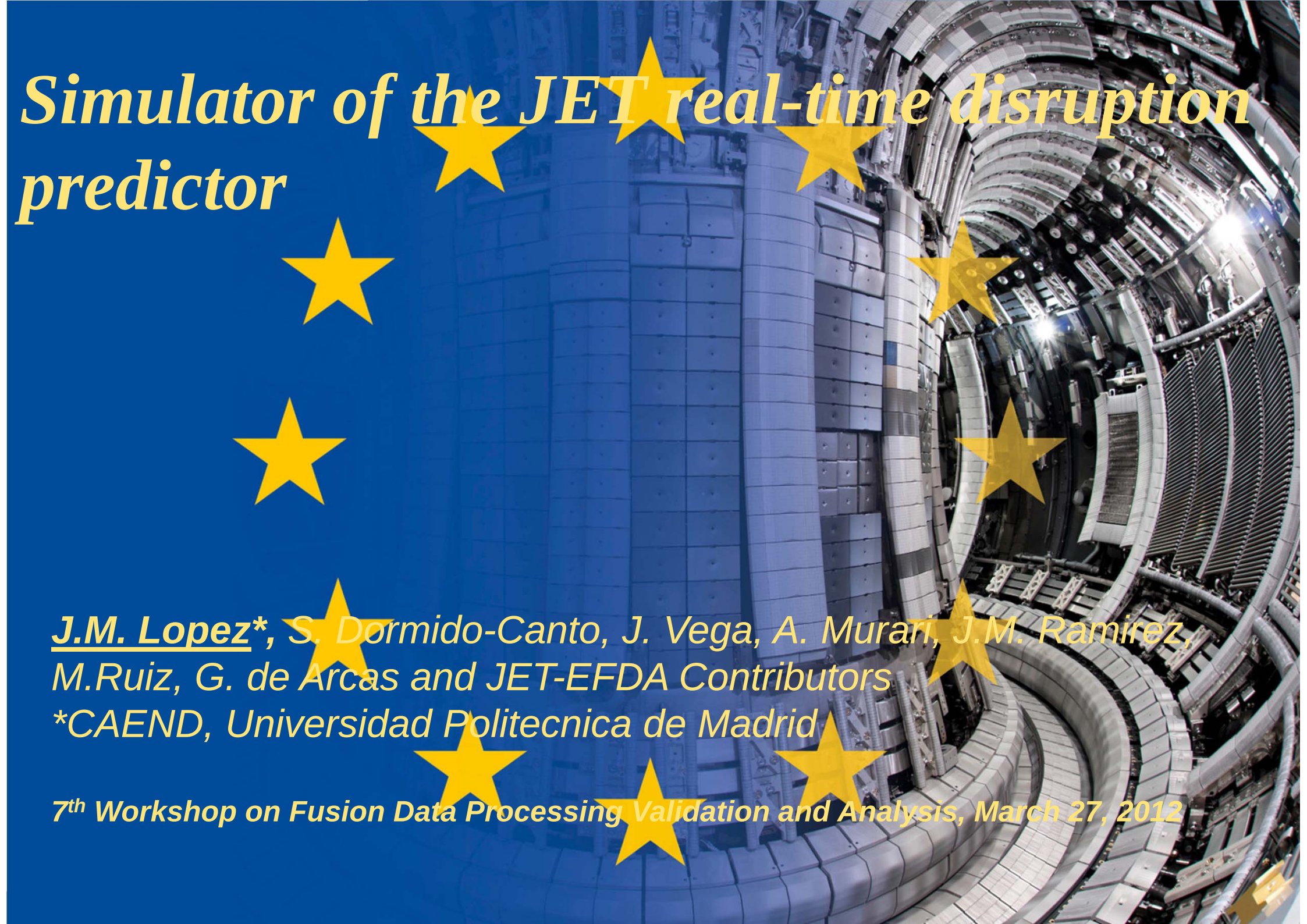
Simulator of the JET real-time disruption predictor



J.M. Lopez, S. Dormido-Canto, J. Vega, A. Murari, J.M. Ramirez, M.Ruiz, G. de Arcas and JET-EFDA Contributors*

**CAEND, Universidad Politecnica de Madrid*

7th Workshop on Fusion Data Processing Validation and Analysis, March 27, 2012

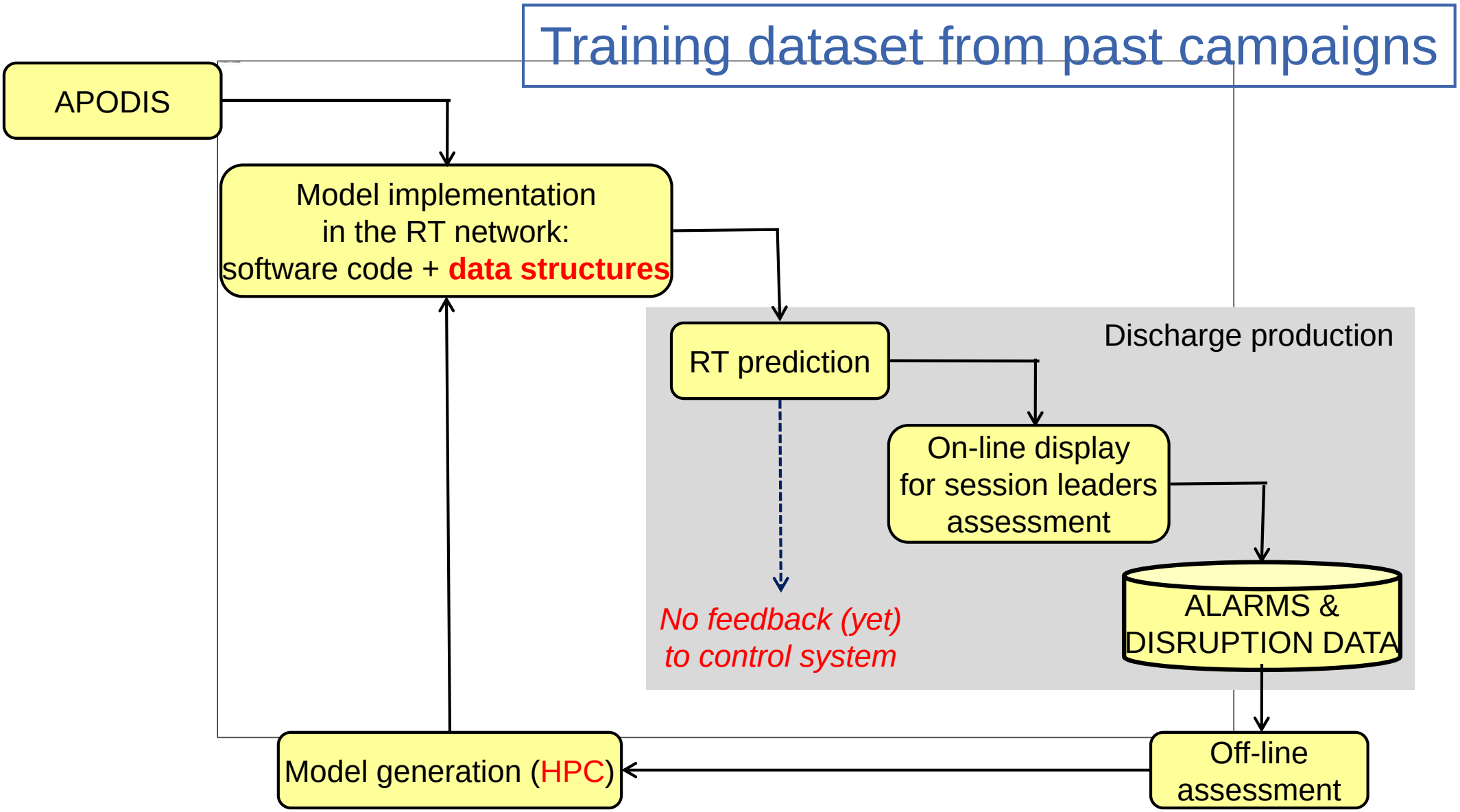
Outline

- Disruption Preditor (Apodis)
- Real-Time simulation constraints
  - Pre-processing
- Real-Time Simulation Implementation
  - JET Real Time Network Peculiarities
- Results
- Acknowledgements

Focus

- Disruption in tokamaks devices are unavoidable and can have catastrophic effects. So it is very important to have mechanisms to predict this phenomenon.
- These mechanisms have to be:
  - Accurate and reliable
    - High success rate
    - Low false alarms
  - With enough time in advance

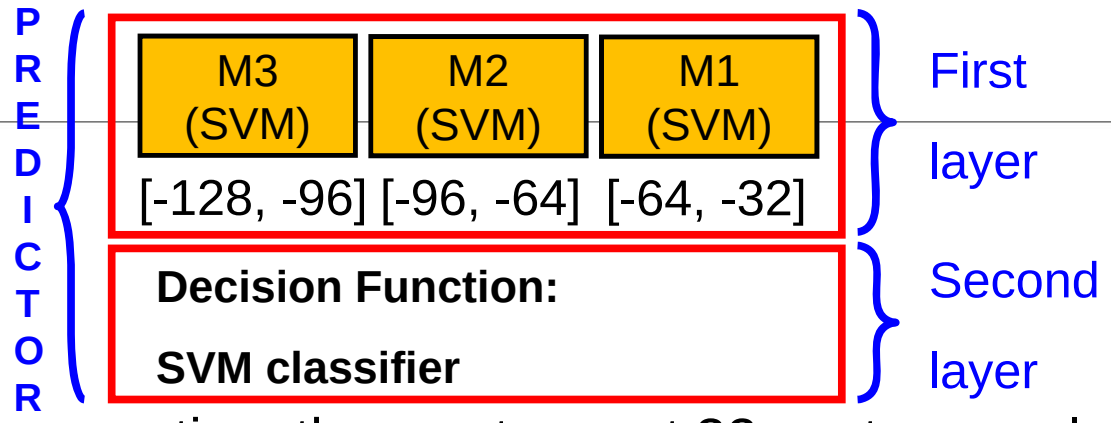
Methodology



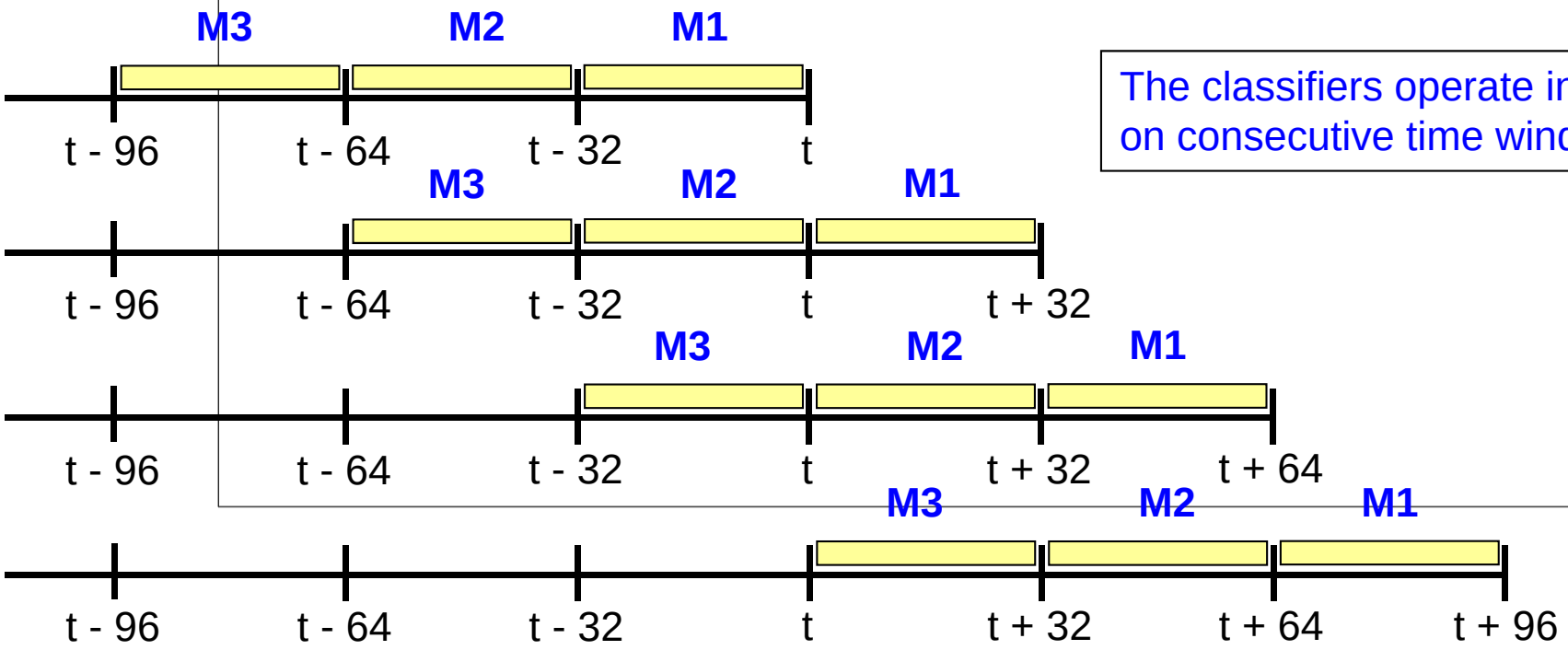
Methodology

- Three steps approach
  - First: Architecture design
    - Model selection and off-line training
  - **Second: Real-time simulator**
    - Simulate the real time acquisition using constraints in JET real-time network
  - Third: Implementation in MARTe framework

Background: Advanced Predictor Of DISruptions (APODIS)



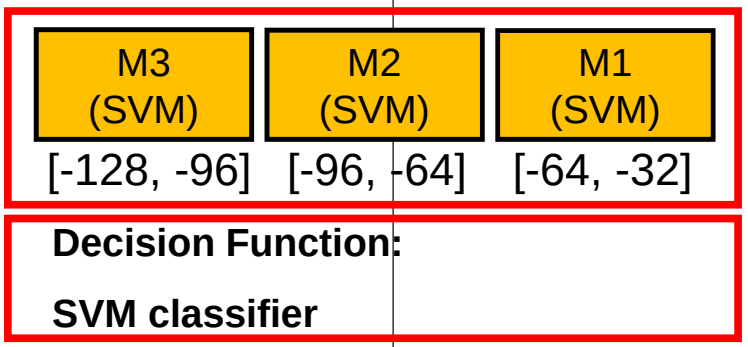
- As a discharge is in execution, the most recent 32 ms temporal segments are classified as disruptive or non-disruptive



- The three models may disagree about the discharge behaviour ⟹ 2nd layer

Background: Advanced Predictor Of DISruptions (APODIS)

- The objective of the training process is to determine a 'predictor model'
- In principle, the predictor model is assessed in terms of success and false alarms rates
- Once determined that balanced datasets are superior to unbalanced ones in relation to training, the real training process started
- 3 sets of features have been used as inputs to the first layer classifiers
  - 14, 16 and 24 features respectively
- 50 random training datasets per set of features were defined for training
  - 100 non-disruptive discharges (randomly selected from 2312)
  - 125 unintentional disruptive discharges (all available disruptions)
- 7500 predictors per set of features have been developed
  - They require a CPU time of 900 h to train the first layer classifiers
  - They require a CPU time of 30 minutes to train the second layer classifier
  - CIEMAT HPC has been used

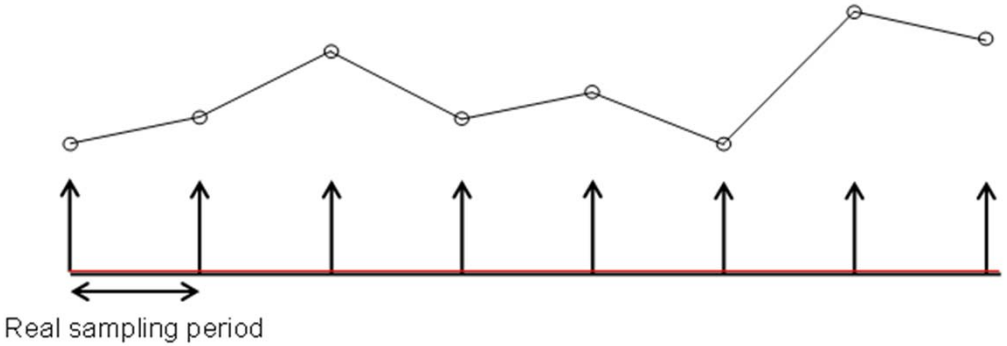


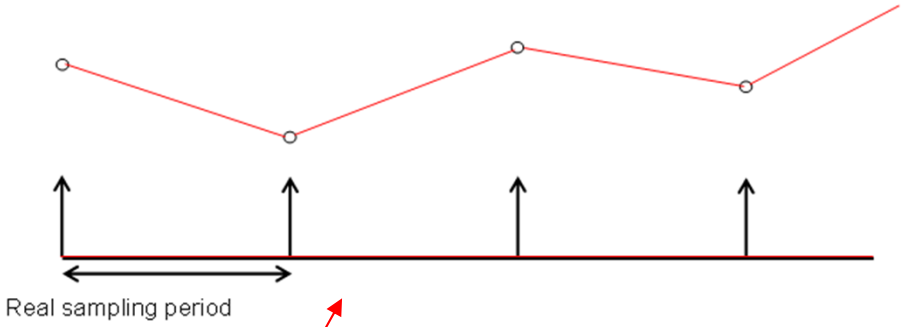
- 240 nodes
- Processors: 2 Quad-Core Xeon (X5450 and X5570) 3.0 GHz
- RAM memory: 16 GB

Background: Advanced Predictor Of DISruptions (APODIS)

7 jpf signals	7 jpf signals + 1 calculated signal	9 jpf signals + 3 calculated signals
Plasma current	Plasma current	Plasma current
Mode lock amplitude	Mode lock amplitude	Mode lock amplitude
Plasma inductance	Plasma inductance	Plasma inductance
Plasma density	Plasma density	Plasma density
Diamagnetic energy time derivative	Diamagnetic energy time derivative	Diamagnetic energy time derivative
Radiated power	Radiated power	Radiated power
Total input power	Total input power	Total input power
	Plasma inductance time derivative	Poloidal beta
		Plasma vertical centroid position
		Plasma inductance time derivative
		Poloidal beta time derivative
		Vertical centroid position time derivative
Mean values & std(abs(FFT(S)))	**Mean values & std(abs(FFT(S)))**	**Mean values & std(abs(FFT(S)))**
14 features	16 features	24 features
7500 predictors	7500 predictors	7500 predictors
MODEL A	MODEL B	MODEL C

Data pre-processing for training

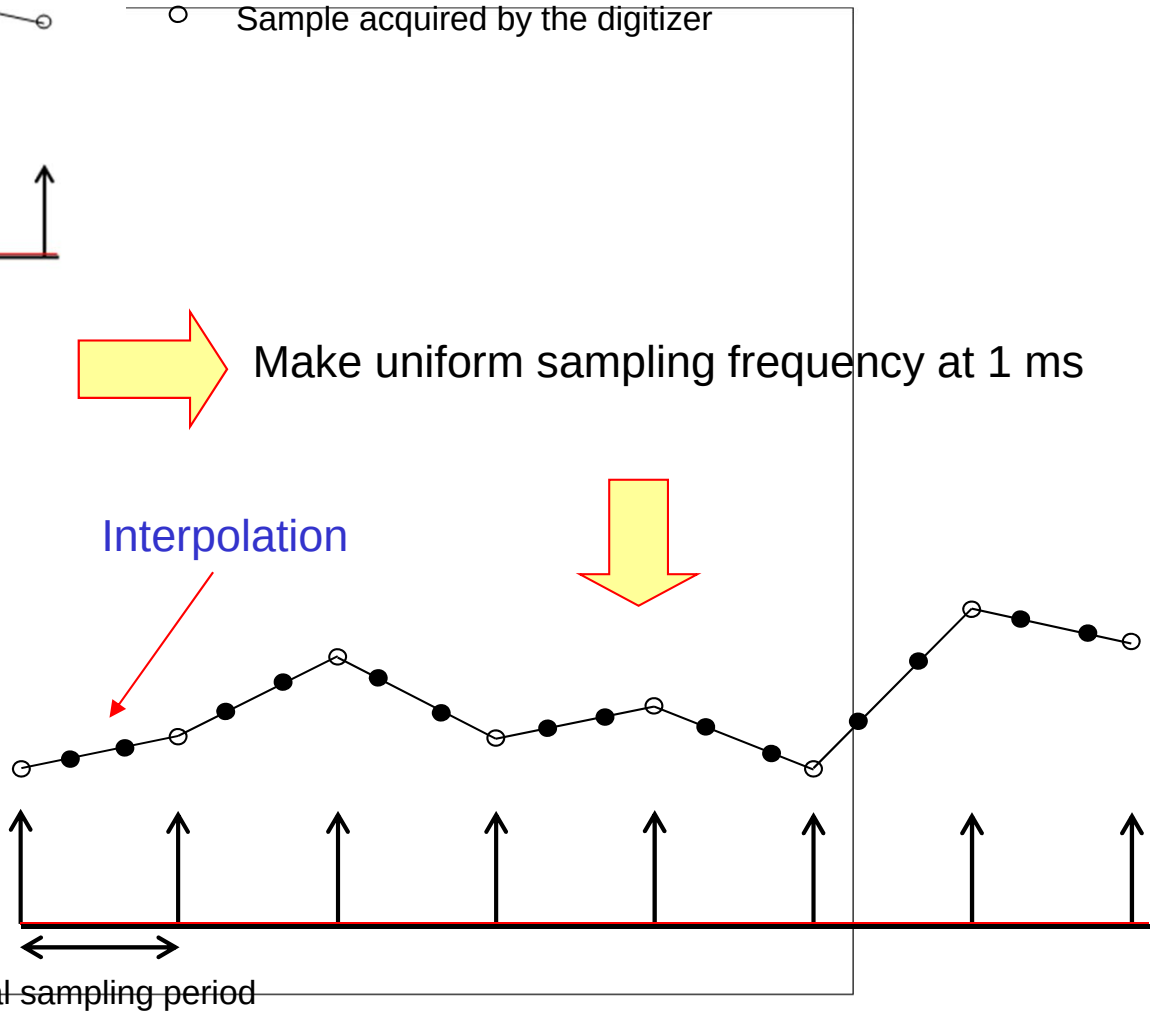




Non homogeneous sampling rate

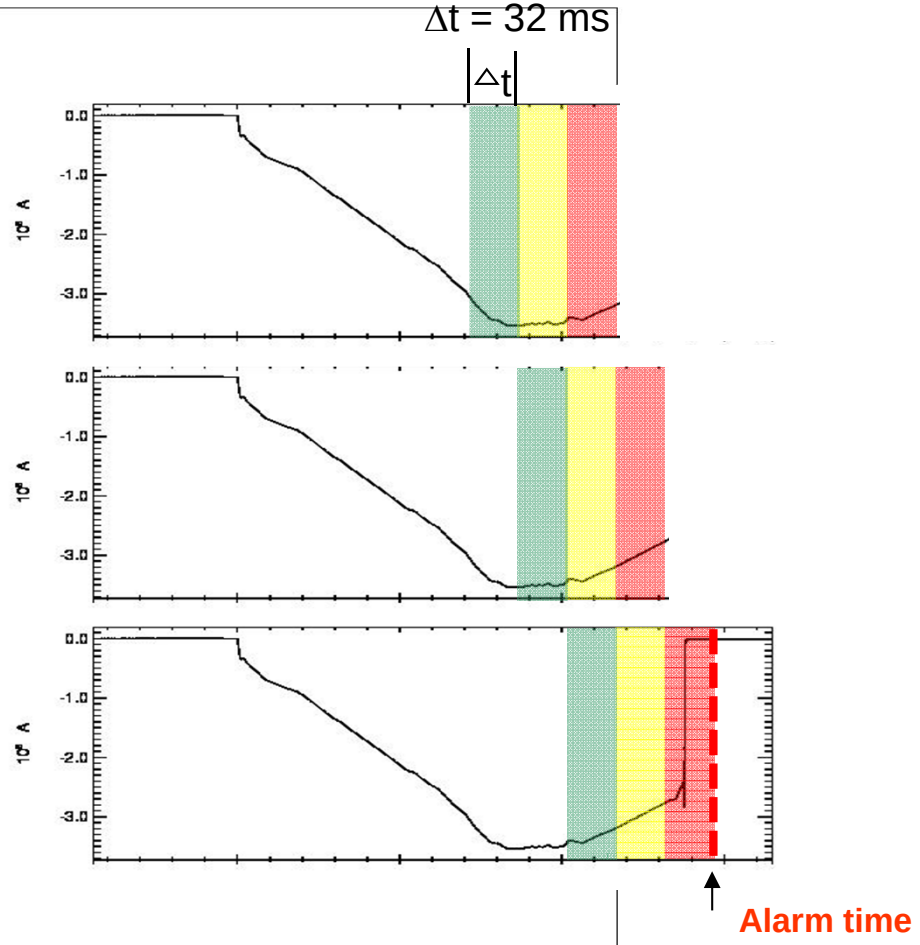
Make uniform sampling frequency at 1 ms

Interpolation



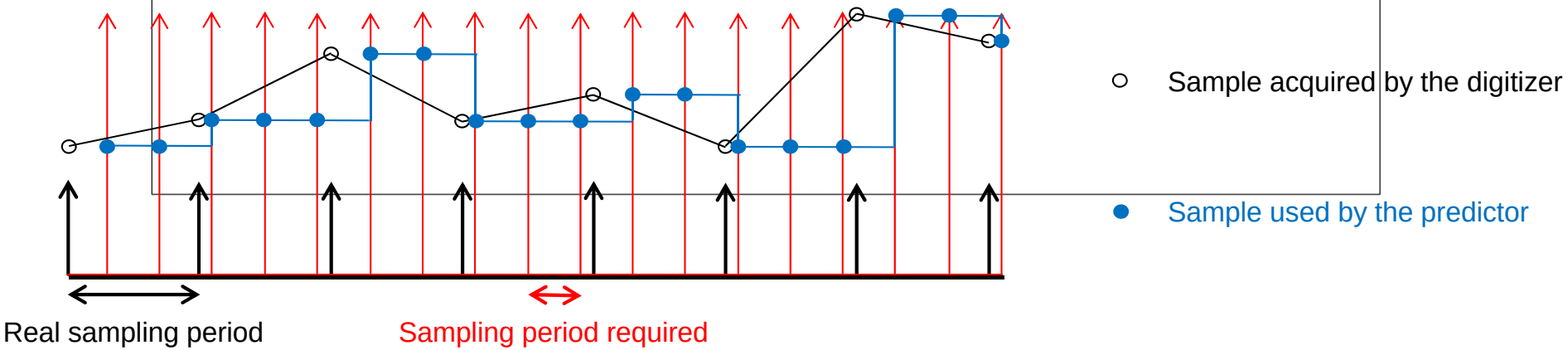
Data pre-processing for training

- The training process is quite different to the real time behaviour
  - Some data manipulation is done to optimize it.
  - Previous knowledge of kind of discharge at disruption time
- In a real time discharge
  - No prior knowledge of windows alignment
  - Time from left to right
  - Fix a threshold trigger to start SVM classifier



Real-time Simulator

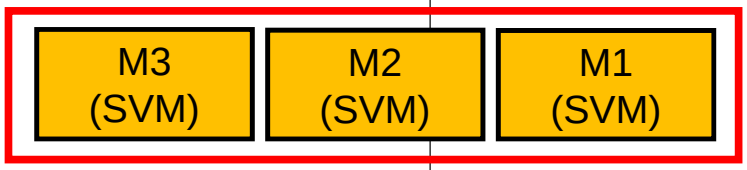
- A software (C language) in the JAC cluster has been developed to simulate the real-time computations
  - The predictor starts when Ip < Threshold (-750 kA)
    - This time instant defines the beginning of the 32 ms long time windows
  - The predictor finishes when Ip > Threshold
  - Input signal from JET Databse
  - Not interpolation but truncation (in some signals, the real sampling period does not meet our sampling requirements)
    - If a sample is request to JET ATM Real Time Network and the sample is not available, then the last one is provide.



Real-time Simulator

- First layer implementation

$$D = \left(\sum_{i=1}^{N} \alpha_i \, e^{-\gamma \left\| vec_i - X \right\|^2}\right) - bias$$



MODEL A	N=210	N=166	N=60
MODEL B	N=210	N=170	N=58
MODEL C	N=213	N=164	N=62

- Decision function

$$R = M_3 \cdot D_3 + M_2 \cdot D_2 + M_1 \cdot D_1 + B$$

Real-time Simulator

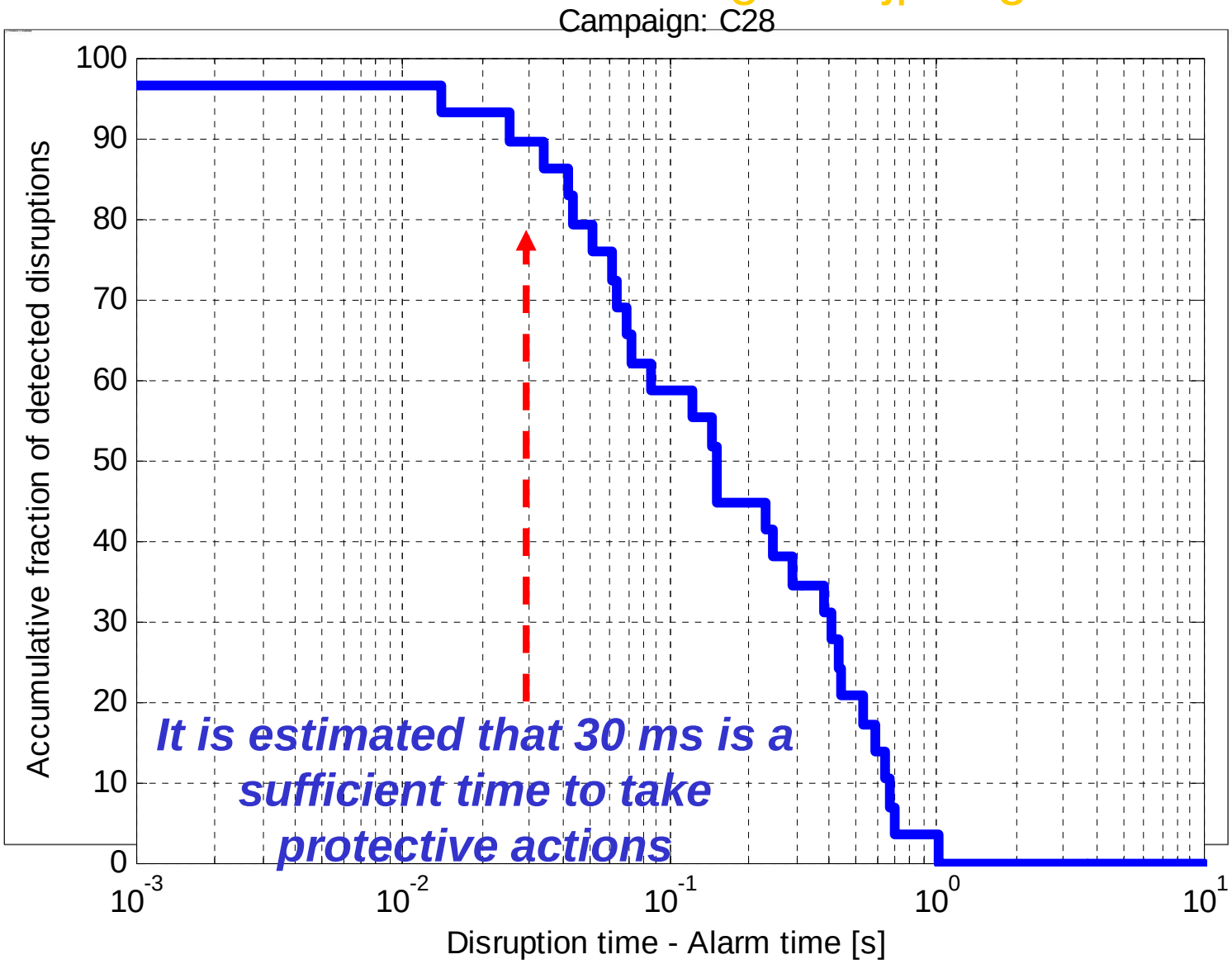
- The simulator is fully configurable by means of text files to select models, signal thresholds, sampling rates, etc

Archivo Edición Ver Insertar Formato Ayuda

```
#**************************************
# Configuration File for Apodis 2
# Lines starting with # are comments
# Remove # for change default values
# **************************************
#
#
#
# Default Threshold value -1100000 A
Threshold  -750000

#Default sampling period 1 ms
#Sampling     0.001

#Default valu for Npoints
#Npoints    32

#Default value for Signals used
#Nsignals      7

#Default value for calculate Signals
NcalculateSignals     0

#Default value for Models windows
#NModels             3

# Default File name for Maximums values
Maximum     ./training02/Max.txt

# Default File name for Mimimumss values
Minimum     ./training02/Min.txt
```

C28 (jpf + ppf) and RT simulation

Non-ILW data for training and jpf signals



Discharge range: 80128 – 81051 [678 discharges analyzed]

Sumary

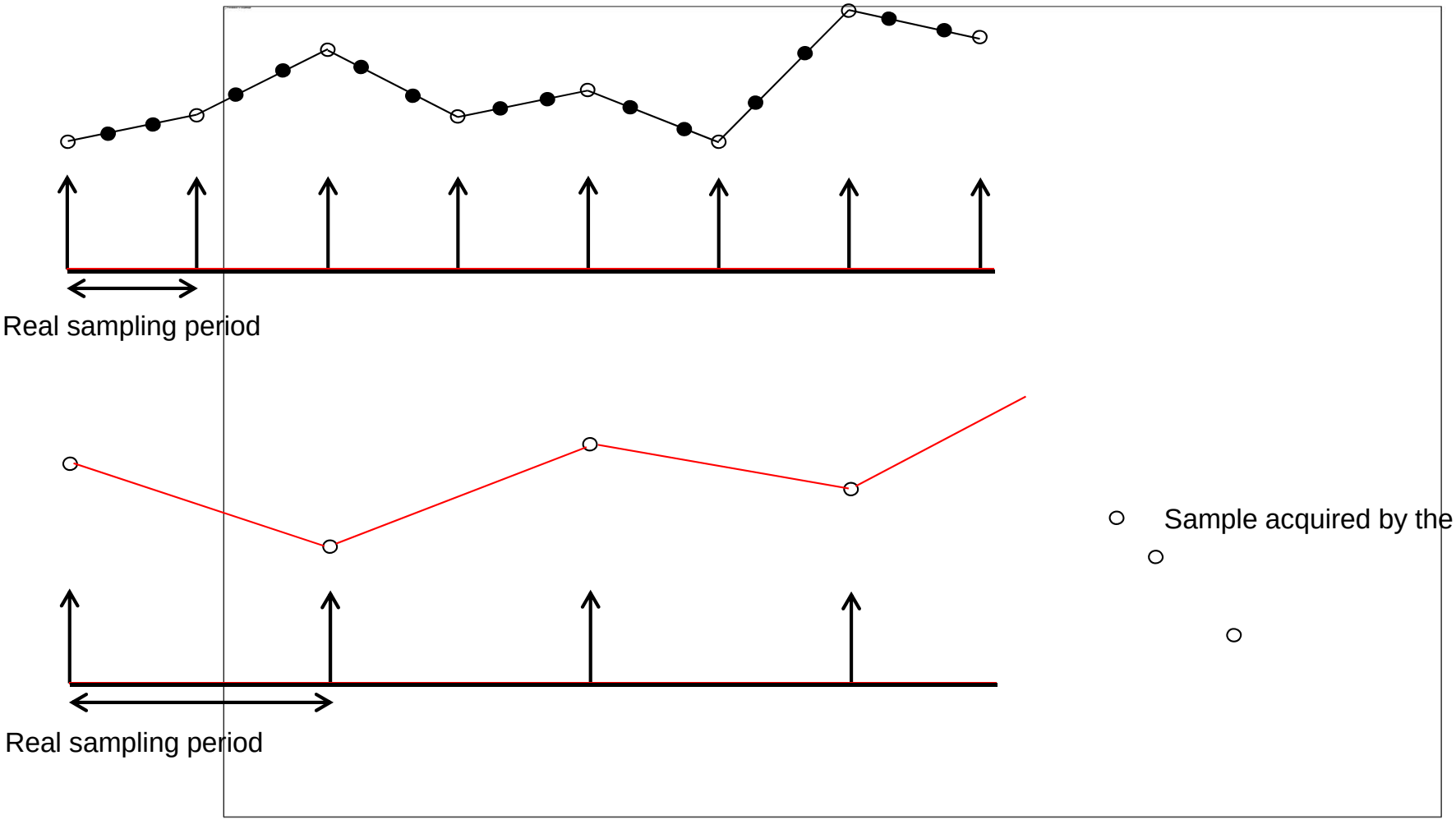
- Tool to test Apodis results using JET Database
  - Works with data files
  - User configurable
  - Can work in background mode (Script)
- Simulate the JET ATM Real Time Network behavior
- Model validation before use the real time application under MARTe framework
- The results are equal that obtained in training phase.

Acknowledgements

- We would like to thank in particular:
  - P. de Vries for the help with the database
  - D. Alves and R. Felton for the support with implementation details in ATM Real Time Network

Thank you very much

Data pre-processing for training



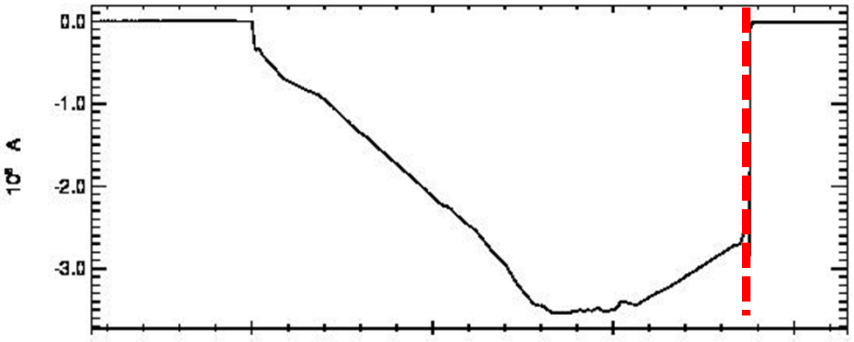
- Resampling at 32 ms

- Backup

Data pre-processing for training

Data preprocessing to optimize the training time

- Data regularity
  - Data resampling at 1 ms
  - Data right alignment to the end pulse
  - Files with complete data window shift evolution



32 ms Data window from left to right

Data window alignment

Files with complete data window and shift evolution

W5	W4	W3	W2	W1
W6	W5	W4	W3	W2
W7	W6	W5	W4	W3
W8	W7	W6	W5	W4
W9	W8	W7	W6	W5

This situation is not real during a discharge

- Time from left to right
- Threshold trigger
  - No prior knowledge of windows alignment