given the constrain of choosing signals that would be measurable in a real experimental facility.

The MSFR state exploration has been carried out by random sampling several combinations of the different mass flow rates and by analyzing the corresponding simulated transient behaviour of the MSFR. This has resulted in the identification of a safe operating zone and an abnormal operating zone. Moreover by means of different detailed rules of classification a precise characterization of the abnormal states has been described, with the most common failure to be identified in the reaching of the salt freezing temperature in the intermediate circuit.

For what concerns the second objective, the k-Nearest Neighbors (kNN) method

has been selected as the data-driven algorithm due to its simplicity and interpretability. Different incident detection and classification models have been trained, using different normalization routines and classification rules, and then validated and tested with accuracy ranging between 89% and 99%, which can be considered satisfactory. Additionally, the incident detection and classification models have demonstrated strong precision performance, between 98% and 99%, which is crucial for the application's success. Specifically, in the case of fault detection, the model's precision is critical for the plant's availability. In the case of (possible) accident classification, high model precision is essential to ensure accurate identification and avoid confusion between different types of accidents.

The foundations for a Fault Diagnosis (FD) analysis (i.e., the a posteriori identification of the "input" root causes - i.e., components failures - responsible for the system anomalous behaviour) have been established and qualitatively tested on four test cases providing promising results.

## 6.1 Future developments

Different parts of the work carried out are worthy of further analysis for the purpose of improvement.

First of all, efficient exploration techniques of the MSFR state space (e.g., combinations of adaptive sampling methods and fast-running emulators and regression models) can be used if any additional data is needed. More data may be required for example to better characterize sparsely populated classes of system failure, which would result in improved training of the classifier and, therefore, in better recognition rates.

It's crucial to also address the issue of what type of data to use in the future, specifically whether to incorporate simulations that have failed numerically into the data set or not. In particular this work has shown how the classifier model is able to perform better on data which does not contain numerical failures. However the omission of this part of the data could result in worst performances in case real-experimental data had to be used. The reason for this is the relevance of such data which may provide also relevant physical information. Simply removing this data may result in a biased (and thus poorly performing) classifier, as entire failure classes could be erased and, thus, would not be available for the classifier training. Nevertheless the presence of numerical failure data is for sure a challenge for the data analysis and requires special attention.

Also, there may be the need of considering, taking into account and adding new controlled (critical) variables to be used for a more accurate and precise identification of the system (safe or failed) status (constraint in a real application, this choice could be driven by the availability - or not - of real-time measurable signals

from the experimental facility or from the power plant).

In addition, notice that in general different data-driven incident detection and classification methods can present different, peculiar characteristics (and correspondingly different performances with respect to different types of failures). In this view, another possible development of the present work could focus on the use of ensembles of methodologies of different nature and on the aggregation of the corresponding classification results, with an expected increase in the overall robustness.

Finally, based on the results produced in this work, there is a broader goal of developing a tool capable of classifying and detecting the presence of anomalous states in the system in real-time (online) conditions.

# Bibliography

- [1] International Energy Agency. *World Energy Outlook: Energy consumers of tomorrow*. Tech. rep. 2021 (cit. on p. 1).
- [2] Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. «Climate Change 2013: The Physical Science Basis.» In: *Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA*. 5.13 (2013) (cit. on p. 1).
- [3] Barry W. Brook, Agustin Alonso, Daniel A. Meneley, Jozef Misak, Tom Blee, and Jan B. van Erp. «Why nuclear energy is sustainable and has to be part of the energy mix». In: *Sustainable Materials and Technologies* 1-2 (2014), pp. 8–16. ISSN: 2214-9937. DOI: <https://doi.org/10.1016/j.susmat.2014.11.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2214993714000050> (cit. on p. 1).
- [4] Generation IV International Forum. *GIF R&D Outlook for Generation IV Nuclear Energy Systems: 2018 Update*. Tech. rep. Generation IV International Forum, 2018 (cit. on p. 1).
- [5] E. S. Bettis, R. W. Schroeder, G. A. Cristy, H. W. Savage, R. G. Affel, and L. F. Hemphill. «The Aircraft Reactor Experiment—Design and Construction». In: *Nuclear Science and Engineering* 2.6 (1957), pp. 804–825. DOI: [10.13182/NSE57-A35495](https://doi.org/10.13182/NSE57-A35495) (cit. on p. 2).
- [6] R C. Robertson. «MSRE Design & Operations Report Part 1 Description of Reactor Design». In: (Jan. 1965). DOI: [10.2172/4654707](https://doi.org/10.2172/4654707). URL: <https://www.osti.gov/biblio/4654707> (cit. on p. 2).
- [7] R C Robertson. «Conceptual design study of a single-fluid molten-salt breeder reactor.» In: (Jan. 1971). DOI: [10.2172/4030941](https://doi.org/10.2172/4030941). URL: <https://www.osti.gov/biblio/4030941> (cit. on p. 2).
- [8] Mariya Brovchenko et al. «Neutronic benchmark of the molten salt fast reactor in the frame of the EVOL and MARS collaborative projects». In: *EPJ Nuclear Sci. Technol.* 5 (2019), pp. 8–16. DOI: <https://doi.org/10.1051/epjn/2018052> (cit. on p. 2).

- 
- [9] SAMOSAFAER. *The SAMOSAFAER website*. URL: <https://samosafer.eu/project/concept/> (cit. on pp. 3, 4, 7).
- [10] Claudio Tripodo, Andrea Di Ronco, Stefano Lorenzi, and Antonio Cammi. «Development of a control-oriented power plant simulator for the molten salt fast reactor». In: *EPJ Nuclear Sci. Technol.* 5.13 (2019) (cit. on pp. 4, 5, 9, 20).
- [11] The Modelica Association. *Modelica*. Version 3.2.2. 2014. URL: <https://modelica.org/> (cit. on p. 4).
- [12] Delphine Gérardin, Anna Chiara Uggenti, Stéphane Beils, Andrea Carpignano, Sandra Dulla, Elsa Merle, Daniel Heuer, Axel Laureau, and Michel Allibert. «A methodology for the identification of the postulated initiating events of the Molten Salt Fast Reactor». In: *Nuclear Engineering and Technology* 51.4 (2019), pp. 1024–1031. DOI: 10.1016/j.net.2019.01.009 (cit. on p. 6).
- [13] Daniel Heuer, Axel Laureau, Elsa Merle-Lucotte, Michel Allibert, and D. Gérardin. «A Starting Procedure for the MSFR: Approach to Criticality and Incident Analysis». In: 2017 (cit. on p. 6).
- [14] E. Cervi, S. Lorenzi, A. Cammi, and L. Luzzi. «Development of a multiphysics model for the study of fuel compressibility effects in the Molten Salt Fast Reactor». eng. In: *Chemical engineering science* 193 (2019), pp. 379–393. ISSN: 0009-2509 (cit. on p. 6).
- [15] Francesco Di Lecce, Antonio Cammi, Sandra Dulla, Stefano Lorenzi, and Piero Ravetto. «Simplified 0-D semi-analytical model for fuel draining in molten salt reactors». eng. In: *EPJ nuclear sciences 'I&' technologies* 5 (2019), pp. 14–. ISSN: 2491-9292 (cit. on p. 6).
- [16] Vidya Sagar Yellapu, Vineet Vajpayee, and Akhilanand Pati Tiwari. «Online Fault Detection and Isolation in Advanced Heavy Water Reactor Using Multiscale Principal Component Analysis». In: *IEEE Transactions on Nuclear Science* 66.7 (2019), pp. 1790–1803. DOI: 10.1109/TNS.2019.2919414 (cit. on p. 6).
- [17] Piero Baraldi, Francesco Di Maio, and Enrico Zio. «Unsupervised Clustering for Fault Diagnosis in Nuclear Power Plant Components». In: *International Journal of Computational Intelligence Systems* 6 (4 2013), pp. 764–777. ISSN: 1875-6883. DOI: 10.1080/18756891.2013.804145. URL: <https://doi.org/10.1080/18756891.2013.804145> (cit. on p. 6).
- [18] SAMOSAFAER. *Young Molten Salt Reactor conference*. URL: <https://www.nuclearenergy.polimi.it/young-molten-salt-reactor-ymsr-conference-2022/> (cit. on p. 7).

- [19] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: <https://www.mathworks.com> (cit. on p. 26).
- [20] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. Springer, 2007. ISBN: 0387310738 (cit. on p. 27).
- [21] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN: 9780262018029 0262018020 (cit. on p. 27).
- [22] The MathWorks Inc. *Predict function documentation*. Natick, Massachusetts, United States, 2022. URL: <https://it.mathworks.com/help/stats/classificationknn.predict.html#d124e767803> (cit. on p. 28).
- [23] Gilbert Strang. *Introduction to Linear Algebra*. Fourth. Wellesley, MA: Wellesley+ Cambridge Press, 2009. ISBN: 9780980232714 0980232716 9780980232721 0980232724 9788175968110 8175968117 (cit. on p. 29).
- [24] IBM. *What is the k-nearest neighbors algorithm?* URL: <https://www.ibm.com/topics/knn#:~:text=The%20k-nearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.> (cit. on p. 31).