

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Energetica e Nucleare

Tesi di Laurea Magistrale

Optimization of machine learning models for the safeguards verification of spent fuel assemblies



sck cen

Relatore: Dr. Raffaella Testoni
Mentor: Dr. Riccardo Rossa
Co-Mentor: Dr. Alessandro Borella

Candidato:
Federica Roberto

Anno Accademico: 2021-2022

Acknowledgments

First, I would like to personally thank my mentors at SCK CEN: Riccardo Rossa, for the patience, time and attention he always gave to this work, and Alessandro Borella for the valuable technical comments on the project. Moreover, I thank my promoter at Politecnico di Torino, Raffaella Testoni, for being supportive and helpful in any situation during this year.

For the pleasant months spent in Belgium, I would like to thank the Amster-gang, in particular Anna for her unquestionable organization skills, Federico for his introspective vision, Emil for his numerous encouragements, and Victor for his precious support and care. Lastly but not the least, Federica for having been the best office mate.

Now, switching to my native language... un grazie alle splendide persone che ho avuto modo di incontrare a Torino, anche se il tempo a disposizione è stato poco: Marco, Pietro, Andre, Fabio, Michele, Davide, Antonio e Giuseppe. Un ringraziamento più profondo va invece alle due persone il cui tempo passato assieme è stato, per mia fortuna, considerevolmente di più, e la cui presenza è stata essenziale nei periodi di quarantena: Marika e Lorenzo.

Vorrei inoltre ringraziare quelli che sono sempre rimasti i miei amici d'infanzia, presenti anche se lontani. Un grazie a Edoardo, che crede in me da molto prima che ci credessi io. E un grazie di cuore ad Alex, che è stato, è e sarà un punto fermo in ogni fase della mia vita.

Non per ultimo, vorrei ringraziare la mia famiglia, che mi ha sostenuto in un percorso che non è stato sempre facile, e supportato nelle decisioni che mi hanno portata fino a questo punto.

Abstract

Among safeguards verification activities, spent nuclear fuel is placed under inspections for the detection of possible diversions of the fissile material present, being of particular concern from the non-proliferation point of view. Several Non-Destructive Assay are utilized, or under investigation, for the verification of spent nuclear fuel. Among the latter, the Partial Defect Tester (PDET) is the one considered in this work.

In this framework, some machine learning methods have been implemented in this thesis project, in order to detect the replacement of fuel pins in spent nuclear fuel assemblies. The input features of the models were based on different combinations of types and locations of detector responses, whose values were provided by a dataset of Monte Carlo simulations, based on the PDET prototype. The machine learning methods used were supervised regression models, namely k-nearest neighbors and neural networks algorithms. The expected outcomes were the predictions of the number of replaced pins and their locations on the grid lattice.

Within the different implemented configurations of the models, the results showed better performances when the responses of the gamma-ray detectors, for all the locations on the assembly, were involved in the input phase, compared to other models using other detector types. Comparing the different machine learning algorithms, larger accuracies were generally obtained with neural networks compared to k-nearest neighbors method, reaching 97% of correct predictions for location of the replacements.

Contents

Acknowledgments	iii
Abstract	iv
List of figures	vii
List of tables.....	xvi
List of acronyms.....	xvii
1. THEORETICAL FRAMEWORK.....	1
1.1 Nuclear safeguards.....	1
1.1.1 Historical background.....	1
1.1.2 Legal framework.....	2
1.2 Spent nuclear fuel characteristics and safeguards.....	3
1.3 Measurement of spent fuel: NDA techniques.....	5
1.4 NDA technique: PDET	8
1.5 Introduction to machine learning.....	10
1.6 Purpose and structure of the study.....	10
2. DESCRIPTION OF THE DATABASE	12
2.1 Simulated detector responses.....	12
2.2 Database used in the study	12
2.3 Visualization of the database	15
3. MACHINE LEARNING	24
3.1 k-Nearest Neighbors regression model.....	25
3.1.1 Number of dummy pins.....	27
3.1.2 Location of dummy pins	29
3.2 Neural Networks regression model.....	32
3.2.1 Number of dummy pins.....	38
3.2.2. Location of dummy pins	41
4. RESULTS.....	45
4.1 k-Nearest Neighbors regression models	45
4.1.1 Number of dummy pins.....	45
4.1.2 Location of dummy pins	51
4.2 Neural Networks regression models	58
4.2.1 Number of dummy pins.....	58
4.2.2. Location of dummy pins	63
5. CONCLUSIONS	68

REFERENCES 70

APPENDIX A: DATABASE VISUALIZATION 72

 Sum in central detectors 73

 Sum in peripheral detectors 75

 Sum in external detectors 77

 Average in central detectors 79

 Average in external detectors 83

APPENDIX B: RESULTS..... 87

k-NN regression models 87

 Number of dummy pins..... 87

NN regression models 94

 Number of dummy pins..... 94

List of figures

Figure 1.1. Fork DETector irradiated fuel measuring system (FDET) (IAEA, 2011).....	7
Figure 1.2. A spent fuel pond (source: IAEA).....	8
Figure 1.3. Prototype of PDET (Ham et al., 2015)	9
Figure 2.1. Model setup for PDET analysis (Rossa, 2019)	12
Figure 2.2. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Thermal neutrons counts and photons current.....	16
Figure 2.3. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Fast neutrons counts and photons current.....	16
Figure 2.4. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Thermal and fast neutrons counts	17
Figure 2.5. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal neutrons counts and photons current - Number of dummy pins.....	17
Figure 2.6. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal neutrons counts and photons current - Initial Enrichment and BurnUp.....	18
Figure 2.7. Database visualization: Defect scenarios - Average in peripheral detectors - Fast neutrons counts and photons current - Number of dummy pins.....	18
Figure 2.8. Database visualization: Defect scenarios - Average in peripheral detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp	18
Figure 2.9. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal and fast neutrons counts - Number of dummy pins	19
Figure 2.10. Database visualization: Defect scenarios - Average in peripheral detectors – Thermal and fast neutrons counts – Initial Enrichment and BurnUp.....	19
Figure 2.11. Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Thermal neutrons counts - Number of dummy pins.....	20
Figure 2.12. Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Fast neutrons counts - Number of dummy pins.....	20
Figure 2.13 Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Photons current - Number of dummy pins	21
Figure 2.14. Database visualization: Complete and defect scenarios – Relative standard deviation of photons current in peripheral detectors - Number of dummy pins	22
Figure 2.15. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors – Thermal neutrons counts – All peripheral detectors and 4 symmetric peripheral detectors	22
Figure 2.16. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors – Fast neutrons counts – All peripheral detectors and 4 symmetric peripheral detectors	23
Figure 2.17. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors - Photons current – All peripheral detectors and 4 symmetric peripheral detectors.....	23
Figure 3.1. Supervised machine learning method workflow (Rossa, 2019).....	24

Figure 3.2. Machine learning working process.....	25
Figure 3.3. k-NN classification model scheme (Rossa et al., 2020)	26
Figure 3.4. Observation on data: Complete scenario fuel assembly configuration - Thermal neutrons counts	30
Figure 3.5. Observation on data: Complete scenario fuel assembly configuration - Fast neutrons counts.....	30
Figure 3.6. Observation on data: Complete scenario fuel assembly configuration - Photons current.....	30
Figure 3.7. Observation on data: Defect scenario fuel assembly configuration - Thermal neutrons counts	30
Figure 3.8. Observation on data: Defect scenario fuel assembly configuration - Fast neutrons counts	30
Figure 3.9. Observation on data: Defect scenario fuel assembly configuration - Photons current.....	30
Figure 3.10. Generic neural network structure representation (Bhardwaj, 2021)	32
Figure 3.11. Prediction error for overfitted and underfitted models (Smith, 2018).....	37
Figure 3.12. Example of a NN architecture for the prediction of number of dummy pins	40
Figure 3.13. Train and validation losses trends for an illustrative NN model	41
Figure 3.14. SIN model architecture for the prediction of location of dummy pins	42
Figure 3.15. ADI model architecture for the prediction of location of dummy pins.....	42
Figure 3.16. STD model architecture for the prediction of location of dummy pins	43
Figure 3.17. Train and validation losses trends for SIN model.....	43
Figure 3.18. Train and validation losses trends for ADI model	44
Figure 3.19. Train and validation losses trends for STD model	44
Figure 4.1. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Thermal neutrons counts	45
Figure 4.2. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Fast neutrons counts.....	46
Figure 4.3. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Photons current.....	46
Figure 4.4. k-NN model RMSE - Sum in detectors groups for all combinations of kinds of detectors responses.....	47
Figure 4.5. k-NN model RMSE - Average and standard deviation in detectors groups for all combinations of kinds of detectors responses.....	47
Figure 4.6. k-NN model RMSE - Single inputs of detectors groups for all combinations of kinds of detectors responses.....	47
Figure 4.7. k-NN model RMSE - Operations on values within all detectors groups for all combinations of kinds of detectors responses.....	48
Figure 4.8. k-NN model Δ - Ave&Std within all detectors groups for all combinations of kind of detectors responses.....	49

Figure 4.9. Defect scenario fuel assembly configuration - Real locations of dummy pins - Photons current	51
Figure 4.10. k-NN SIN model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	51
Figure 4.11. k-NN ADI model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	51
Figure 4.12. k-NN STD model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	51
Figure 4.13. k-NN SIN model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted scenarios	52
Figure 4.14. k-NN ADI model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios.....	53
Figure 4.15. k-NN STD model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios.....	53
Figure 4.16. Defect scenario fuel assembly configuration – 1/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	55
Figure 4.17. Defect scenario fuel assembly configuration – 1/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	55
Figure 4.18. Defect scenario fuel assembly configuration – 2/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	55
Figure 4.19. Defect scenario fuel assembly configuration – 2/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	55
Figure 4.20. Defect scenario fuel assembly configuration – 3/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	55
Figure 4.21. Defect scenario fuel assembly configuration – 3/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	55
Figure 4.22. Defect scenario fuel assembly configuration – 4/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	56
Figure 4.23. Defect scenario fuel assembly configuration – 4/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	56
Figure 4.24. Defect scenario fuel assembly configuration – 5/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	56
Figure 4.25. Defect scenario fuel assembly configuration – 5/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	56
Figure 4.26. Defect scenario fuel assembly configuration – 6/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	56
Figure 4.27. Defect scenario fuel assembly configuration – 6/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	56
Figure 4.28. Defect scenario fuel assembly configuration – 7/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	57

Figure 4.29. Defect scenario fuel assembly configuration – 7/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	57
Figure 4.30. Defect scenario fuel assembly configuration – 8/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	57
Figure 4.31. Defect scenario fuel assembly configuration – 8/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	57
Figure 4.32. Defect scenario fuel assembly configuration – 9/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	57
Figure 4.33. Defect scenario fuel assembly configuration – 9/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	57
Figure 4.34. Defect scenario fuel assembly configuration – 10/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current	58
Figure 4.35. Defect scenario fuel assembly configuration – 10/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current	58
Figure 4.36. NN model RMSE - Sum in detectors groups for all combinations of kinds of detectors responses	59
Figure 4.37. NN model RMSE - Average and Standard Deviation in detectors groups for all combinations of kinds of detectors responses.....	59
Figure 4.38. NN model RMSE - Single inputs of detectors groups for all combinations of kinds of detectors responses.....	59
Figure 4.39. NN model RMSE - Operations on values within all detectors groups for all combinations of kinds of detectors responses.....	60
Figure 4.40. NN model Δ - Ave&Std within all detectors groups for all combinations of kinds of detectors responses.....	61
Figure 4.41. NN model Δ - Single central, peripheral, and external detectors values for all combinations of kinds of detectors responses.....	61
Figure 4.42. NN SIN model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	63
Figure 4.43. NN ADI model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	63
Figure 4.44. NN STD model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins	63
Figure 4.45. NN SIN model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted scenarios	64
Figure 4.46. NN ADI model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios	64
Figure 4.47. NN STD model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios.....	65
Figure 4.48. Defect scenario fuel assembly configuration – Wrong interpreted scenario 5/9 - Real locations of dummy pins - Photons current.....	66

Figure 4.49. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9- Predicted locations of dummy pins - Photons current	66
Figure 4.50. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 8/9- Real locations of dummy pins - Photons current.....	66
Figure 4.51. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 8/9- Predicted locations of dummy pins - Photons current	66
Figure 4.52. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9 - Real locations of dummy pins - Photons current.....	67
Figure 4.53. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9 - Predicted locations of dummy pins - Photons current	67
Figure A.1. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Thermal neutrons counts and photons current	73
Figure A.2. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Fast neutrons counts and photons current.....	73
Figure A.3. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Thermal and fast neutrons counts	73
Figure A.4. Database visualization: Defect scenarios - Sum in central detectors - Thermal neutrons counts and photons current - Number of dummy pins	74
Figure A.5. Database visualization: Defect scenarios – Sum in central detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp.....	74
Figure A.6. Database visualization: Defect scenarios – Sum in central detectors - Fast neutrons counts and photons current - Number of dummy pins	74
Figure A.7. Database visualization: Defect scenarios – Sum in central detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp.....	74
Figure A.8. Database visualization: Defect scenarios – Sum in central detectors - Thermal and fast neutrons counts - Number of dummy pins.....	74
Figure A.9. Database visualization: Defect scenarios – Sum in central detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp	74
Figure A.10. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Thermal neutrons counts and photons current.....	75
Figure A.11. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Fast neutrons counts and photons current.....	75
Figure A.12. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Thermal and fast neutrons counts	75
Figure A.13. Database visualization: Defect scenarios - Sum in peripheral detectors - Thermal neutrons counts and photons current - Number of dummy pins.....	76
Figure A.14. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp	76
Figure A.15. Database visualization: Defect scenarios – Sum in peripheral detectors - Fast neutrons counts and photons current - Number of dummy pins	76

Figure A.16. Database visualization: Defect scenarios – Sum in peripheral detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp.....	76
Figure A.17. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal and fast neutrons counts - Number of dummy pins	76
Figure A.18. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp.....	76
Figure A.19. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Thermal neutrons counts and photons current	77
Figure A.20. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Fast neutrons counts and photons current.....	77
Figure A.21. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Thermal and fast neutrons counts	77
Figure A.22. Database visualization: Defect scenarios - Sum in external detectors - Thermal neutrons counts and photons current - Number of dummy pins	78
Figure A.23. Database visualization: Defect scenarios – Sum in external detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp	78
Figure A.24. Database visualization: Defect scenarios – Sum in external detectors - Fast neutrons counts and photons current - Number of dummy pins	78
Figure A.25. Database visualization: Defect scenarios – Sum in external detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp.....	78
Figure A.26. Database visualization: Defect scenarios – Sum in external detectors - Thermal and fast neutrons counts - Number of dummy pins	78
Figure A.27. Database visualization: Defect scenarios – Sum in external detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp.....	78
Figure A.28. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Thermal neutrons counts and photons current.....	79
Figure A.29. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Fast neutrons counts and photons current.....	79
Figure A.30. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Thermal and fast neutrons counts	79
Figure A.31. Database visualization: Defect scenarios - Average in central detectors - Thermal neutrons counts and photons current - Number of dummy pins.....	80
Figure A.32. Database visualization: Defect scenarios – Average in central detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp	80
Figure A.33. Database visualization: Defect scenarios – Average in central detectors - Fast neutrons counts and photons current - Number of dummy pins	80
Figure A.34. Database visualization: Defect scenarios – Average in central detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp.....	80
Figure A.35. Database visualization: Defect scenarios – Average in central detectors - Thermal and fast neutrons counts - Number of dummy pins	80

Figure A.36. Database visualization: Defect scenarios – Average in central detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp.....	80
Figure A.37. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Thermal neutrons counts - Number of dummy pins	81
Figure A.38. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Fast neutrons counts - Number of dummy pins	81
Figure A.39. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Photons current - Number of dummy pins.....	81
Figure A.40. Database visualization: Complete and defect scenarios - Relative standard deviation of photons current in central detectors - Number of dummy pins	82
Figure A.41. Database visualization: Complete and defect scenarios – Average in external detectors and relative uncertainties – Thermal neutrons counts and photons current.....	83
Figure A.42. Database visualization: Complete and defect scenarios - Average in external detectors and relative uncertainties – Fast neutrons counts and photons current.....	83
Figure A.43. Database visualization: Complete and defect scenarios - Average in external detectors and relative uncertainties – Thermal and fast neutrons counts	83
Figure A.44. Database visualization: Defect scenarios - Average in external detectors - Thermal neutrons counts and photons current - Number of dummy pins.....	84
Figure A.45. Database visualization: Defect scenarios – Average in external detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp	84
Figure A.46. Database visualization: Defect scenarios – Average in external detectors - Fast neutrons counts and photons current - Number of dummy pins	84
Figure A.47. Database visualization: Defect scenarios – Average in external detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp.....	84
Figure A.48. Database visualization: Defect scenarios – Average in external detectors - Thermal and fast neutrons counts - Number of dummy pins	84
Figure A.49. Database visualization: Defect scenarios – Average in external detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp.....	84
Figure A.50. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Thermal neutrons counts - Number of dummy pins	85
Figure A.51. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Fast neutrons counts - Number of dummy pins	85
Figure A.52. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Photons current - Number of dummy pins.....	85
Figure A.53. Database visualization: Complete and defect scenarios - Relative standard deviation of photons current in external detectors - Number of dummy pins.....	86
Figure B.1. k-NN model Δ - Sum within central detectors for all combinations of kinds of detectors responses	87
Figure B.2. k-NN model Δ - Sum within peripheral detectors for all combinations of kinds of detectors responses.....	87

Figure B.3. k-NN model Δ - Sum within external detectors for all combinations of kinds of detectors responses.....	88
Figure B.4. k-NN model Δ - Sum within central and peripheral detectors for all combinations of kinds of detectors responses	88
Figure B.5. k-NN model Δ - Sum within central and external detectors for all combinations of kinds of detectors responses	88
Figure B.6. k-NN model Δ - Sum within peripheral and external detectors for all combinations of kinds of detectors responses	89
Figure B.7. k-NN model Δ - Sum within central, peripheral and external detectors for all combinations of kinds of detectors responses.....	89
Figure B.8. k-NN model Δ - Ave&Std within central detectors for all combinations of kind of detectors responses.....	89
Figure B.9. k-NN model Δ - Ave&Std within peripheral detectors for all combinations of kinds of detectors responses.....	90
Figure B.10. k-NN model Δ - Ave&Std within external detectors for all combinations of kinds of detectors responses.....	90
Figure B.11. k-NN model Δ - Ave&Std within central and peripheral detectors for all combinations of kinds of detectors responses	90
Figure B.12. k-NN model Δ - Ave&Std within central and external peripheral detectors for all combinations of kinds of detectors responses	91
Figure B.13. k-NN model Δ - Ave&Std within peripheral and external detectors for all combinations of kinds of detectors responses.....	91
Figure B.14. k-NN model Δ – Single central detectors values for all combinations of kinds of detectors responses.....	91
Figure B.15. k-NN model Δ – Single peripheral detectors values for all combinations of kinds of detectors responses.....	92
Figure B.16. k-NN model Δ – Single external detectors values for all combinations of kinds of detectors responses.....	92
Figure B.17. k-NN model Δ – Single central and peripheral detectors values for all combinations of kinds of detectors responses	92
Figure B.18. k-NN model Δ – Single central and external detectors values for all combinations of kinds of detectors responses	93
Figure B.19. k-NN model Δ – Single peripheral and external detectors values for all combinations of kinds of detectors responses.....	93
Figure B.20. k-NN model Δ – Single central, peripheral and external detectors values for all combinations of kinds of detectors responses	93
Figure B.21. NN model Δ - Sum within central detectors for all combinations of kinds of detectors responses	94
Figure B.22. NN model Δ - Sum within peripheral detectors for all combinations of kinds of detectors responses.....	94

Figure B.23. NN model Δ - Sum within external detectors for all combinations of kinds of detectors responses.....	95
Figure B.24. NN model Δ - Sum within central and peripheral detectors for all combinations of kinds of detectors responses	95
Figure B.25. NN model Δ - Sum within central and external detectors for all combinations of kinds of detectors responses	95
Figure B.26. NN model Δ - Sum within peripheral and external detectors for all combinations of kinds of detectors responses	96
Figure B.27. NN model Δ – Sum within central, peripheral and external detectors for all combinations of kinds of detectors responses.....	96
Figure B.28. NN model Δ - Ave&Std within central detectors for all combinations of kinds of detectors responses.....	96
Figure B.29. NN model Δ - Ave&Std within peripheral detectors for all combinations of kinds of detectors responses.....	97
Figure B.30. NN model Δ - Ave&Std within external detectors for all combinations of kinds of detectors responses.....	97
Figure B.31. NN model Δ - Ave&Std within central and peripheral detectors for all combinations of kinds of detectors responses	97
Figure B.32. NN model Δ - Ave&Std within central and external detectors for all combinations of kinds of detectors responses	98
Figure B.33. NN model Δ - Ave&Std within peripheral and external detectors for all combinations of kinds of detectors responses.....	98
Figure B.34. NN model Δ - Single central detectors values for all combinations of kinds of detectors responses.....	98
Figure B.35. NN model Δ – Single peripheral detectors values for all combinations of kinds of detectors responses.....	99
Figure B.36. NN model Δ – Single external detectors values for all combinations of kinds of detectors responses.....	99
Figure B.37. NN model Δ - Single central and peripheral detectors values for all combinations of kinds of detectors responses	99
Figure B.38. NN model Δ - Single central and external detectors values for all combinations of kinds of detectors responses	100
Figure B.39. NN model Δ - Single peripheral and external detectors values for all combinations of kinds of detectors responses	100

List of tables

Table 1.1. Spent fuel measurement systems (IAEA, 2011)	6
Table 2.1. Examples of defect scenarios configurations	14
Table 2.2. Extract from PDET detectors responses dataset	15
Table 3.1. Activation functions in NN (Jagtap, 2022)	33
Table 4.1. k-NN models: Mean RMSE and standard deviation – Operation on values: Sum.....	50
Table 4.2. k-NN models: Mean RMSE and standard deviation – Operation on values: Average and Standard Deviation	50
Table 4.3. k-NN models: Mean RMSE and standard deviation – Operation on values: Single input.....	50
Table 4.4. k-NN models: Location of dummy pins - Summarized results.....	54
Table 4.5. NN models: Mean RMSE and standard deviation – Operation on values: Sum	62
Table 4.6. NN models: Mean RMSE and standard deviation – Operation on values: Average and Standard Deviation	62
Table 4.7. NN models: Mean RMSE and standard deviation – Operation on values: Single input	62
Table 4.8. NN models: Location of dummy pins - Summarized results.....	65

List of acronyms

AI	Artificial Intelligence
API	Application Programming Interface
BU	BurnUp
C/S	Containment and Surveillance
CLAB	Central Interim Spent Fuel Storage Facility
CSA	Comprehensive Safeguards Agreement
CT	Cooling Time
CV	Cross Validation
DA	Destructive Analysis
DCVD	Digital Cherenkov Viewing Device
FA	FAst neutrons
FDET	Fork DETector irradiate fuel measuring system
GWd/t_{HM}	GigaWatt-days/ton of Heavy Metal
IAEA	International Atomic Energy Agency
ICVD	Cherenkov Viewing Device
IE	Initial Enrichment
IRAT	IRradiated fuel Attribute Tester
JCPOA	Joint Comprehensive Plan Of Action
k-NN	k-Nearest Neighbors
LLNL	Lawrence Livermore National Laboratory
ML	Machine Learning
MSE	Mean Squared Error
NDA	Non-Destructive Assay
NMA	Nuclear Material Accountancy
NPT	Non-Proliferation Treaty
P	Gamma-rays
PDET	Partial DEfect Tester
PGET	Passive Gamma Emission Tomography
PWR	Pressurized Water Reactor
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error

RSD	Relative Standard Deviation
SFAT	Spent Fuel Attribute Tester
SGD	Stochastic Gradient Descent
SSAC	State System for Accounting for and Control of nuclear material
TH	Thermal neutrons
UN	United Nations
UV	Ultra-Violet

1. THEORETICAL FRAMEWORK

1.1 Nuclear safeguards

1.1.1 Historical background

The history of the use of nuclear energy for civil purposes has been carried on a parallel trail with the military development since its first appearance on the global scene. Indeed, the first artificial nuclear reactor went critical in December 1942 and the first nuclear weapon test was carried out in July 1945. Nuclear weapons were then used in August 1945 in Hiroshima and Nagasaki, bringing to the end the Second World War. The destructive potential of nuclear material when used in nuclear weapons urged States to draft legislation to regulate the use of nuclear technology.

Even if few attempts have been tried to build up organizations and legislations among the States involved in second post-war period, the very first effective program was achieved only in the 1955, when the base for the International Atomic Energy Agency (IAEA) was founded and later formally established in 1957. In the same year another supernational organization was established: Euratom, the European Atomic Energy Community which has the aim of promoting the peaceful uses of nuclear energy.

From the IAEA statute, the main objective of the agency were to “promote global dissemination of civilian nuclear technology and know-how; and to supervise and control this technology and know-how in order to prevent the proliferation of nuclear weapons (Article II)” (ESARDA, 2008). In order to fulfill this task, the Agency implemented in Article XII of the statute the main points which allowed the rights of inspections and supervision by the Agency itself. The latter can be summarized in the following seven points (IAEA, 1989):

1. To examine and approve the design of nuclear related facilities, in order to verify they will not promote military purposes and that safeguards measures will be easily applied (Article XII.A.1);
2. To require the fulfillment of health and safety Agency prescribed measures (Article XII.A.2);
3. To demand operating records of the materials that have to be traced (Article XII.A.3);
4. To demand and obtain progress reports (Article XII.A.4);
5. To approve procedures and methods used for chemical processing of irradiated materials, to avoid possible military misuse, to require that special fissionable materials recovered or produced as by-product are used for peaceful purpose, and to require deposit of possible excess of these by-products by the Agency, in order to avoid stockpiling of material when State does not necessitate it (Article XII.A.5);
6. To send Agency inspectors in nuclear facilities, who must have access any time to any place, data and person dealing with materials, equipment or facilities which require to be safeguarded, to verify compliance of the agreements (Article XII.A.6);
7. To suspend/terminate assistance by the Agency or withdraw any material and equipment in case corrective steps are not implemented within a reasonable time, when an event of non-compliance by the State is detected (Article XII.A.7).

However, it was just with the Non-Proliferation Treaty (NPT), opened for signature in 1968 and entered into force in 1970, that the IAEA confirmed his status of supernational party responsible for safeguards on world scale. The NPT has been eventually ratified in 2022 by 191 states, and covers three critical objectives: (1) to prevent the dissemination of nuclear weapons, (2) to promote nuclear disarmament, (3) to promote the peaceful use of nuclear energy (ESARDA, 2008).

After a few decades of relative successful non-proliferation-system, the misleading trend of some of the countries belonging to the IAEA safeguards control, highlighted the need of additional measures to prevent further inappropriate actions. Following this idea, in May 1997 the Model Additional Protocol was approved, which requires the States to provide to the IAEA more information, and extends the inspections rights of the Agency, especially allowing a system of short-notice inspections.

In recent years the Joint Comprehensive Plan of Action (JCPOA) was given ample space in media. The JCPOA is a milestone political agreement on the Iranian nuclear program, negotiated between Iran and the so-called P5+1 (the permanent members of the UN Security Council, China, France, Russian Federation, United Kingdom, and the United States of America – plus Germany) and the European Union. Under the JCPOA the capabilities of Iran to enrich uranium (i.e., increase the percentage of ^{235}U) are limited and additional inspections by IAEA are foreseen.

1.1.2 Legal framework

Safeguards rely on a wide area of agreements that have been stipulated between the IAEA and a Member State, ensuring that the latter must be compliant with the agreements accepted. In this perspective, under the IAEA Comprehensive Safeguards Agreement (CSA), the objectives are “timely detection of diversion of significant quantities of nuclear material from peaceful nuclear activities to the manufacture of nuclear weapons or of other nuclear explosive devices for purposes unknown, and deterrence of such diversion by the risk of early detection” (IAEA, 1972). In addition, the Additional Protocol extends the possibilities of the IAEA to carry out inspections and enlarges the information that States must provide to the IAEA (e.g., research and development plants). In case a non-compliance condition is detected, the IAEA Director General shall report to the IAEA Boards of Governors, and lately to all members of the Security Council and General Assembly of UN, which will force the State in question to remedy the situation (IAEA, 2002).

Referring to the CSA, the safeguards provisions cover the nuclear fuel cycle, as soon as any nuclear material, ready for fuel fabrication or for being isotopically enriched due to its composition and purity, leaves the plant or the process stage in which it has been produced. The same must be applied when any nuclear material, at any level of the nuclear fuel cycle, is being transferred to other states. The safeguards inspections are carried on until the material cannot be further used for any nuclear activities relevant from the safeguards point of view (i.e., consumed, diluted, or transferred to a non-nuclear use). In addition, under IAEA supervision has to be also included the information regarding location or further processing of intermediate or high-level waste containing plutonium, high enriched uranium or ^{233}U , on which the direct safeguards approach has been concluded.

Many actions are foreseen with the aim of enforcing the safeguards agreements between IAEA and the State: Nuclear Material Accountancy (NMA), Containment and Surveillance (C/S) measures and monitoring, visits and inspections are some of the complementary activities carried out for safeguards verification.

NMA activities are implemented at three different levels independently: by the facility operators, by the State System for Accounting for and Control of nuclear material (SSAC), and finally by the IAEA itself to verify the correctness of the information provided in the records and reports provided by the two previous levels respectively. In general, the NMA process has the meaning of establishing the quantities of nuclear materials located in certain predefined areas, and the variation of those quantities within periods of time, taking into account also measurements uncertainties.

The C/S measures and monitoring are complementary to the NMA activities, and generally lead to less frequent IAEA inspections and to achieve acceptable costs. Indeed, the intent of C/S is to maintain the continuity of knowledge between inspections. Examples of containments are seals that are used to impede access to an area or item. Examples of surveillance are cameras that are covering the movements in areas where nuclear material is stored. Monitoring devices are implemented as a complementary function since they verify the flow of nuclear materials within a certain area. In general, they are identified as radiation measurements monitoring systems.

In accordance with the safeguards agreements, inspectors are supposed to carry out visits and inspections on facilities and locations outside the facilities. In case a State is also adherent to the Additional Protocol, the rights of complementary access can be as well foreseen. The scopes of the inspections are to verify the NMA declarations, examine the records, perform independent measurements of nuclear materials under the safeguards agreements, and establish C/S measures. The inspections could be planned, short-notice or no-notice inspections. Some examples are Physical Inventory Verification (PIV), and Interim inspections.

1.2 Spent nuclear fuel characteristics and safeguards

Among the materials that can be identified along the nuclear fuel cycle, the IAEA defines the so-called "Direct use material", i.e., those that can be directly used in the development of nuclear weapons without transmutation or further enrichment. The definition includes plutonium (having less than 80% ^{238}Pu), high enriched uranium (where the percentage of the isotope ^{235}U is at least 20%) and ^{233}U . In the same category can be identified all the mixtures or chemical compounds of the materials above mentioned, and as well the plutonium from spent reactor fuel. This category can be further subdivided in unirradiated and irradiated direct use material. The former is the one not containing substantial amount of fission products, thus it can be easily used into nuclear explosive devices, whereas the latter contains a higher amount of fission products (e.g., spent nuclear fuel) and needs further processing before being used in nuclear explosive devices. On the contrary, in the "Indirect use material" category, it can be recognized all the nuclear material which requires further procedures for producing direct use material (e.g., depleted, natural and low enriched uranium, thorium) (IAEA, 2002). Hence, since the irradiated fuel, i.e., the spent

nuclear fuel, contains fissile material as irradiated direct use material, it requires specific attention from the safeguards perspective.

Spent nuclear fuel stands for the fuel which has been appropriately irradiated in a nuclear power plant, and that cannot further sustain the nuclear chain reaction in a thermal reactor, therefore it has to be substituted with fresh fuel for a suitable production of energy. Having been irradiated for an amount of time suitable for the energy production, which depends basically on the kind of reactor and on the operational power, the fuel has gradually decreased the amount of fissile nuclides present and increased the quantity of neutron-absorbing nuclides (i.e., non-fissile actinides and fission products). Moreover, it can be characterized by a measure of fuel burnup, which refers to how much energy has been extracted from the nuclear fuel. This parameter is strictly correlated to the isotopic composition of the spent fuel.

The presence of fission products and transuranic elements requires the management of the spent fuel paying attention to radioactivity and decay heat, due to the radioactive decay process of the elements involved. A common procedure for spent fuel management, in a nuclear power plant, is to store it in a spent fuel pool for a suitable cooling time, usually several years, until the decay heat and the radioactivity have reached an acceptable level to be moved to long-term storage or reprocessing facilities.

The composition of the nuclear fuel changes substantially over the operational years in a nuclear power reactor. Considering a common thermal reactor, the fresh fuel is composed by low enriched uranium (enriched uranium containing less than 20% of the isotope ^{235}U , normally limited to 4-5% in thermal reactor), where the enriched part is slowly depleted whereas the fertile ^{238}U isotope, the most abundant part, leads mainly to the formation of the fissile ^{239}Pu . Discharged fuel eventually contains still around 95% of Uranium-238, 0.8% of Plutonium, 1% of Uranium-235 and about 0.5% Uranium-236. About 3% of irradiated fuel is constituted by fission products, as a result of the fission reactions. More precisely, different levels of burnup affect the spent fuel composition (Hargraves & Moir, 2010). Most of the fission products are highly unstable and undergo through radioactive decays, being short- or mid- to long-life isotopes.

Due to the particular isotopic composition of the spent fuel, containing in fact fissile material, it is of concern from the non-proliferation point of view, and generally even more than low enrichment fresh fuel (categorized as indirect use material). Furthermore, the large radiation and decay heat from spent fuel make it noticeably difficult to be handled, adding also further challenges for the safeguards measurements. Spent fuel is generally measured with Non-Destructive Assay (NDA), other than complementary measures such as containments, seals, surveillance, and monitoring, which are used to reduce radiation exposure to operators and inspectors. Moreover, for safety reasons, the movement and handling of spent fuel is generally limited (IAEA, 2018).

In general, stakeholders such as operators, regulators and inspectors require to know the location and the composition of spent fuel assemblies, as well as the characteristic of fuel and the presence/absence of some fuel pins. In more detail, IAEA's intention is to assure the detection of gross, partial, or bias defects during the verification of a facility.

1.3 Measurement of spent fuel: NDA techniques

In the set of techniques that have been developed for safeguards aims, the NDA techniques imply the measurement of the nuclear material composition without compromising or altering the physical and chemical integrity of the item under investigation. The methods generally foreseen a radiometric emission or response from the item, and, in certain cases, the comparison of this observation with an equivalent calibration item based on a previously performed NDA, or by means of computer simulations. NDA methods can be split in two wide categories:

- Passive NDA: where the radiation measurements are conducted on spontaneous decay of the item;
- Active NDA: where the item is stimulated by irradiation or activation before or in the meanwhile the radiation measurements are carried out.

The main NDA techniques are principally classified as calorimetry, gamma-ray assay, neutron assay and spent fuel measurements, where a relevant role is covered by the Cherenkov viewing devices (IAEA, 2011).

Calorimetry is mainly used for the evaluation of the content of plutonium in an item, specifically it relies on its thermal power emission, due to the radioactive decay of Pu isotopes, in combination with the information of the Pu isotopic mass ratio. It delivers a measure of the total Pu mass in samples of unknown composition. Anyway, it has the disadvantage of needing a long time (e.g. several hours) for reaching thermal equilibrium, thus is not a common routine measurement used for spent fuel (ESARDA, 2008).

Radioactive decay may lead to emission of gamma-rays, which interact with surrounding matter and leave part of the energy to the electrons present, in a process called ionization. The ionized electron may further collide to other atoms and liberate other electrons. These electrons might be collected, in order to record the presence of gamma-rays and to measure the energy carried. Moreover, the resulting response is an electrical pulse whose voltage is proportional to the energy deposited in the detector, thus the measurement of the energy spectrum of the sample is also investigated (Relly, 1991). Due to the very strong gamma-ray emission from spent fuel, gamma-ray measurements have to be carried out with proper shielding of the detector. Nevertheless, gamma-ray measurements are used for spent fuel verifications, to confirm characteristics, cooling time, initial enrichment and burnup of fuel assemblies (ESARDA, 2008).

Neutron counting methods measure the emission of neutrons from nuclear material in case of (α , n) reactions, spontaneous or induced emission. It basically consists of counting the number of neutrons emitted, regardless their specific energy, with the purpose of estimate the mass of fissile material in the measured sample. These methods can be divided in different categories: detection of neutrons emitted by non-irradiated fissile fuel, gross neutron counting, neutron coincidence counting (subdivided in passive and active detector systems), and multiplicity coincidence counting (IAEA, 2011).

Some specific NDA are foreseen for the measurement of the spent fuel by IAEA: they are listed in the *Table 1.1*, transcript from the IAEA report on “Safeguards Techniques and Equipment” (IAEA, 2011). Some of these key methods are consequently detailed.

CODE	EQUIPMENT NAME	DESCRIPTION/PRIMARY APPLICATION
AEFC	Advanced Experimental Fuel Counter	Characterization of spent fuel from research reactors stored under water.
CBVB	CANDU Bundle Verifier	Verification of the presence of CANDU fuel bundles stored in either stacks or baskets in a spent fuel pond.
CRPS	Cask Radiation Profiling System for dry storage casks	Gross defect device takes radiation profiles from spent fuel storage containers for re-verification
DCVD	Digital Cherenkov Viewing Device	Highly sensitive digital device for viewing Cherenkov light from long cooled, low burnup fuel.
FDET	Fork DETector irradiated fuel measuring system	Detector system that straddles light water reactor fuel assemblies with pairs of neutrons and γ -ray detectors. Gross γ -ray and neutron intensities and ratios of intensities can give specific information on the fuel assembly.
ICVD	Improved Cherenkov Viewing Device	Hand-held light intensifying device optimized to view Cherenkov light (near ultraviolet) in a spent fuel storage pond. System can be used in a lighted area. Primarily used to identify irradiated light water reactor fuel assemblies.
IRAT	Irradiated Fuel Attribute Tester	Gross defect device used for verifying fission product presence in an irradiated fuel assembly.
NGAT	Neutron and Gamma Attribute Tester	Gross defect device used for verifying spent fuel assemblies, fresh MOX fuel assemblies and open or closed containers holding various irradiated and non-irradiated materials including non-fuel items.
PGET	Passive Gamma Emission Tomography	Partial defect device for detecting missing or substituted pins in a light water reactor spent fuel assembly
SFAT	Spent Fuel Attribute Tester	Gross defect device used for verifying the presence of fission products or activation products at the top of the irradiated fuel assembly.
SFCC	Spent Fuel Coincident Counter	Underwater verification of Pu in canned fast breeder reactor spent fuel.
SMOPY	Safeguards MOX PYthon	Gross defect device combines gross neutron counting with low level γ spectroscopy to characterize any kind of spent fuel without movement of spent fuel.

Table 1.1. Spent fuel measurement systems (IAEA, 2011)

The FDET performs both the gamma-ray and the neutrons measurements of the spent fuel, confirming emission of both radiations from the item being measured and with the possibility of verifying the burnup

declaration. The structure of the instrumentation, reported in *Figure 1.1*, is formed by a detector head, an extension pipe several meters long, a gamma-ray and a neutron detector electronics units, and a PC. The detector head is the one responsible for the assessment of the γ -rays, reaching high intensity, and of the neutron counting, able to detect neutrons rate besides the presence of gamma-rays on the background. The fuel assembly to be measured is lifted from the storage position and placed between the FDET detectors arms. The processing of the collected radiation data, integrated with other complementary data, conducts to the characterization of the fuel assembly, i.e., the neutron exposure in the reactor, the initial fissile content, and the irradiation history. In combination with Passive Gamma Emission Tomography (PGET), it can identify partial defects, as fuel pin diversions. The PGET consists in a directionally collimated detector array system, which detects radiations emitted to different directions and produces an image of the data measured. In the image are collected the gamma-emitter concentrations, distributed in the fuel assembly rods. By a visual or computer-based evaluation it is possible to detect the rods absence or substitution.

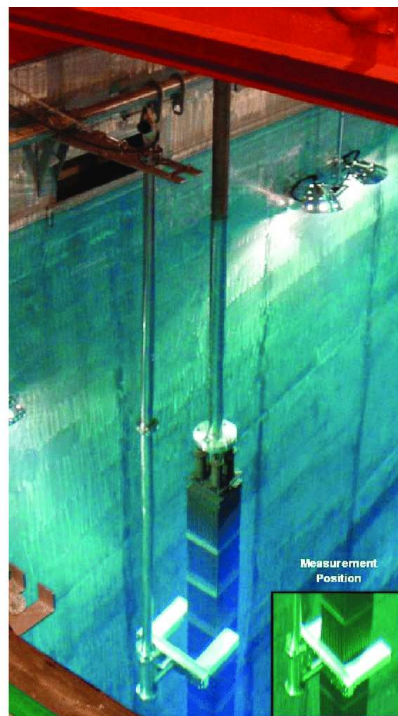


Figure 1.1. Fork DETector irradiated fuel measuring system (FDET) (IAEA, 2011)

A different and innovative technical approach, which relies on the idea of contemporary measurement of neutrons and gamma-ray fluxes, is the one introduced by the Partial DEfect Tester (PDET) (Ham et al., 2015), specifically developed for the detection of fuel pins diversion. This device is further investigated in the next paragraph.

Both IRAT and SFAT systems refer to the class of gamma-ray energy spectral analysis. The IRAT detector is basically utilized to differentiate between non-fuel and fuel irradiated items stored in the spent fuel pool:

it detects γ -ray of both fission products (as ^{137}Cs , ^{134}Cs , ^{144}Pr , ^{154}Eu) contained in the spent fuel and activation products (e.g., ^{60}Co) in irradiated structural materials. SFAT provides a qualitative verification of the presence of spent fuel relying on the detection of gamma-ray produced by certain fission products (^{137}Cs , ^{95}Zr , ^{95}Nb). It is particularly helpful to support Cherenkov viewing devices, in case of weak signal due to low burnup, long cooling time or opaque water in the storage pond (IAEA, 2011).

ICVD and DCVD are tools provided for the detection of the so-called Cherenkov radiation. This is an electromagnetic radiation emitted by a charged particle when it travels through a medium, having a velocity which overcomes the speed of light in that medium. Among the particles emitted from spent fuel, the beta particles (high-energy electrons) are the main responsible of the Cherenkov radiation. This phenomenon leads to a glow surrounding the fuel assemblies in the spent fuel pool, which has the characteristic of lying in the ultraviolet part of the spectrum, as visible in *Figure 1.2*.

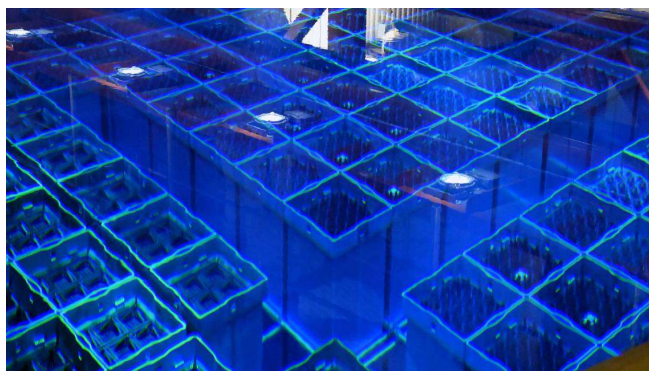


Figure 1.2. A spent fuel pond (source: IAEA)

The related detectors above mentioned are imagine intensifier viewing devices sensitive to the UV radiation in the water bordering spent fuel assemblies. They are able to cut off the spectrum of the visible light, while intensifying the radiation useful for detection aims, later used for qualitative verification, i.e., to distinguish irradiated fuel items from non-fuel items. DCVD has the feature of being used for verification of assemblies having weak Cherenkov signal, typical of low level burnup and/or long cooling time spent fuel storage (IAEA, 2002), (IAEA, 2011).

1.4 NDA technique: PDET

The NDA technique on which this study is based on is the Partial DEFect Tester (PDET). It is an innovative method under study as next generation safeguards for spent fuel verification, specifically for the identification of partial defects. It tries to overcome some of the disadvantages that could be encountered in the identification of partial defects, while using available and approved instruments. For instance, DCVD requires high burnup, short cooling time and transparent spent fuel pool water, whereas FDET is foreseen to verify operators' declarations and requires the movement of the spent fuel assemblies. In this framework, a prototype of the PDET has been designed at the Lawrence Livermore National Laboratory (LLNL) and lately tested in the Central Interim Spent Fuel Storage Facility (CLAB) in Sweden (Ham et al., 2015). The prototype is reported in *Figure 1.3*.

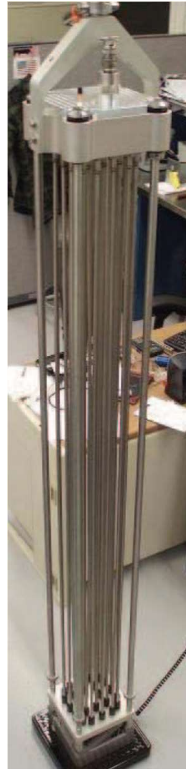


Figure 1.3. Prototype of PDET (Ham et al., 2015)

The PDET prototype has been developed for a PWR fuel assembly: a 17x17 square grid where the majority of the sites contains the fuel rods, and a set of guide tubes are foreseen for the insertion of the control rods. The control rods are in place during the operation of the reactor, lately withdrawn from the spent fuel assembly, thus the guide tubes are filled by water in the spent fuel pool. In the concept of PDET, these guide tubes are reused as containers of gamma and neutron flux detectors for partial defect measurements. Performing analyses on intact spent fuel assemblies, a base signature of the passive gamma and neutron signals is obtained, which depends on the irradiation history of the fuel assemblies (e.g., burnup and cooling time). According to the developers: “The base signature is the arrangement of the signals at each of the guide tube locations normalized to the maximum among them in a particular pattern” (Ham et al., 2015). Having defined this base signature, any fuel diversions can be outlined looking at the distortions between the signature under investigation and the base one. The basic concept is that at wider distortion corresponds greater diversion.

The deployed detectors are high-dose environment fission chambers and ionization chambers. Fission chambers are of ^{235}U type and are sensitive to each individual neutron event, while insensitive to gamma-rays. The ion chambers are instead the chosen devices to measure the gamma-rays. In the prototype, the 24 guide tubes locations were provided by fission and ion chambers equally split: 12 of each type. The PDET assembly is submerged by water during the measurement.

1.5 Introduction to machine learning

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) methods increase their popularity and use on science and technology fields. According to a recent publication of IAEA (IAEA, 2022), this discipline can assist and provide benefits also in nuclear related fields, beyond applications in human health area, optimization in food and agriculture productions other than improvements in environmental responses to climate changes, financial markets, logistics and industries.

In the safeguards framework, the increase of data provided by different verification techniques and activities, on a rising amount of nuclear material under investigation, requires implementation of AI and ML in order to optimize and increase the efficiency of such activities, according to (IAEA, 2022). AI and ML methods are intended to be implemented also in the verification of spent nuclear fuel, in data collection or calibration, and in video surveillance. For instance, the productivity in safeguards activities could enhance if integration of AI and ML methods are foreseen, for decreasing the number of repetitive tasks performed by inspectors. Moreover, the application of AI could support the procedures of classification of data, in particular for the identification of data anomalies also in the spent fuel verification. In fact, ML models have been used for recognizing defect fuel assemblies from complete assemblies, for verifying burnup, cooling time and initial enrichment of nuclear fuel inventory data, and for assessing bias between measured and calculated performances of spent nuclear fuel properties. However, the accuracy of these methodologies are still a concern for the accomplishment of the safeguards purposes, which prevents the utilization of the AI as an autonomous process and requires the human presence in parallel in order to avoid both false alarms and false negatives (IAEA, 2022).

Artificial intelligence concept relies on educating machines to perform tasks traditionally performed by humans, emulating human intelligence. In the AI domain, machine learning is a branch based on data analysis, with particular aim of increasing the efficiency of data processing, comprehension, management and predictions, simulating a kind of human learning process.

1.6 Purpose and structure of the study

The main objective of this work is to apply machine learning methodology for the safeguards verification of the spent fuel assemblies. In detail, the goal is to develop models able to detect spent fuel diversions and to estimate the number and the location of the replaced pins on the fuel assembly, in case a diversion is identified. The machine learning method works on the concept of processing a collection of data, building a model which has learnt from the features of the data provided, and finally using this model to predict the outcomes of a dataset having similar features of the training one. These three aspects have been suitably chosen for the purpose of this study.

The collection of data used as input of the models are the one provided by a set of Monte Carlo simulations, produced to create a database of an NDA detector responses (the PDET detector, in this case). The simulations have been performed to determine the detectors responses for both intact and diverted scenarios. The responses have been employed in several different combinations in the input phase, with the intention of subsequently investigate the dependences of the results on the imported features.

The machine learning methods used for this study were based on regression models, in detail two main types have been investigated: the k-nearest neighbors and the neural networks algorithms. The coding part of the work has been conducted with Python tools such as scikit-learn and TensorFlow & Keras, respectively for the k-nearest neighbors and the neural networks implementation.

From the models constructed, outcomes of interest for safeguards verification have been designated. In particular, based on the collection of data above mentioned, the number of diverted pins and the location of these diverted pins in the fuel assembly have been predicted, using both methods. These outputs have been further treated with the aim of compare different models, different features and different pre-processing methods of the dataset.

The current chapter was dedicated to an introduction on nuclear safeguards frameworks, spent nuclear fuel and related measurement techniques, with particular focus on the one used in this study, plus a briefly introduction on machine learning applications. Next, along this thesis, some of these topics are explored more in details. In the second chapter the database of simulated detectors responses based on the developed PDET prototype is going to be described, and a preliminary analysis of the dataset is provided. In the third chapter the machine learning methods used in this work are introduced, followed by the configuration of the prediction models, tailored on the outcomes of interest of each model. In the fourth chapter the most promising results are reported, whereas the fifth one is dedicated to the conclusions.

2. DESCRIPTION OF THE DATABASE

2.1 Simulated detector responses

Starting from the concept of the PDET device, a database of simulated detectors responses has been configured and built at SCK CEN. The model of the device used as reference for the calculations is shown in *Figure 2.1*. Two groups of detectors are placed in the guide tubes locations (9 centrals in red, and 16 peripherals in yellow), and a third group is added on top of the ones foreseen by the original prototype (40 externals in green). The remaining locations on the fuel assembly are occupied by fuel pins. Thus, in the 17x17 geometry arrangement, a total amount of 264 fuel pins and 25 detectors are included, while other 40 detectors are within the surrounding area (Rossa, 2019).

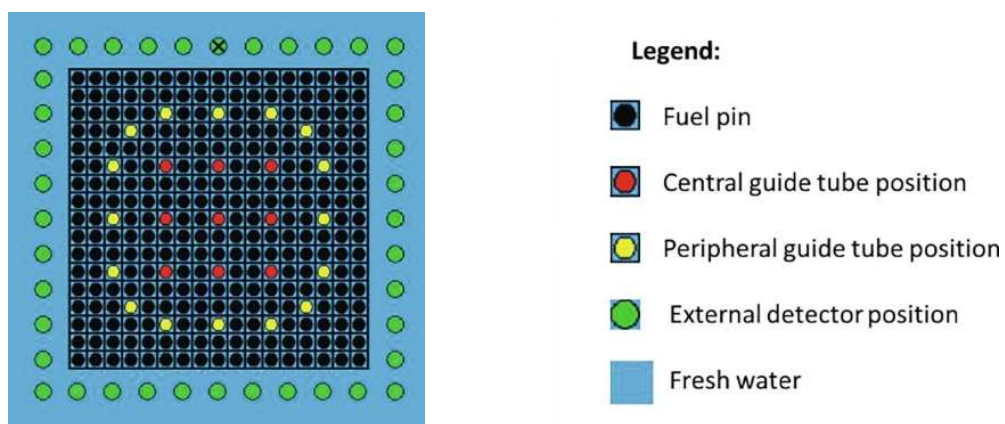


Figure 2.1. Model setup for PDET analysis (Rossa, 2019)

The detectors responses derived from the results of the Monte Carlo simulations (Rossa, 2021), and this study in particular includes three kinds of detectors, sensitive to:

- Thermal neutrons (TH): responses of bare ^{235}U fission chamber. The responses are in counts per second;
- Fast neutrons (FA): responses of bare ^{238}U fission chamber. The responses are in counts per second;
- Gamma-ray (P): responses of ionization chamber. The responses are proportional to the current of the ionization chamber.

The detectors responses are the bases of the input features used in the machine learning models under investigation.

2.2 Database used in the study

The guiding principle for the construction of the database was to model, and therefore analyze, some of the possible diversion scenarios that a potential proliferator could operate on spent fuel assemblies. In this perspective, the simulations for the detectors responses were intended to emulate both complete and

defect fuel assemblies scenarios. The simulations were therefore carried out by varying different properties of the spent fuel, to cover a wide range of situations. The version of the database used for this study has been expanded in the following years with respect to the one proposed in (Rossa, 2019).

The first part of the database contains the Monte Carlo simulations performed for the complete fuel assemblies scenarios. As complete scenarios are identified those where all fuel pins are present, having equal material composition and source strength, whose data were taken from SCK CEN reference spent fuel library (Rossa et al., 2013). The 1372 complete scenarios were parametrized by the following values:

- Initial Enrichment (IE): 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0 %
- Burnup (BU): 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70 GWd/t_{HM}
- Cooling time (CT): 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100 years

The wide range of parameters examined was driven by the intention of investigate the effect of irradiation history (namely, IE, BU and CT) on the responses identified by the detectors.

The remaining part of the dataset covers the defect scenarios simulations, i.e., 963 possible diversions where a certain number of fuel pins are replaced by so-called dummy pins, composed of stainless steel. The intact part of the fuel assemblies still refers to the material composition and source strength of the complete scenarios, whereas the dummy pins have no source terms. In this phase, the parameters considering the radiation history were reduced to:

- Initial Enrichment (IE): 2.0, 3.5, 5.0 %
- Burnup (BU): 10, 30, 60 GWd/t_{HM}
- Cooling time (CT): 5 years

For each of the combination of the above parameters, different amount of replaced pins (from 4 to 180) were introduced, and different geometries for the location of the replaced pins were configured. Some examples are reported in *Table 2.1*: the fuel pins are colored in white, dummy pins in grey and guide tube positions are marked with crosses. Each scenario is reported with the correspondent number of replaced pins.

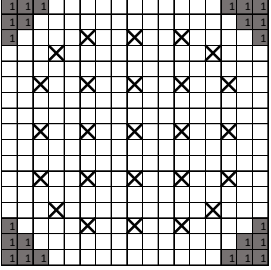
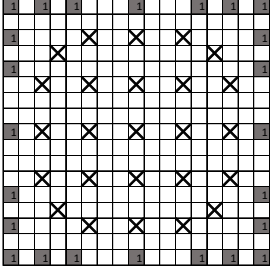
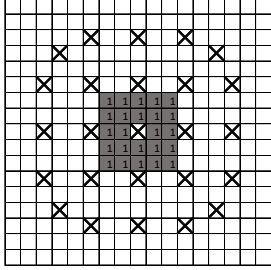
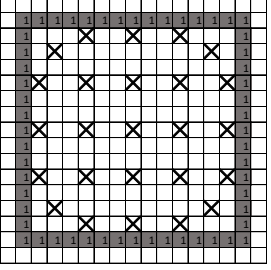
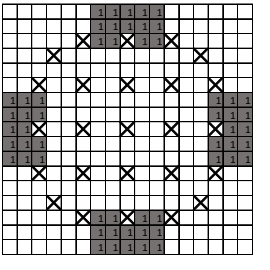
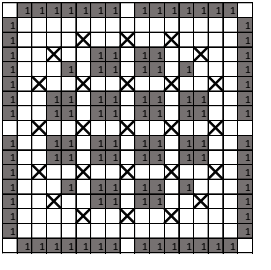
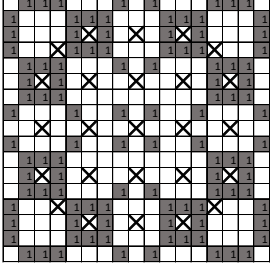
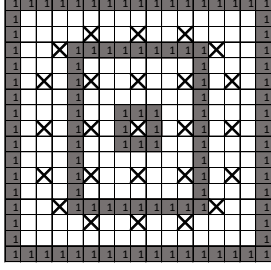
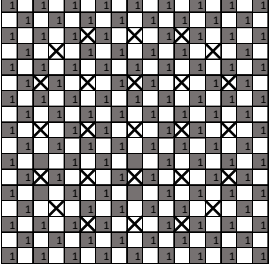
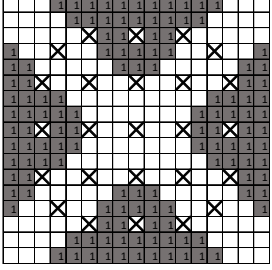
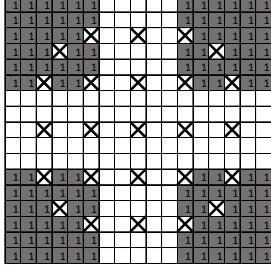
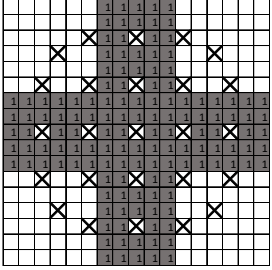
<p>Number of Dummy Pins: 24</p>	<p>Diversion 02401</p> 	<p>Diversion 02404</p> 	<p>Diversion 02407</p> 
<p>Number of Dummy Pins: 56</p>	<p>Diversion 05601</p> 	<p>Diversion 05602</p> 	
<p>Number of Dummy Pins: 108</p>	<p>Diversion 10803</p> 	<p>Diversion 10805</p> 	<p>Diversion 10806</p> 
<p>Number of Dummy Pins: 128</p>	<p>Diversion 12801</p> 	<p>Diversion 12802</p> 	<p>Diversion 12803</p> 
<p>Number of Dummy Pins: 136</p>	<p>Diversion 13601</p> 		

Table 2.1. Examples of defect scenarios configurations

Some extracts from the dataset developed for the detectors responses of the PDET are reported in *Table 2.2*. To clarify the notation used, an explanation of the content of each column is provided as follows:

- File ID: Filename of the simulation. For the complete fuel scenarios, the filename starts with “R” and is consequently formed by the correspondent IE, BU and CT of the fuel assembly. For the defect fuel scenarios, the identification starts with “D” and is followed by the pattern of the diversion scenario (i.e., number of dummy pins and geometry configuration of the diversion);
- IE, BU, CT: as previously described;
- DummyPins: number of replaced pins in the fuel assembly;
- TH: responses of the fission chamber for the thermal neutrons, reported in counts per second. Namely, there is a column for each of the 65 locations of the detectors;
- FA: responses of the fission chamber for the fast neutrons, reported in counts per second. Namely, there is a column for each of the 65 locations of the detectors;
- P: responses of the ionization chamber, values proportional to the current of the ionization chamber, reported in 10^6 nA. Namely, there is a column for each of the 65 locations of the detectors;
- Pin: 264 columns referring to the presence (“0”) or replacement (“1”) of each individual fuel pin. Pin locations on the fuel assembly are numbered from the top left corner to the bottom right one.

FileID	IE	BU	CT	DummyPins	TH01	TH02	FA37	FA38	P64	P65	Pin159	Pin160
R200050101	2.0	5	1	0	0.0739	0.0775	0.0019	0.0025	5.58E+05	4.30E+05	0	0
R300300121	3.0	30	100	0	0.7396	0.7639	0.0229	0.0291	2.73E+04	2.11E+04	0	0
R400250101	4.0	25	1	0	5.2456	5.4396	0.2019	0.2566	1.56E+06	1.20E+06	0	0
R400700121	4.0	70	100	0	6.5001	6.6746	0.2114	0.2699	6.21E+04	4.81E+04	0	0
R450050101	4.5	5	1	0	0.0355	0.0373	0.0015	0.0020	5.53E+05	4.26E+05	0	0
D00401	2	10	5	4	0.3203	0.3319	0.0084	0.0109	1.26E+05	9.73E+04	0	0
D04802	5	30	5	48	3.2880	3.1046	0.1221	0.1544	4.26E+05	3.30E+05	0	0
D06401	2	30	5	64	17.5810	18.6659	0.3866	0.5161	3.00E+05	2.29E+05	0	0
D10802	2	10	5	108	0.1391	0.1456	0.0028	0.0037	6.24E+04	5.11E+04	0	1
D18001	5	60	5	180	17.0395	16.2752	0.5465	0.6551	7.43E+05	5.83E+05	1	1

Table 2.2. Extract from PDET detectors responses dataset

2.3 Visualization of the database

In order to have a comprehensive vision of the data provided in the database, first a preliminary analysis was conducted by plotting the detectors responses for all the observations in the database, with respect to different radiation history parameters.

In the first place, all neutrons detectors responses were considered for a measurement time of 10 minutes: any observation, in counts per second, was multiplied by 600 seconds. For each processed observation the relative uncertainty was hence introduced. Concerning the values of thermal and fast neutrons counts, the relative uncertainties were calculated as the squared root of the respective values, whereas for the data related to the gamma-ray an arbitrary uncertainty of 2% was added to each value. Thus, the detectors were separated in three groups: central, peripheral, and external, to make an independent analysis on the base of location common similarities. Considering one group of detectors at a time, the sum and the

average of the values for each kind of detector within each group were computed. The relative uncertainty for each new retrieved data was also calculated with propagation of errors formula for both the sum and the average approaches. Comparisons among the scenarios were afterwards investigated, based on observation data and uncertainties. Since similar results were obtained for the analyses performed, some of the results are presented in this section, while others are attached in the [Appendix A](#) for additionally details.

As examples, the graphs produced for the average values of the peripheral detector responses are here reported with the relative uncertainties. In the first instance, comparisons among the complete and defect scenarios were performed, considering thermal neutrons counts and photons current (*Figure 2.2*), fast neutrons counts and photons current (*Figure 2.3*), and thermal and fast neutrons counts (*Figure 2.4*). It shall be noticed that the values of thermal and fast neutrons counts are reported for a measurement time of 600 seconds.

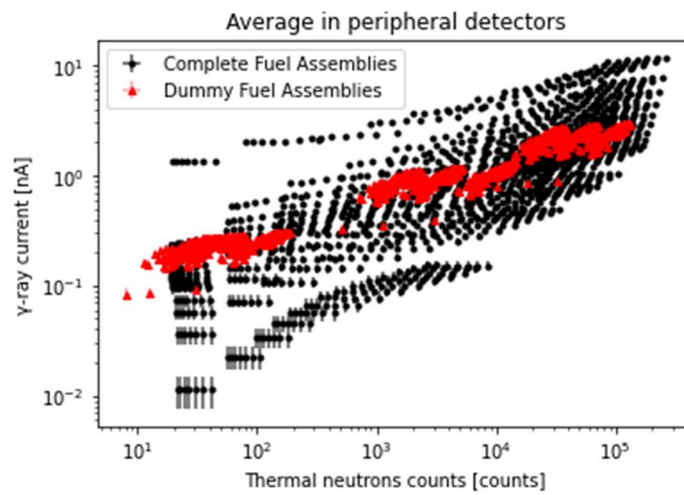


Figure 2.2. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Thermal neutrons counts and photons current

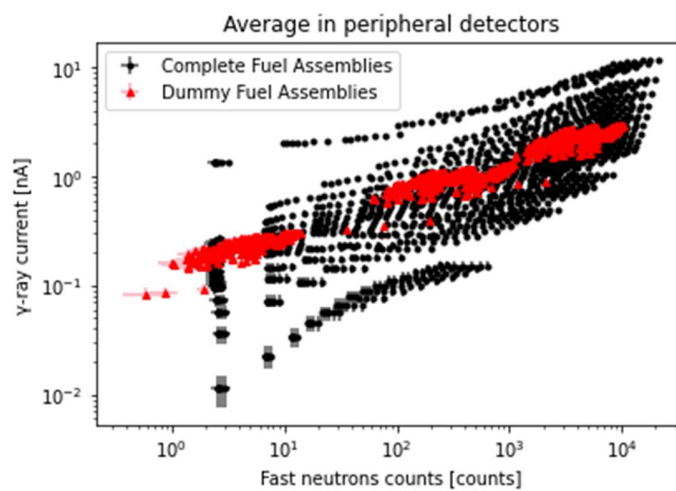


Figure 2.3. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Fast neutrons counts and photons current

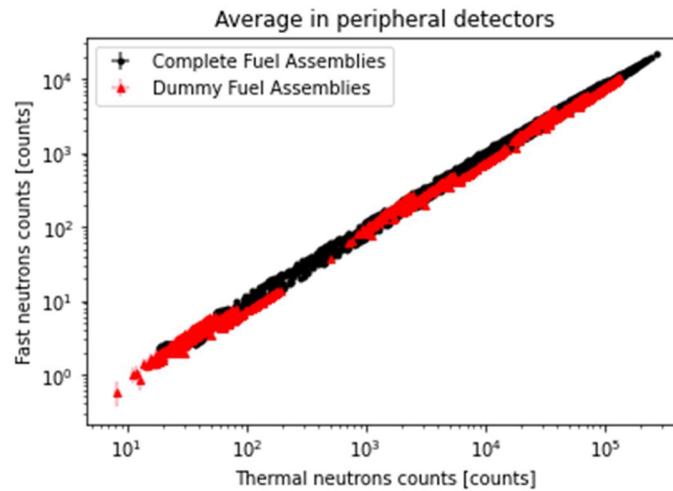


Figure 2.4. Database visualization: Complete and defect scenarios - Average in peripheral detectors and relative uncertainties – Thermal and fast neutrons counts

The general information inferred from the first part of the plotting of the dataset is that defect scenarios locate indicatively in the midpoint of the complete scenarios values, due to the choice of the radiation history parameters for the development of the scenarios. Similar trends were observed considering the sum or the average values of the detectors responses. Therefore, the classification between complete and defect assembly seems not possible by simply placing threshold values on the detector responses.

To further examine how the detectors responses related to the defect scenarios respond to the considered parameters, e.g., the number of replaced pins, the initial enrichment and the burnup of each configuration, two further analyses have been developed, focusing only on defect scenarios. First, an investigation of the data with respect to the actual number of dummy pins have been performed, and second a comparison among the different radiation history parameters, examined at fixed cooling time of 5 years. For each combination of detectors responses reported above, these investigations are presented in the *Figures 2.5-2.10*, considering the average values of peripheral detectors. Also in this analysis, the values of the neutrons detectors responses are considered for a measurement time of 600 seconds.

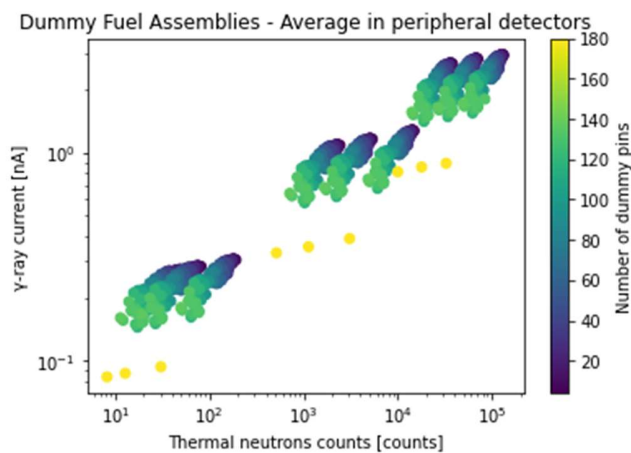


Figure 2.5. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal neutrons counts and photons current - Number of dummy pins

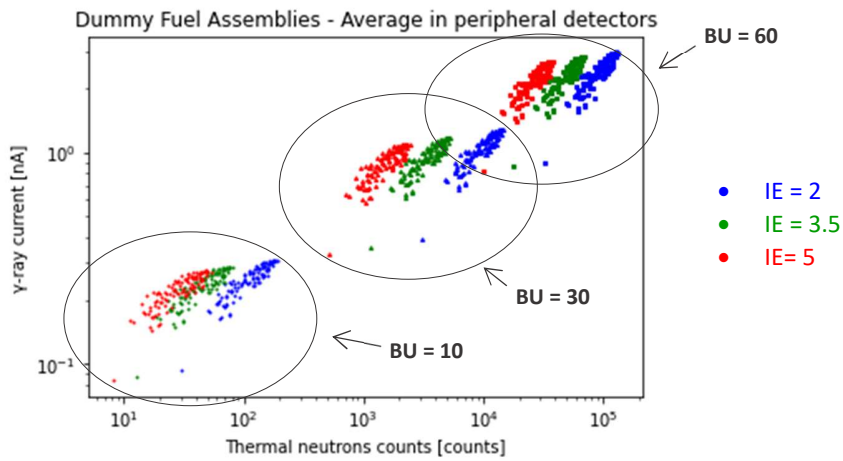


Figure 2.6. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal neutrons counts and photons current - Initial Enrichment and BurnUp

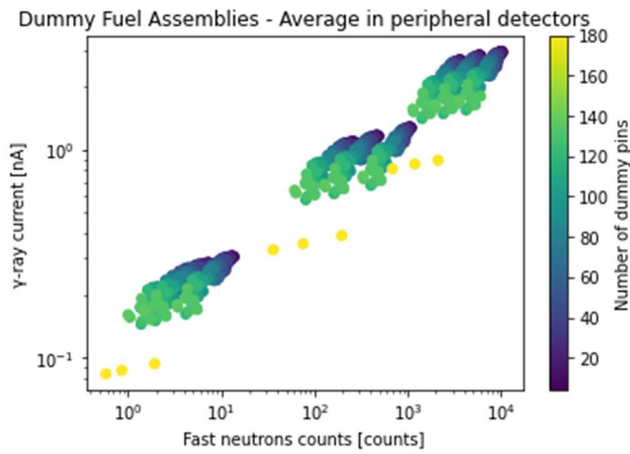


Figure 2.7. Database visualization: Defect scenarios - Average in peripheral detectors - Fast neutrons counts and photons current - Number of dummy pins

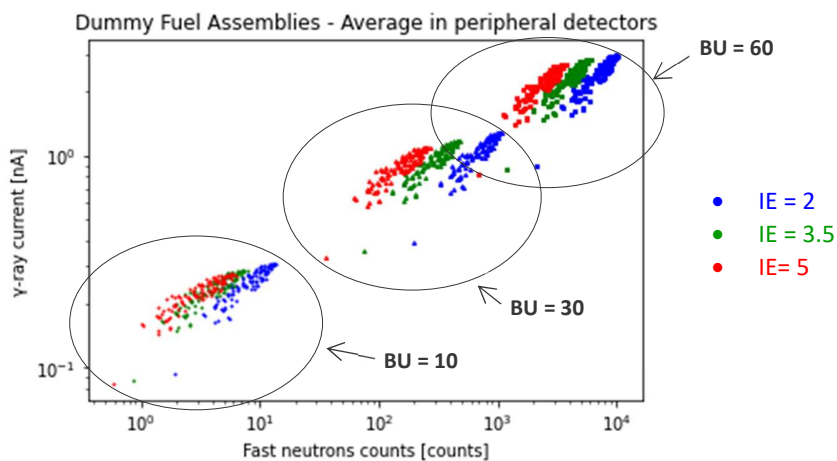


Figure 2.8. Database visualization: Defect scenarios - Average in peripheral detectors - Fast neutrons counts and photons current - Initial Enrichment and BurnUp

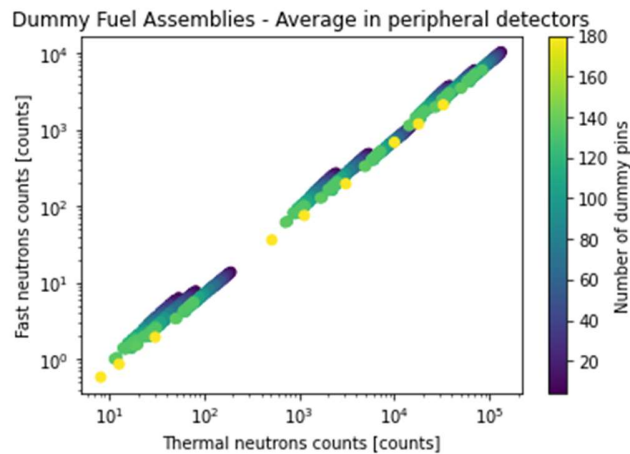


Figure 2.9. Database visualization: Defect scenarios - Average in peripheral detectors - Thermal and fast neutrons counts - Number of dummy pins

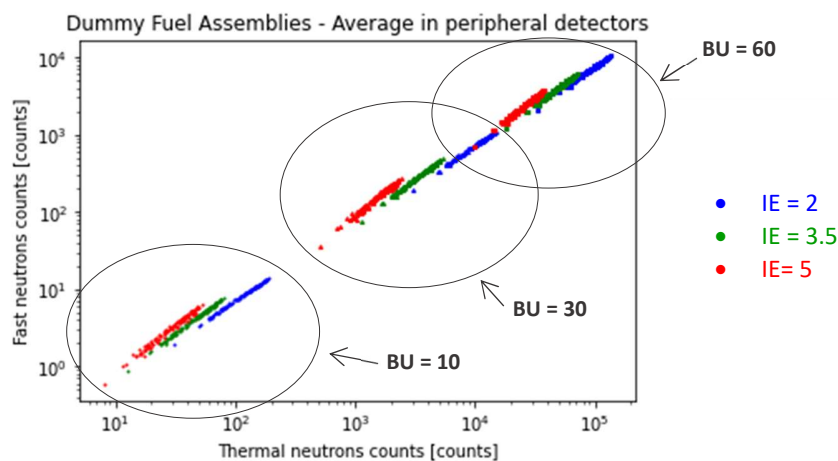


Figure 2.10. Database visualization: Defect scenarios - Average in peripheral detectors – Thermal and fast neutrons counts - Initial Enrichment and BurnUp

The responses of the detectors follow similar trends among the different comparisons. Regarding the analysis which considers the number of replaced pins (Figure 2.5, Figure 2.7, Figure 2.9), on the same conditions, at higher number of dummy pins corresponds lower counting rate: the diversion of the spent fuel pins induce a lower amount of fission products and depleted fuel in the assembly, thus a lower value of radioactivity present.

Looking into details at the radiation history of each observation, comparisons among detectors responses of fuel assemblies with different values of initial enrichment and burnup can be carried out (Figure 2.6, Figure 2.8, Figure 2.10). At the same level of burnup, the neutrons emission is greater at lower values of initial enrichment. Indeed, the burnup is also a measure of the fuel depletion and depends on the density of the fissile material present (namely, ^{235}U). To reach the same level of burnup in a given time, in a situation of lower level of enrichment, the fuel must be exposed to a larger neutron fluency rate, which leads to a larger production of neutrons emitters (in particular ^{242}Cm and ^{244}Cm), hence also to a greater spent fuel neutrons emission. On the other hand, fixing the level of initial enrichment, the burnup seems

to contribute more to the detected responses, since higher level of burnup and depletion of the fuel leads to a major production of fission products and correspondent radioactive decaying processes, driving to an higher neutron and gamma-ray emission. Similar trends were observed for neutron and gamma-ray detectors.

Comparisons among complete and defect scenarios were further performed on the average and the Relative Standard Deviation (RSD) values of the detectors responses among each group, with respect to the number of dummy pins in each observation. In the *Figures 2.11, 2.12 and 2.13* the analyses based on thermal neutrons counts, fast neutrons counts and photons current are reported, respectively.

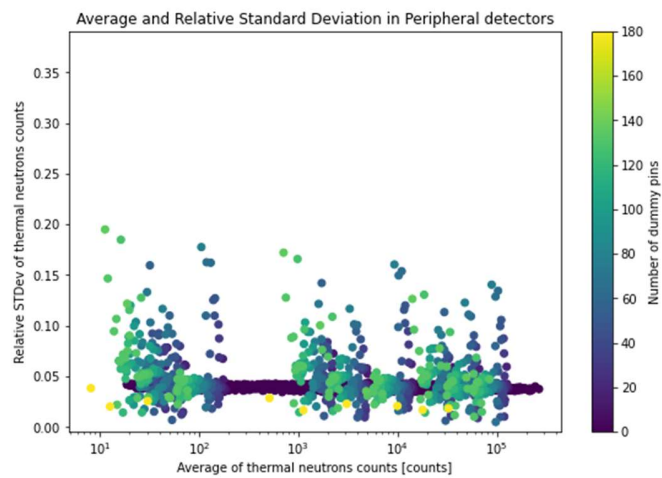


Figure 2.11. Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Thermal neutrons counts - Number of dummy pins

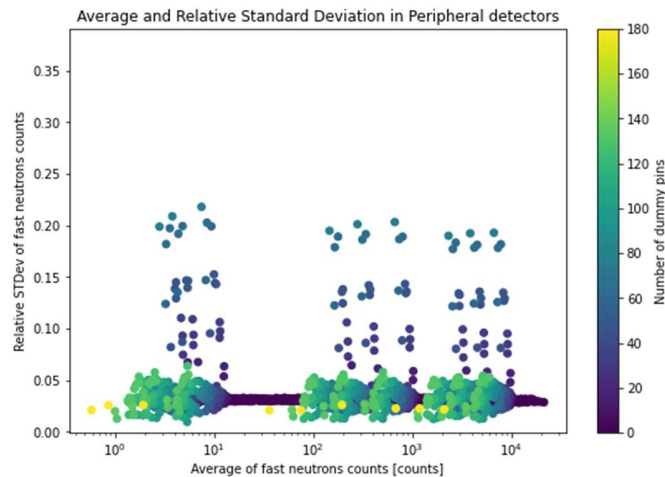


Figure 2.12. Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Fast neutrons counts - Number of dummy pins

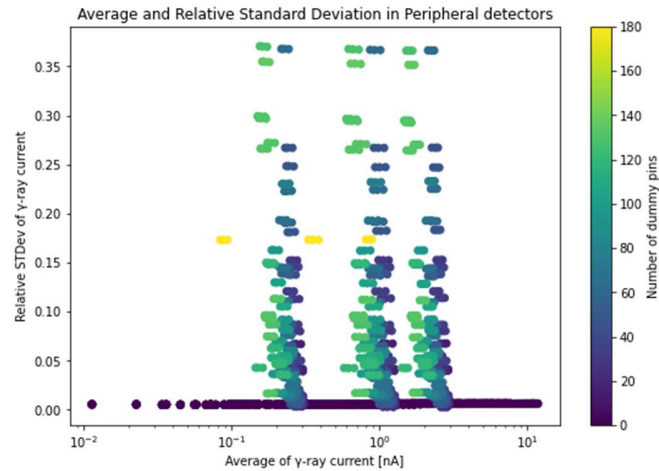


Figure 2.13 Database visualization: Complete and defect scenarios - Average and relative standard deviation in peripheral detectors - Photons current - Number of dummy pins

The analysis allows to identify the diversions of the fuel assembly pins as actual deviations of the detectors responses with respect to the complete scenarios. On the plot, the latter tend to concentrate on a constant value of relative standard deviation, while the observations pertaining to the defect scenarios are visible as clusters of dispersed data around that more compact trend. The dispersion appears to be more evident in the case of photons current values, which is an indicator of the higher sensitivity of the gamma-ray responses on the diversion scenarios, with respect to thermal and fast neutrons responses. This larger sensitivity to local defects is related to the different mean free paths of the detected particles, indeed gamma-rays have a lower mean free path with respect to neutrons, thus the responses show larger sensitivity. This characteristic has been further investigated and employed in the models developed for the machine learning analyses of the database.

Focusing on the values of the relative standard deviation of photons current, for peripheral detectors locations, with respect to the number of dummy pins of complete and defect scenarios, it can be obtained the plot reported in *Figure 2.14*. The discrepancies of values of the RSD within complete and defect scenarios are evident also in this visual configuration of the parameters. In detail, all the complete scenarios, which refer to a number of dummy pins equal to zero, superimpose on the same value of RSD, while the observations related to the defect scenarios, carrying higher number of replaced pins, lead in most cases to higher and more dispersed values of relative standard deviation. Even in this case the sensitivity of the γ -ray detectors on the diversions of the fuel assemblies can be inferred.

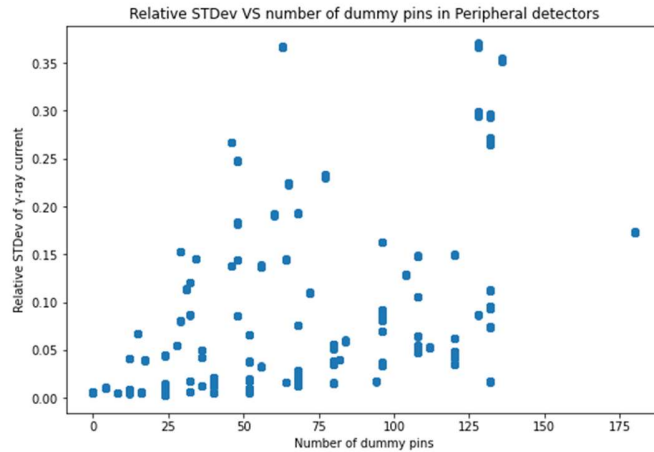


Figure 2.14. Database visualization: Complete and defect scenarios – Relative standard deviation of photons current in peripheral detectors - Number of dummy pins

A further investigation has been performed starting from the above reported results, focusing only on the complete scenarios and the peripheral detectors location. To explain the reasons of the occurrence of a non-null relative standard deviation identified in the complete assembly scenarios, it has been considered the case of four symmetrical detectors in the peripheral perimeter, in comparison to all peripheral detectors (i.e., 16). They are represented respectively as blue and yellow dots in *Figure 2.15* (for thermal neutrons counts), *Figure 2.16* (for fast neutrons counts) and *Figure 2.17* (for photons current). The relative standard deviation, when considering all the 16 peripheral detectors, appears to be higher in value with respect to the one calculated in the case of the symmetric detectors only. The discrepancy may be related to the larger dispersion of the data introduced by the geometrical distribution of the detectors on the perimeter, which faces some asymmetries due to the location of the detectors. The relative standard deviation for the four symmetric detectors is related to the uncertainties introduced by the Monte Carlo simulations performed for the construction of the database.

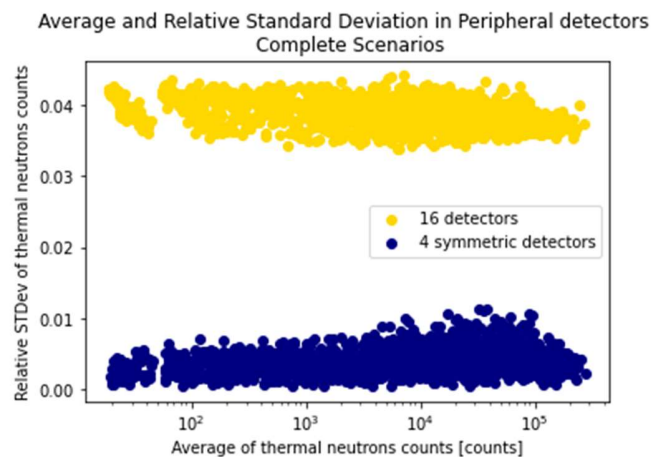


Figure 2.15. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors – Thermal neutrons counts – All peripheral detectors and 4 symmetric peripheral detectors

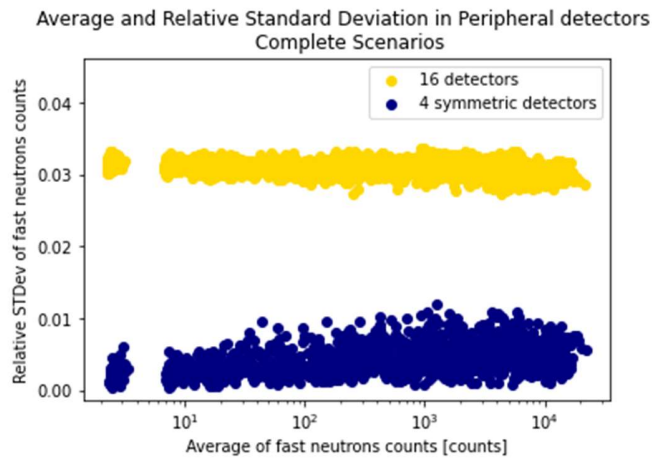


Figure 2.16. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors – Fast neutrons counts – All peripheral detectors and 4 symmetric peripheral detectors

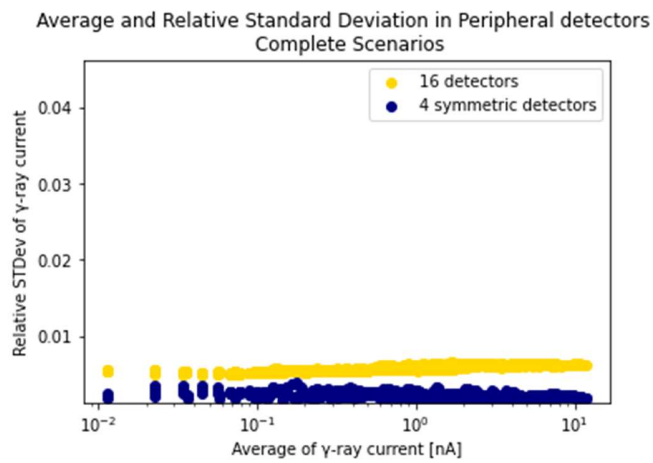


Figure 2.17. Database visualization: Complete scenarios - Average and relative standard deviation in peripheral detectors - Photons current – All peripheral detectors and 4 symmetric peripheral detectors

3. MACHINE LEARNING

Machine Learning (ML) techniques imply the use of algorithms suitably generated that, starting from a collection of data, improves the learning process itself and learns with a view to treat a random sample, similar to the provided data, without being explicitly programmed to predict the outcome of that particular sample from the beginning. First of all, ML methods are divided in two broad categories: supervised and unsupervised. In the supervised approach, both inputs and outputs of a dataset are known and labeled, whereas in case of unsupervised approach the outputs associated to the inputs are unlabeled. Considering the database described in the previous chapter, the data allow the use of supervised models for this study. A generic scheme of a supervised machine learning procedure is shown in *Figure 3.1* (Rossa, 2019).

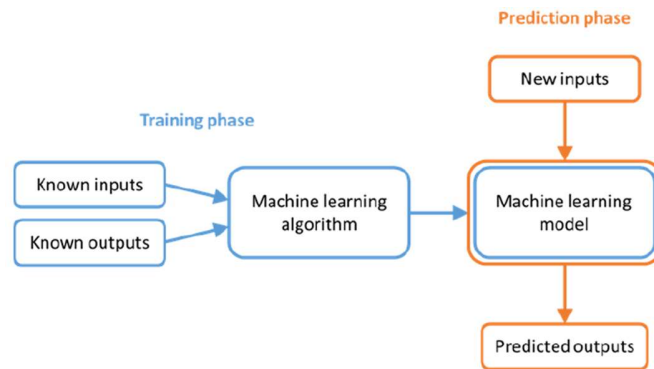


Figure 3.1. Supervised machine learning method workflow (Rossa, 2019)

In the training phase the known inputs (features) and known outputs (responses) are provided as the training set of observations to the ML algorithm developed, which is trained on this set with the aim of making it able to predict unknown outputs when new input data are supplied to the model, which occurs in the second phase, called prediction phase.

In the supervised learning framework, an additional subdivision between classification and regression methods is introduced, based on the type of responses which is foreseen in the analysis.

In the classification approach the responses are categorized, to identify if they belong or not to a given class. In case the number of classes is equal to two, the method is called binary classification, for instance to identify if a fuel pin is present/absent in a definite location. On the other hand, if a higher number of classes is foreseen, it is called multiclass classification (Murphy, 2012). For example, a multiclass classification could consider few response classes, based on the percentage of removed pins from a spent fuel assembly, as investigated in (Giani, 2019), (Rossa et al., 2020).

Regression methods have the aim to predict response variables which assume continuous values, as the exact number of replaced pins in a diverted spent fuel assembly scenario. The purpose of this study was based on the application of this second type of supervised machine learning method, as a complementary development of the previous analysis performed in (Giani, 2019).

Either for classification or for regression methods several algorithms, implying different approaches and performances, have been developed: Support Vector Machines, Linear Models, k-Nearest Neighbors, Decision Trees, Discriminant Analysis, Neural Networks are few of them. In this study, as regression methods were considered k-nearest neighbors and neural networks.

As a summary of the working procedure for a machine learning model, it is reported the scheme in *Figure 3.2*.

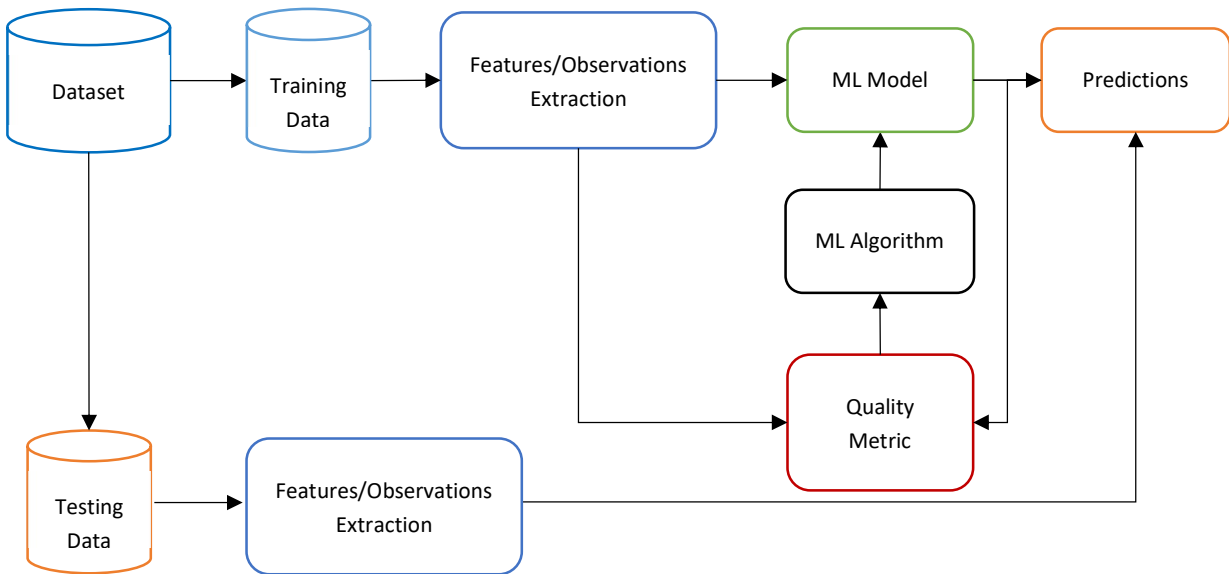


Figure 3.2. Machine learning working process

The starting point is the complete dataset, which is then divided in training and testing data, introduced at different stages in the ML process. From the subsets are extracted the observations related to certain selected features of interest (for example, a certain kind of detector), which are the same for both subsets. The training set is used to train the model in two different steps: part of it is first destined to the training of the ML model that is going to be subsequently employed for the predictions, and part is supplied to the (figurative) Quality Metric block in the diagram. The latter represents the quantification, and the further optimization, of the error introduced by the model itself, indeed the data here involved are used to assess the goodness of the parameters utilized in the algorithm to predict new data, and to optimize these parameters to produce a more refined prediction model. At the end of this process, the inputs pertaining to the testing subset are supplied to the trained ML model, which performs the expected predictions. The outputs in the testing set are instead used for the estimations of the error, i.e., the discrepancies among the predicted values and the true values.

3.1 k-Nearest Neighbors regression model

The first supervised machine learning method investigated in this work is the k Nearest Neighbors (k-NN) regression algorithm, developed using the scikit-learn library implemented in Python. In *Figure 3.3* is reported an illustrative scheme for a classification problem, for the sake of more straightforward

visualization of the working principle. The method is based on the concept of exploiting the similarities among data samples. Precisely, once the training data have been memorized by the model, the label of a new point (the black square in the *Figure 3.3*) is predicted considering a defined number of training samples which are the closest, in distance, to the query point. k-nearest objects are considered for the prediction of the label of the query point, where the number of neighbors k is arbitrarily defined. In the reported example, considering k equal to 1 (inside the inner line circle) or equal to 8 (inside the outer line circle) leads to different prediction labels of the query point, if equal importance to the k-nearest samples is given. Particular attention has to be given to the value of k, since an insufficient number of neighbors may lead to a lack of exhaustiveness of information, whereas an excessive number could be responsible for misperception due to a crowded neighborhood. Moreover, the other parameter that has to be defined is the weight attributed to each k-neighbor point, i.e., how much each neighbor contributes to the prediction. The k-nearest neighbors can either contribute uniformly to the prediction, or proportionally to the inverse of the distance with respect to the query point, i.e., closer points provide larger contribution to the prediction. For the regression model, the label assigned to the target point is computed as the mean of its nearest neighbors, and assigned to the query as a continuous variable (Pedregosa et al., 2011).

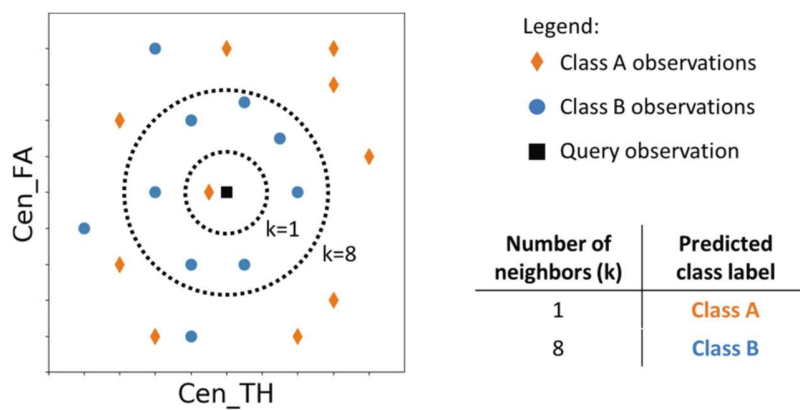


Figure 3.3. k-NN classification model scheme (Rossa et al., 2020)

Configuration of the models

Python was the environment adopted for the development of the models, specifically it has been taken advantage of the scikit-learn library for the implementation of the k-NN regression models. The general procedure for the development of the k-NN regression models used in this study can be subdivided in the following steps:

1. Import of the dataset described in [Section 2.2](#), and division of the samples in training and testing data. The training dataset was used to train the model: it contains both inputs and outputs of a certain number of observations. The testing dataset contains the remaining part of the observations, where inputs were used to predict the responses, based on the model previously trained, and the outputs were used for measuring the correctness of the prediction model.
2. Select the features of interest from the input database.

3. Pre-process the data: it is a common requirement for learning algorithm to standardize, normalize, or scale the data if different unit of measures and/or wide differences in scales among features are present.
4. Choose the number of k-nearest neighbors and the weight to assign to each neighbor. Given that these parameters depend on the data under analysis, the selection was made using an hyperparameter optimization model: the algorithm is able to choose autonomously the best parameters which optimize the learning model. Here, the GridSearchCV approach has been consider, implemented in scikit-learn library, that exhaustively considers all the parameter combinations, which in this case were a number of k ranging between 1 and 50 and the setting of the weight either “uniform” or “distance”. The evaluation metric adopted for the optimization of the parameters is the default scoring function used by the GridSearchCV method, namely the R^2 score: briefly, it estimates how well unknown samples are likely to be predicted by the model. The distance metric, i.e., the method to calculate the distance between two points, was set to Euclidean (the length of the line segment connecting the two points).
5. Fit the k-nearest neighbors regressor on the training dataset, using the best parameters retrieved from point 4, tuned on the input features introduced in point 2.
6. Predict the outputs associated to the input variables of the test dataset, by mean of the fitted k-NN model.
7. Evaluate the goodness of the model, by means of different error metrics performed on the discrepancies among predicted and actual values.
8. Validate the model: run the algorithm several times with the intention of increasing the randomness of the observations stored in train and test datasets.

This process has been followed for the prediction of two different outcomes, based on the same dataset: the number of dummy pins associated to a certain scenario and the locations of the dummy pins in a certain geometry configuration. Due to the different outcomes under examination the steps in the illustrated procedure have been adjusted consequently.

3.1.1 Number of dummy pins

In this section of the work, the k-NN ML method has been applied for the prediction of the number of dummy pins in the scenarios introduced by the described dataset. Different combinations of kinds of detectors, groups of detectors and pre-processing operations on the values were considered as input features in the training phase of the learning process. Eventually, the performances of each trained model, based on different input features, has been compared in order to identify the best approach for the implementation of the latter.

As introduced in the first point of the procedure description, the dataset has been split in training and testing data. One of the common practices in machine learning algorithms is to keep a proportion of 80% train – 20% test in the subdivision of the investigated database (Hastie, 2017). In this case, the subdivision was strictly correlated to the validation scheme, which has been implemented adopting a 5-fold Cross Validation (CV) scheme (Hastie, 2017), in accordance to the percentage of data pertaining to each subset.

The total number of models investigated in this phase was equal to 147, where each machine learning model relied on different input features, selected as a combination of the following characteristics:

- The kind of detectors: namely TH, FA, P and all the association among the latter, i.e., TH&P, FA&P, TH&FA, TH&FA&P;
- For each of the combination above mentioned, the group to which belongs the kinds of detectors was then selected, namely central (Cen), peripheral (Per) or external (Ext). As for the previous point, also all the associations of different groups of detectors were investigated, i.e., Cen&Per, Cen&Ext, Per&Ext, Cen&Per&Ext;
- Within each group of detectors (Cen, Per, Ext), some operations on values were introduced before supplying the data to the ML model. Specifically, three separated operations were performed: the sum within the values (Sum), the average and standard deviation (Ave&Std) within the values, or no operation introduced, i.e., taking the values of the detectors responses as they were presented in the database (Sin). Each operation recalled was applied to each combination of kinds and groups of detectors above mentioned.

Once the input features were selected, the consecutive training and test data have been additionally pre-processed to fulfil the requirement of having standardized, normalized, or scaled data in input of the ML models. Standardization of the data implies the reshaping of their original distribution in order to obtain a standard normally distribution: Gaussian with zero mean and unit variance. Instead, in the normalization process the vector containing the values of each observation (in this case, the scenario) is rescaled to have unit norm (i.e., the square root of the sum of each squared element in the vector is equal to 1), independently from the original distribution of the observations. On the other hand, features can be scaled in order to lie between a given minimum or maximum value, which is often set between zero and one to fulfill the requirement of some functions employed in specific ML methods (Pedregosa et al., 2011). In this case, it has been chosen to standardize the data, since marginally better results were envisaged by some trials and errors.

Afterwards, the algorithms have used the standardized training dataset to tune and select the best parameters, fit the models, and predict the outcomes from the inputs of the standardized testing dataset.

The outcomes of the k-NN models, i.e., the number of dummy pins for each of the scenarios pertaining to the testing dataset, were used to evaluate the goodness of the models by means of two different evaluation metrics. The first one was the arithmetic difference between the predicted and the real number of dummy pins, called Delta (Δ) henceforth. The second one was the so-called Root Mean Squared Error (RMSE), which is one of the frequently used machine learning evaluation metrics for regression model. The related formula is reported in *Equation (3.1)*:

$$RMSE (y, \hat{y}) = \sqrt{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2} \quad (3.1)$$

where \hat{y}_i and y_i are respectively the predicted and correspondent real value of the number of dummy pins, for the i -th sample. The square errors among all samples belonging to the testing dataset are averaged on the $n_{samples}$ of the subset. This metric provides an accuracy of the model and permits to compare the predictions errors of different models, when referring to the same database, as in this case. Moreover, the RMSE is on the same scale of the data, where lower value implies more accurate estimations (Hyndman, 2006).

Eventually, the process for the implementation of the CV scheme can be summarized in the following steps:

- A flag (a number from 1 to 5) was randomly assigned to each observation (scenario) in the original dataset, thus it was divided in 5 folds.
- All the scenarios having a flag = “1” were stored in the testing subset, while the remaining scenarios pertained to the training subset.
- The k-NN regression model under investigation was trained on the training subset and evaluated on the testing subset. The hyperparameter optimization algorithm was also contextually tuned on the training subset. The evaluation parameters, RMSE and Δ , were computed and stored.
- The last two points were repeated, changing the observations to be introduced in the testing subset, i.e., incrementing the flag number until reaching a flag = “5”. In this way, the 5 testing subsets, which corresponded to the 5 folds generated at the beginning, covered all the scenarios constituting the dataset.
- To increase the randomness of the input data, the previous points were repeated 5 times, changing the random assignment between flags, i.e., folds, and observations. Eventually, each learning algorithm was performed 25 times, and 25 RMSE values were retrieved and used to calculate their mean value and the correspondent standard deviation. The Δ values were instead stored only in the first 5 turns, as an illustrative example of the difference between predicted and real values of the number of dummy pins for all the scenarios constituting the database.

3.1.2 Location of dummy pins

In this section, the implementation of the k-NN machine learning algorithms for the prediction of the locations of the replaced pins in the fuel assembly configuration is outlined. The general approach is similar to the one presented for the prediction of the number of replaced pins, besides some procedures, primarily the selection of the input features, were changed to consider mainly some intrinsic characteristics of the simulated database. Moreover, additional pre-processing operations on values were investigated also in this case, and some comparisons among the predictions of the trained ML models were examined.

Consistently with the previous models, the ratio between train and test split of the dataset has been kept equal to 80% - 20 %, in accordance with the implemented cross validation procedure, which considered 5 folds.

The input features to supply to the models were selected from some observations on the data. An illustrative example is here reported to explain the concept at the base of the selection of the features. In a complete fuel assembly configuration, the detectors responses are symmetrical with respect to the central point, as can be inferred by the exemplificative complete scenario reported in *Figure 3.4, 3.5 and 3.6* which considers thermal neutrons counts, fast neutrons counts and photons current, respectively.

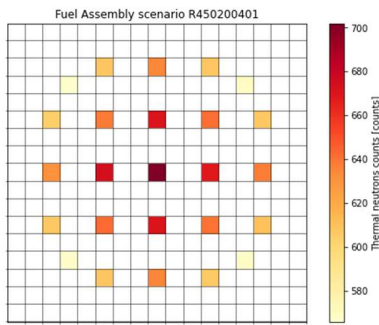


Figure 3.4. Observation on data: Complete scenario fuel assembly configuration - Thermal neutrons counts

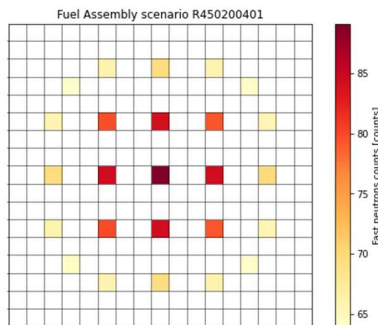


Figure 3.5. Observation on data: Complete scenario fuel assembly configuration - Fast neutrons counts

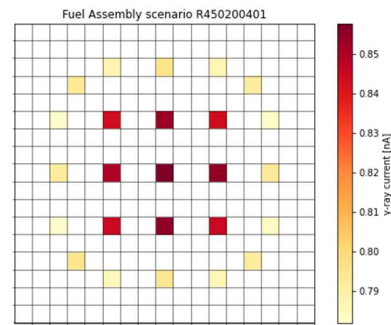


Figure 3.6. Observation on data: Complete scenario fuel assembly configuration - Photons current

Performing the same analysis on a defect scenario, the detectors responses show different sensitivities to the presence of replaced pins. As representative examples are reported the *Figure 3.7, 3.8 and 3.9* which considers thermal neutrons counts, fast neutrons counts and photons current, respectively, in a particular defect fuel assembly configuration. Comparing the responsiveness of the different kinds of detectors to the diverted pin locations (identified by the black squares), the photons current values were selected as input features due to their higher sensitivity to the diversion locations. Moreover, all the different locations of the detectors were used as input features to the model, to consider a comprehensive picture of the data available.

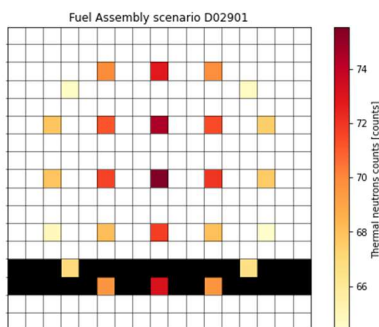


Figure 3.7. Observation on data: Defect scenario fuel assembly configuration - Thermal neutrons counts

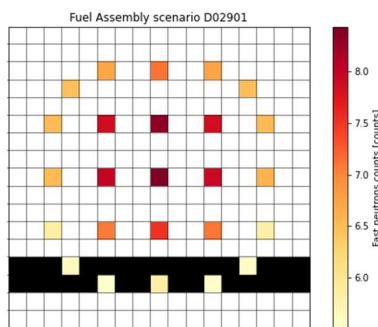


Figure 3.8. Observation on data: Defect scenario fuel assembly configuration - Fast neutrons counts

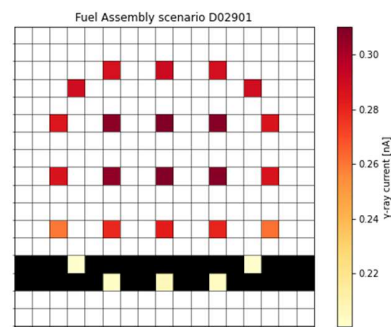


Figure 3.9. Observation on data: Defect scenario fuel assembly configuration - Photons current

In order to exploit the heterogeneity of the values, which is directly linked to their location on the fuel assembly configuration, the selected input features were not manipulated in the preprocessing of the data,

but introduced as they present in the algorithms. In addition, other information were also introduced, besides the ones already present. Indeed, three different preprocessing of the data were investigated:

- SIN model: where the photons detectors responses, for all locations on the geometry of the fuel assembly, were introduced as they are reported in the original database. Thus, the number of input features was equal to N , where $N = 65$ (total number of photons detectors responses).
- ADI model: where the single values and the differences of the single values with respect to the average of the correspondent group of detectors were introduced as inputs. Thus, the number of input features was equal to $2N$, where $N = 65$ (total number of photons detectors responses).
- STD model: where the single values and the standard deviations of the correspondent group of detectors were introduced as inputs. Thus, the number of input features was equal to $N+3$, where $N = 65$ (total number of photons detectors responses).

After the selection of the input features, the procedure was equivalent to the one detailed in the previous section. The train and test data were standardized, and later provided to the algorithms to tune and select the best parameters, fit the models, and predict the outcomes from the inputs of the standardized testing dataset.

If a classification approach had been considered, the outcome would have been a binary classification, which would have returned a value equal to 0 in case it was foreseen the presence of the pins, or equal to 1 in case of the prediction of a replaced pin location. Since it was instead considered a regression approach, the predicted outcomes were a continuous number between 0 and 1, interpretable as a probability of having the replacement of the pin in a certain location on the fuel assembly. In this perspective, the evaluation of the goodness of the models has been instituted consequently, relying also on the implementation of the cross validation of the models, differently with respect to the evaluation metrics introduced in the case of the models for the prediction of the number of dummy pins.

In details, also in this case it was introduced the division of the database in 5 folds, and each fold has been used at a time as testing dataset, to have prediction of outcomes for all the scenarios listed in the original database. In this way, a comparison between the predicted locations of dummy pins and the real locations has been envisaged for all the scenarios in the database. Several parameters of evaluation have been then introduced. First, the outcome values for the presence/replacement have been rounded to the closer integer value. Then, the correct localization of the dummy pin was identified when the real and the rounded predicted outcomes matched, otherwise the localization of the dummy pin has been classified as mis-interpreted. All the uncorrected predictions were then summed up, and as a result the total number of wrong location predictions was given for each scenario. It can be noticed that the rounding condition can be tighten up, for example putting equal to 0.6, 0.7, 0.8 or 0.9 the value of the threshold to consider the predicted location as a replaced pin location. In this way, a larger number of false negatives would have been faced, and vice versa if the threshold was relaxed, possibly leading to a greater number of false alarm scenarios. Next, a division among the correct predicted scenarios and the wrong prediction scenarios was introduced, and a subsequent subdivision that considered the correctness of the complete and defect scenarios cases separately. Eventually, it has been counted the frequency of wrong predictions of complete

and defect scenarios with respect to the total correspondent scenarios configurations, in order to understand which were the most frequent misinterpreted geometrical configurations for the locations of the replacements.

3.2 Neural Networks regression model

Neural Networks (NN) algorithms aim to resemble and mimic the nervous system structure of the human brain, able to memorize and learn from the external stimuli and to elaborate a specific response. The common base structure between NN methods and human nervous system can be deduced from *Figure 3.10*, where the grey dots represent in fact the so-called neurons of the network.

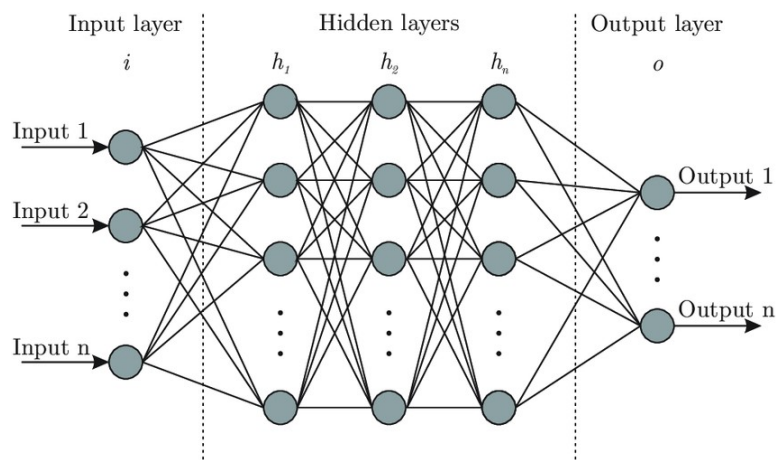


Figure 3.10. Generic neural network structure representation (Bhardwaj, 2021)

The structural objects involved in the networks are the neurons, the layers, and the connections among neurons of different layers. In the first layer (input layer) the neurons are constituted by the input variables in vectorial form, while in the last layer (output layer) the neurons correspond to the vector of the output variables investigated. Among the two, a certain number of hidden layers are present, which contain the hidden variables connected to each neuron of the previous layer by means of activation functions and weights, associated to each neuron. The input variables operate on the output variables by means of the neurons on the hidden layers. Each network is constituted by an input layer, an output layer, and a customized number of hidden layers (at least one), while each hidden layer is constituted by a customized number of neurons.

A neural network is a two-stage regression, or classification, model, which is entirely, or at least partially, non-linear. The derived hidden neurons z_j are linear combinations of the input variables, x_i , and the target variables, y_k , are constructed as a linear combination of the z_j variables. A typical formulation is reported on the *Equation (3.2)* and *Equation (3.3)*, for the modeling of the hidden neurons and the output variables respectively:

$$z_j = f_0 \left(\sum_{i \rightarrow j} w_{ij} x_i + w_{j0} \right) \quad (3.2)$$

$$y_k = f_1 \left(\sum_{j \rightarrow k} w_{jk} z_j + w_{k0} \right) \quad (3.3)$$

Where w_{ij} and w_{jk} are the so-called weights, i.e., parameters to be determined, f_0 and f_1 are predefined activation functions, while w_{j0} and w_{k0} are the so-called biases, constants which are added to the product of variables and weights to offset the results. The structure of a neural network is indeed equivalent to a convolution of functions. In case all the activation functions are linear functions, the outputs are going to be linear combinations of the inputs, vice versa the structure of the network is not linear. The concept of the activation function is to activate just a certain number of outputs of a layer, the ones that, depending on the associated weights, has reached a certain threshold. This results in a non-linearity of the network since the neurons are not always activated contemporarily. Without these non-linear functions, the whole complex structure of the network would collapse to an agglomeration of linear functions which basically eliminates the hidden layers (Hastie, 2017).

As activation function between the n^{th} hidden layer and the output layer, in regression algorithms is generally used the identity function, reported in Equation (3.4):

$$f_1(z) = z \quad (3.4)$$

On the other hand, the selection of the activation functions in the previous layers relies on different possible options. It is important to notice that f_0 is the same for all the neurons belonging to the same layer. Among the several possibilities for the choice of the activation function in neural networks, the ones typically used can be grouped in three main types: the logistic sigmoid function (logistic), the hyperbolic tangent function (tanh), and the Rectified Linear Unit function (ReLU). The choice strongly depends on the problem which is required to be solved (Pedregosa et al., 2011). The trend of the three functions, with their formulation, are reported in Table 3.1 (Jagtap, 2022).

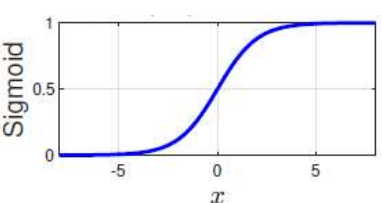
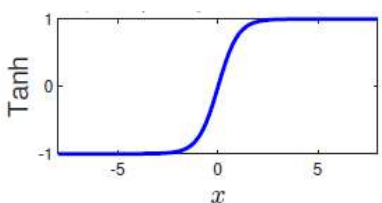
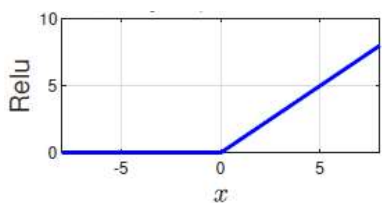
Logistic Sigmoid function	Hyperbolic tangent function	ReLU function
		
$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.5)$	$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.6)$	$\text{ReLU}(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (3.7)$

Table 3.1. Activation functions in NN (Jagtap, 2022)

The sigmoid function is largely employed in classification problems: the codomain of the function is indeed bounded between 0 and 1, and it returns large variation in the results for small variation on x in proximity of the zero axis, implying a good learning in classification problems. On the other hand, it gives some disadvantages at the extremes of the function, where wide changes of x do not affect the changing of y values. This could lead to a problem in the second stage of the NN learning process, when the derivatives of the activation functions in the extreme regions are convergent to zero, implying that these values would not have any impact on the optimization algorithm that minimize the error, and on the improvement of the efficiency of the weights attached to those values in the construction of the networks (Jagtap, 2022).

Moreover, another largely utilized activation function is the hyperbolic tangent function, which has a structure similar to the sigmoid function, but it is defined between -1 and +1. With respect to the previous function, it leads to a steeper derivative and a faster learning process, however it carries the same issue faced at the extremities of the function, where the saturation region, which slows the learning process, is again encountered (Jagtap, 2022).

The ReLU function has been introduced to overcome the vanishing gradient problem discussed above for the previous functions, where relevant changes of the values, if located far from zero, are not detected by the function, thus they are not going to be considered in the learning process. On the contrary, the ReLU function returns null values for each negative value of the domain, and it is linearly increasing in the positive side of the domain, so the neurons belonging to the previous layer that have negative outputs are taken out by the ReLU function. This requires a specific treatment of the input data, which must be scaled between 0 and 1, since the function does not consider negative values. Due to the simplicity of its mathematical formulation, the function has also the advantage of having small computational cost, and it more prone to locally adapt to the data. Nevertheless, the behavior in the negative region could lead to wipe many neurons (Jagtap, 2022).

The activation functions are arbitrary defined in the configuration phase of the neural networks, while the weights associated to each neuron are tuned by an intrinsic iterative process of the algorithm, called backpropagation, which is the learning algorithm at the base of the NN structure. This algorithm compares the predicted outputs of the network with the expected outputs, i.e., the objectives. On the base of this difference an error is computed, used by the algorithm to modify the weights of the neurons in the network, in order to make progressively convergent the output values with the expected ones. This is done iteratively: after every propagation on the network, from the input to the output layer, the associated error is computed, then the algorithm feeds back for adjusting the weights on the computed error, and later forward propagate using the new tuned weights, thus new predictions are associated to the outputs and a new error is defined. The backpropagation is carried on as long as a predefined number of iterations or a predefined tolerance on the error value is reached.

The objective error to minimize is called loss function, which corresponds to the mean squared error for regression problems, computed among the predicted outputs and the actual values of the training set (Pedregosa et al., 2011).

The optimization algorithm which is responsible for updating the weights associated to the neurons is instead called Gradient Descent: it works on the outputs of the activation function feed to a given neuron in the successive layer, noticing which are the neurons contributing more efficiently for the predictions (whose weights are then going to be increased), and which are the ones less effective in the prediction phase (thus the importance of the respective weights are going to be decreased). This task is performed on all the hidden layers and associated neurons. The gradient descent algorithm is linked to the loss function since it has to find the local minimum of the loss function computing its gradient and use this result to tune the weights and decrease the loss function at the next epoch (an epoch is defined as once through the entire training data). The algorithm proceeds in the minimization of the objective function until the maximum number of predefined epochs is reached, or when it is reached a certain tolerance on the improvement of the loss.

Several algorithms exist as solvers for the weights optimization, however all of them have to deal with the presence of more than one local minimum of the loss function. This could rise an issue in the process of the gradient descent algorithm since at a certain point it could reach a local minimum, which does not correspond to the function lowest minimum, and get stuck there, leading to reach a convergence of the overall results which are not at the lowest possible. To partially get rid of this problem, i.e., getting out from a local minimum to reach an overall lower one, Stochastic Gradient Descent (SGD) algorithms have been introduced (Bottou, 1991). In total gradient descent algorithms, the loss function is averaged over all training samples, so the computation of its gradient requires to consider all the samples, and the weights are updated after every loop over the whole training set, in order to converge to a local minimum of the loss function. The innovation introduced by the SGD algorithms is to update the weights considering a single training sample at a time, according to the gradient of the loss function, i.e., the value of the loss for this sample only. In this way the parameters are estimated for every observation, and not to the whole sample set: it increases the randomness in the searching phase of the local minimum, since it considers more widely the trend of the function, and it would more likely find a global minimum. Due to the characteristics of the SGD algorithms (faster, more reliable, and less prone to converge at non-performant local minima), they have been largely used for big dataset in neural networks models (Bottou, 1991). The mathematical formulation of the regularized training error to minimize in the SGD procedure is reported in *Equation (3.8)* (Pedregosa et al., 2011):

$$Error(\hat{y}, y, w) = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{y}_i) + \alpha R(w) \quad (3.8)$$

Where:

- $L(y_i, \hat{y}_i)$ is the loss function for the evaluation of the discrepancy between actual values y_i and predicted values \hat{y}_i . For regression models it relies on the formula of the Mean Squared Error (MSE), which comes down to the Squared Error since the SGD considers one observation at a time, as reported in *Eq. (3.9)*:

$$L(y_i, \hat{y}_i) = SE(y_i, \hat{y}_i) = \frac{1}{2}(y_i - \hat{y}_i)^2 \quad (3.9)$$

- α is the hyperparameter that control the regularization strength;
- $R(w)$ is the regularization term, or penalty, which penalizes the model complexity.

The introduction of a penalization function is foreseen in order to avoid the problems related to overfitting of the data, discussed in the next section.

Intuitively, adopting an algorithm which follows singularly every sample at each step, instead of considering the loss function on the whole training set, would increase the computational effort. The compromise between the advantages of the stochastic method and a lower computational effort is reached introducing the batch concept. In details, the training dataset is randomly subdivided in batches of a certain size, arbitrarily defined, containing the samples which are processed before updating the internal model parameters, thus the samples are processed in groups and the parameters for the loss function are not calculated one by one. This implies that just part of the dataset is analyzed at each iteration, which is anyway changed at each iteration, with the aim of increasing the variability of the data and the possibility to reach a more suitable local minimum. A training dataset can be divided into one or more batches: the batch size could cover the size of the training set (and in this case an iteration corresponds to an epoch), vice versa the batch size can be set equal to 1, or a size can be set in between these two extremes. In the latter case, the algorithm is called mini-batch gradient descent and an epoch is completed once the iterations of the gradient descent algorithm have considered all the subgroups of the training set, covering the whole dataset (Smith, 2018).

Eventually, training neural networks could lead to the problem of overfitting and the underfitting of the data, intrinsically introduced by the high number of weights and the complex structure involved in the algorithm. Both issues rely on imprecise concepts learned by the model while fitting the data in the training phase, which will later reflect on the accuracy of the predictions for new data which had not been seen by the model during the learning phase.

Overfitting is encountered when the constructed model is too refined on the training data, i.e., on both the features and the statistical noises introduced by the data, thus it is not able to adapt to the new data that are provided to the trained model, leading to inaccurate predictions. Contrarily, when the model has not learnt enough about the data features in the training phase, it is later inefficient to predict outcomes on both the training and the testing data.

In order to be aware and prevent these effects on the model during the training phase, it is normally introduced a validation set, obtained by leaving aside part of the training dataset during the forward propagation phase, later used as unseen data against which is going to be evaluated the goodness of the model, for tuning the internal parameters in the backward phase. To increase the validation accuracy, the validation set can be randomly changed at every epoch.

An illustrative example of the trends of the errors in case of overfitting and underfitting, is reported in *Figure 3.11*. As above mentioned, normally the goodness in the training phase is assessed against the validation subset, whose error is represented by the orange line in the figure. While the model complexity is increasing, which can be inferred as the progressive adjustments of the weights along the epochs, the error on the training data decreases, meaning that the network is going to adapt more and more on the training data. On the other hand, the model is enhancing its performances on the predictions of the new data, as long as it is learning more features on the dataset, thus the validation error decreases, till it is going to increase again due to overfitting of the data. The optimal trade-off is encountered when the validation error reaches a plateau on the curve: at this point the progressive refinement of the network should be stopped.

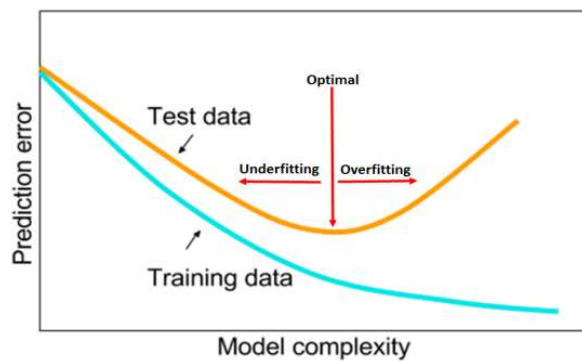


Figure 3.11. Prediction error for overfitted and underfitted models (Smith, 2018)

Several hyperparameters and techniques can be introduced to deal with overfitting and underfitting: among others, learning rate, penalization functions and early stopping are following presented. In detail:

- Learning rate: it controls how much the weights are updated each time that the error on the loss function is estimated. A small value of this parameter results in higher accuracy in the loss function, but a larger computational time and the possibility of overfitting. On the other side, a large value of learning rate could accelerate the algorithm, since it requires fewer epochs if large changes are provided to the weights, but it could conduct to the divergence of the training process (Smith, 2018).
- Penalization functions: they penalize too complex solutions and make the networks less prone to overfit the data, preventing to associate excessively large values to the weights. This regularization term was introduced in Eq. (3.8), and the function $R(w)$ is normally selected among the L2 norm, the L1 norm and the elastic net (Pedregosa et al., 2011).
- Early stopping: it stops the training of the model when the error on the validation set start to degrade, to reduce overfitting. It requires a monitor criterion, to evaluate the point where to stop the iteration process, namely the dataset over which are evaluated the performances and an evaluation metric. It is also necessary to define a priori the trigger for stopping the process: since the evaluation metric is a statistical inference, it may happen that the first insight of the increasing of the error is a false alarm of the overfitting, and the curve could continue to decline after it, reaching several other local minima. Thus, the trigger must be suitable chosen to consider this occurrence, and several possibilities are consequently defined (Pedregosa et al., 2011).

Configuration of the models

Python was the environment adopted for the development of the models also in the case of neural networks, where the tools selected were provided by the Keras API, which runs over TensorFlow. The general procedure for the implementation of the NN regression models is similar to the one already presented for the k-NN, thus here the steps of the procedure are reported, highlighting the differences with respect to the former one:

1. Import of the dataset described in [Section 2.2](#), and division of the samples in training and testing data.
2. Select the features of interest from the input database.
3. Pre-process the data.
4. Construct the architecture of the networks. To build and compile each layer individually, the model selected was Sequential, while the number of layers and the number of neurons for each layer have been defined with an hyperparameter optimization model. The chosen approach was the Hyperband optimization algorithm, and the parameters selected for the optimization process were the number of layers, between 2 and 10, and the number of neurons within each layer, ranging between 50 and 1000. The loss function to minimize for the optimization of the parameters was the Mean Squared Error, in accordance with the common practice in regression algorithms. Contextually, the activation functions must be defined. Eventually, the last information that has to be provided for the models construction is the choice of the kernel initializer.
5. Compile the models: the optimization algorithm, the loss function to minimize and the evaluation metric to monitor along the training process require to be defined.
6. Fit the NN regressor on the training dataset, using the parameters retrieved from point 4, tuned on the input features introduced in point 2. Other hyperparameters for the training configuration have to be selected, namely the number of epochs, the batch size, and the ones involved in the early stopping method.
7. Predict the outputs associated to the input variables of the test dataset, by means of the fitted NN models.
8. Evaluate the goodness of the models, using different error metrics performed on the discrepancies among predicted and actual values.
9. Validate the models: run the algorithms several times with the aim of increasing the randomness of the observations stored in train and test datasets.

Analogously to the configuration of the k-NN models, the procedure has been followed for predicting both the number of dummy pins associated to a certain scenario and the locations of the dummy pins in a certain geometry configuration. The procedure steps were changed accordingly to the outcomes under examination.

3.2.1 Number of dummy pins

In this section of the thesis, the neural networks algorithm has been applied for the prediction of the number of dummy pins in the scenarios introduced by the described dataset. The implementation process is equivalent to the one performed for the k-nearest neighbors models, since a comparison between the

two methods, at parity of other conditions, was one of the objectives of this work. Consequently, are following presented the variations introduced in this method, with respect to the one referring to the k-NN, due to the intrinsic differences among the structures of the two methods.

Concerning the percentage of splitting between training and testing dataset, it has been kept equal to the proportion of 80-20%, in accordance with the 5-fold Cross Validation scheme.

The different combinations of kinds of detectors, groups of detectors and pre-processing operations on the values considered as input features in the training phase of the learning process were the same introduced for the k-NN case. The final objective was again to compare the performances of the models trained on different input features.

Once the input features were selected, the consecutive train and test data have been additionally pre-processed to fulfil the requirement of having standardized, normalized, or scaled data in input of the ML models. Specifically, this stage was intrinsically dependent on the choice of the functions involved in the next steps, thus both the training and the testing dataset were scaled using the MinMaxScaler function implemented in scikit-learn library, setting the range between 0 and 1, hence all the sample data were scaled within the defined range.

In the construction phase of the models, it was introduced the HyperBand method for the optimization of the structural objects of the network. This method was chosen since it joins the advantage of using a random searcher model to the concept that losses in error functions are bigger in the initial epochs of the training process. In the first stage are trained many possible configurations of the network, for a small number of epochs, integrating the early stopping criterion. Then, the most promising architectures encountered in this preliminary stage are subsequently re-tuned for a maximum number of predefined epochs. At the last stage, the best configuration is chosen for development of the model (Li et al., 2018). The HyperBand method allows to optimize contextually other parameters, but this path was not followed with the intention of sparing computational time, due to the large effort required for the training of all the models foreseen in this phase.

Contextually, the activation functions have been set. For what concern the function between the last hidden layer and the output layer, it was set to linear, as requested by regression models theory, while for the preceding layers the ReLU function was selected. This had implied the scaling of both the training and testing dataset between 0 and 1, in the pre-processing phase, as required by the function itself, as above described. Eventually, the kernel initializer was implemented since it defines the way to set the initial random weights of the layers. Among the several available initializers in Keras, the one selected was the HeNormal class, which assured better performances in combination with ReLU functions (Chollet et al., 2015).

Regarding the compiling phase, the Adam algorithm has been chosen as optimizer, which is a variation of the stochastic gradient descent: it involves limited computational effort and memory requirement and introduce different adaptative learning rates for different parameters in the network, typically necessitating little tuning (Kingma, 2015). The learning rate initially introduced was set by several trials

and errors. As already introduced, the loss function selected was the mean squared error, and the evaluation metrics for the prediction accuracy was set to root mean squared error, to be consistent with the evaluation metrics introduced for the evaluation of the goodness of the model.

An exemplificative neural network architecture is reported in *Figure 3.12*, namely the model that considered as input features the values of thermal neutrons counts and photon current for all the detectors positions. It starts from the input layer, where 130 vectorial features are fed to the network, it continues with 9 hidden layers, each having a number of neutrons reported in the second term of the label “kernel”, and eventually ends with the output layer having one vectorial output. The label “bias” refers to the number of biases associated to the correspondent layer. Each layer is defined as “Dense”, meaning that every neuron in a layer is connected to every neuron in both the previous and successive layers, while the activation function of each layer is reported in the red rectangle, except the last layer which employs a linear activation function.

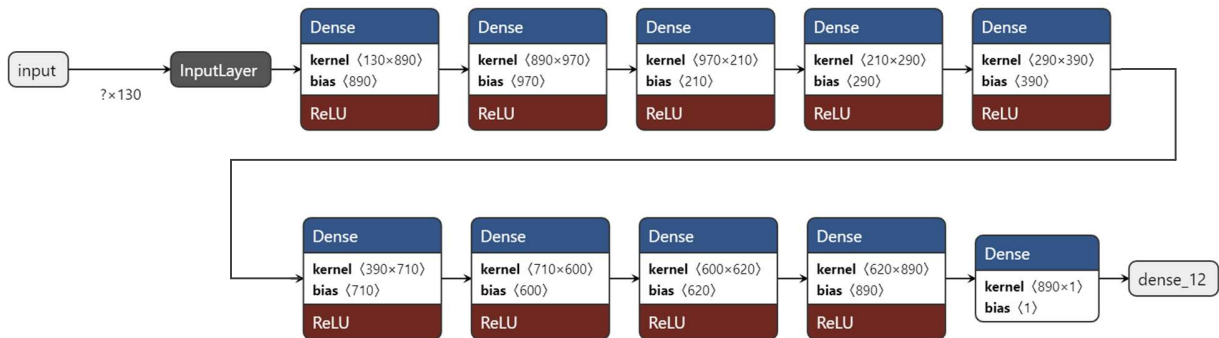


Figure 3.12. Example of a NN architecture for the prediction of number of dummy pins

The last hyperparameters to be tuned for the training configuration were the number of epochs, set to 500 (which was a tradeoff between the large computational time required and the necessity to avoid overfitting, prevented in combination with the early stopping criterion), the batch size, set equal to 32, and the ones related to the early stopping. The latter requires a monitoring criterion, chosen as the value of the RMSE calculated on the validation set and, moreover, the learning process stopped after 50 epochs in which a minimum change of the RMSE equal to 0.01 has not been found. An illustrative example of the trends of the validation and train losses along the progression of the epochs, reported for the same model represented in *Figure 3.12*, is reported in *Figure 3.13*. It can be inferred that the desirable plateau, on the validation losses, to avoid overfitting and underfitting is reached.

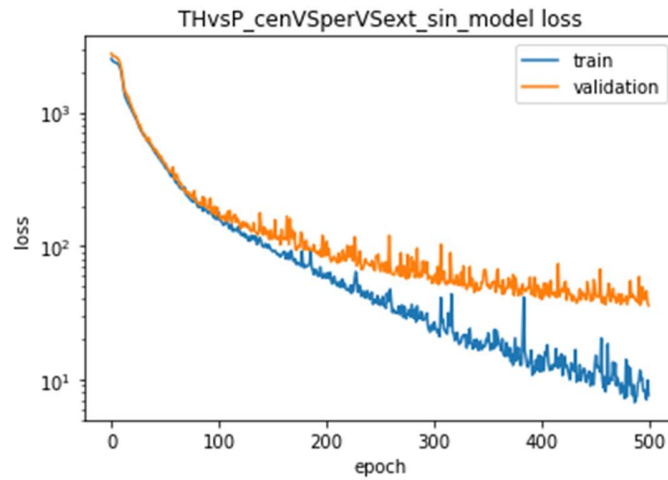


Figure 3.13. Train and validation losses trends for an illustrative NN model

Afterwards, the algorithms have used the scaled training dataset to fit the models and predict the outcomes from the inputs of the scaled testing dataset. The outcomes of the NN models have been used to evaluate the performances of the models exactly as stated for the k-NN models for the prediction of the number of dummy pins, thus the RMSE and the Delta were the calculated evaluation metrics for each model.

Eventually, the Cross Validation scheme was constructed similarly to the one previously introduced in the [Section 3.1.1](#). A discrepancy with respect to the previous process was established, due to the large computational time caused by the repetition of the HyperBand scheme for all the 147 models investigated. Specifically, the optimization algorithm was implemented for the searching of the best numbers of layers and neurons just in the first of the 25 loops foreseen by the Cross Validation, and later the same architecture was introduced for the remaining loops of the validation process for the model that considered the same input features.

3.2.2. Location of dummy pins

Eventually, this last section is dedicated to the implementation of the neural network method for the prediction of the locations of the replaced pins in the fuel assembly configurations. The procedure has integrated both the structure of the neural network introduced in the last paragraph, and the scheme seen in the [Section 3.1.2](#) for the same expected outcomes.

Consistently with the previous models, the ratio between train and test split of the dataset has been kept equal to 80% - 20 %, in accordance with the implemented cross validation procedure, which considered 5 folds.

The input features were selected correspondingly to the k-NN models constructed for the prediction of the locations of the dummy pins, hence considering the photons current values, more sensitive to the diversions on the grid lattice, for all the locations of the detectors. Moreover, the same preprocessing

operations on values before introduced were adopted: SIN (having N input features), ADI (having 2N input features) and STD (having N+3 input features).

After the selection of the input features, the train and test data were scaled to fulfill the requirement of the ReLU activation function, and later provided to the HyperBand algorithm to tune and select the best parameters, seeking among the same ranges of layers and neurons reported in the previous paragraph. Analogously to the previous NN models, the activation functions were set as ReLU among the hidden layers, and as linear on the output layer. As well, the kernel initializer was selected as HeNormal.

Concerning the compiling step, the Adam optimizer was implemented, and contextually also the mean square error as loss function and the root mean square error as evaluation metric.

The architectures of the three analyzed models are following reported in *Figure 3.14*, *Figure 3.15* and *Figure 3.16* for SIN, ADI and STD case respectively. With respect to the former model, in *Figure 3.12*, here 264 outputs are expected from the output layer, namely the 264 locations of the pins on the squared grid lattice. On the other hand, different inputs are feed in the corresponding layer, indeed the N (65), 2N (130) and N+3 (68) input features of the diverse models.

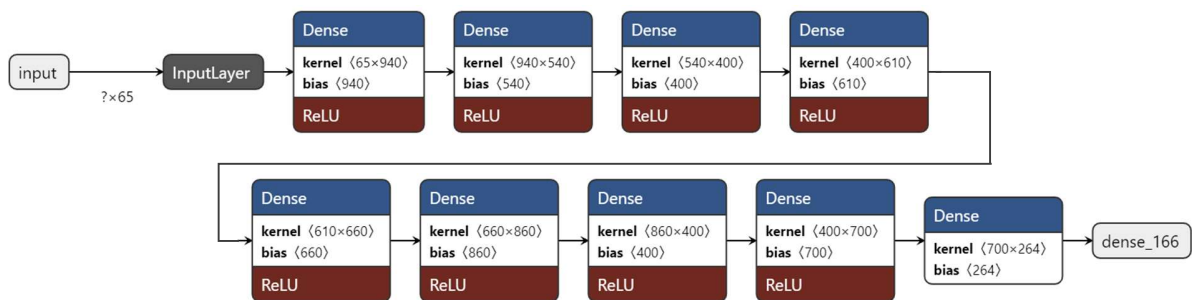


Figure 3.14. SIN model architecture for the prediction of location of dummy pins

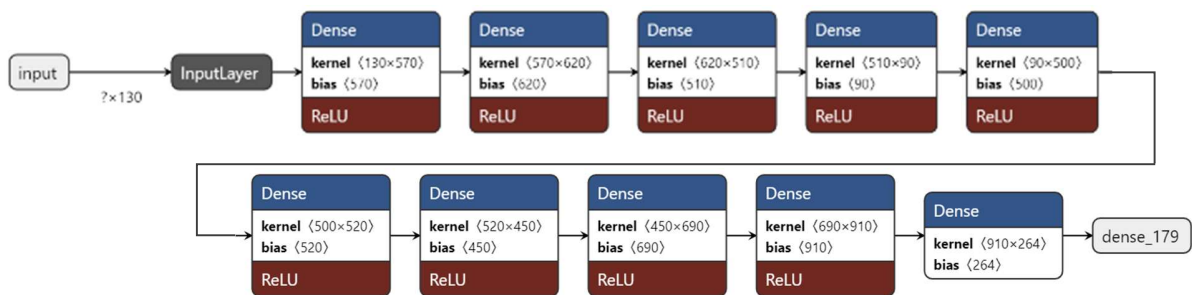


Figure 3.15. ADI model architecture for the prediction of location of dummy pins

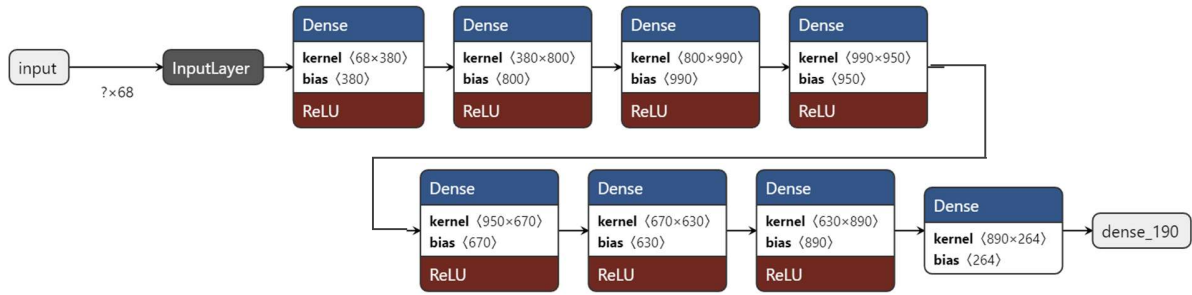


Figure 3.16. STD model architecture for the prediction of location of dummy pins

Regarding the tuning of the hyperparameters pertaining to the training configuration, the maximum number of epochs was set equal to 1000, since the low number of developed models (three) allowed to handle a larger computational effort. Besides, the early stopping criteria was either way employed, resulting in training processes having the trends reported in *Figure 3.17*, *Figure 3.18* and *Figure 3.19* for the SIN, ADI and STD models respectively. The monitoring criterion chosen was again the value of the RMSE calculated on the validation set while the learning process stopped after 20 epochs in which a minimum change of the RMSE equal to 0.001 has not been found. The latter was lowered since the expected outcomes in this case were numbers between 0 and 1, nearly one order of magnitude lower than the situation encountered in the models predicting the number of dummy pins present. From the following plots, it can be better inferred the result of the introduction of the early stopping criterion: all the training processes stopped at around 250 epochs, when not a large improvement of the loss function was addressed, and the occurrence of overfitting was almost encountered.

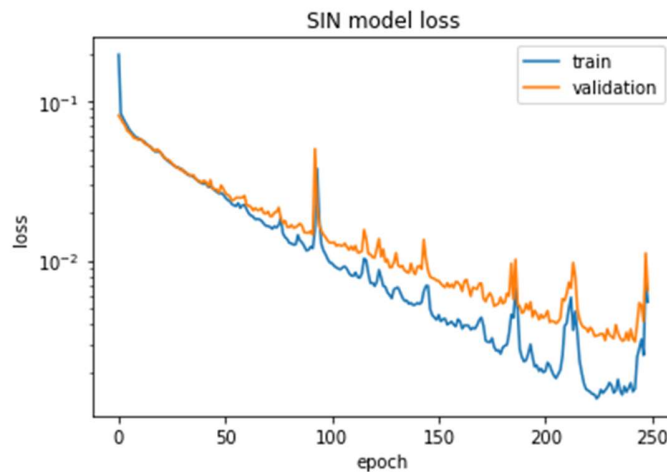


Figure 3.17. Train and validation losses trends for SIN model

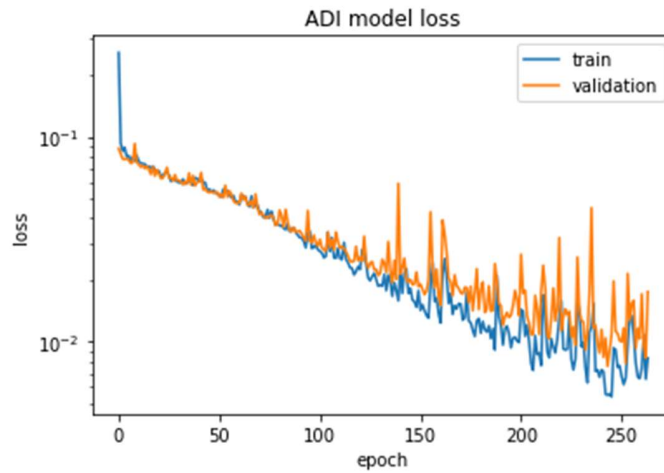


Figure 3.18. Train and validation losses trends for ADI model

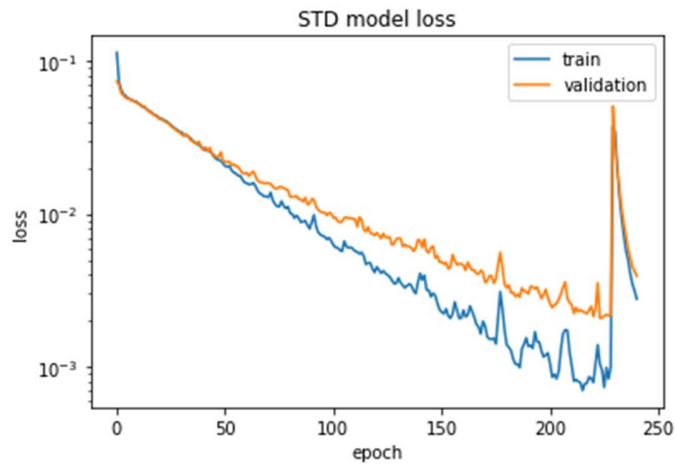


Figure 3.19. Train and validation losses trends for STD model

Subsequently, the scaled training datasets has been feed to the algorithms to fit the models and predict the outcomes for the scaled testing datasets inputs. Even in this case, the models have returned values between 0 and 1, interpreted as the probability to have a pin replacement. Indeed, the evaluation metrics were the same presented in the correspondent chapter of the k-NN configuration models. The evaluation process was strictly related to the Cross Validation procedure, which even in this case has considered the 5 folds subdivision of the dataset and the use of each fold as testing data once, to perform predictions for all simulated scenarios.

4. RESULTS

In this chapter the results of the analyses performed by means of the developed k-nearest neighbors and neural networks machine learning models are reported. Only the most insightful outcomes are described in the next paragraphs, while the others are reported in [Appendix B](#).

4.1 k-Nearest Neighbors regression models

Concerning the k-NN regression algorithms, the results of the analyses performed for the prediction of the number of dummy pins in the simulated scenarios and the location of the dummy pins in the fuel assembly geometrical configurations are presented.

4.1.1 Number of dummy pins

First some examples of the models outcomes are reported just for the sake of illustration of the working principle of the algorithms. Few comparisons between the real and the predicted number of dummy pins are following presented, considering certain input features, i.e., kind, group and preprocessing operation of the detectors responses values. To follow a logical continuity with the graphs introduced in the database visualization paragraph, the outcomes of the models that considered the average and the standard deviation within the peripheral detectors are following reported. In *Figure 4.1*, *Figure 4.2* and *Figure 4.3* are represented the comparisons of exact and predicted number of dummy pins when as detector responses inputs were considered thermal neutrons counts, fast neutrons counts and photons current respectively.

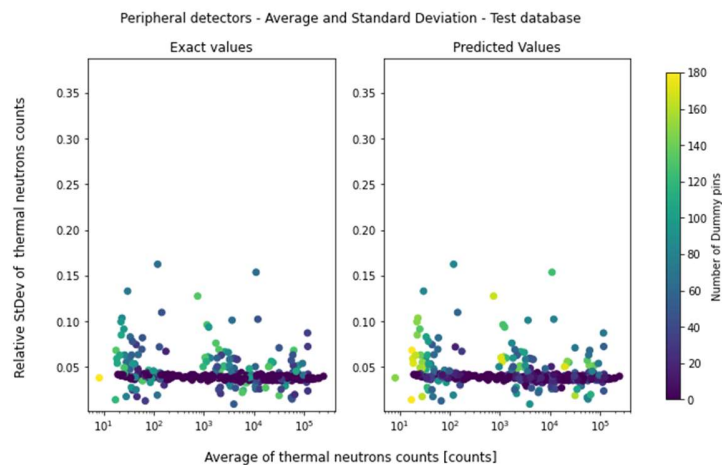


Figure 4.1. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Thermal neutrons counts

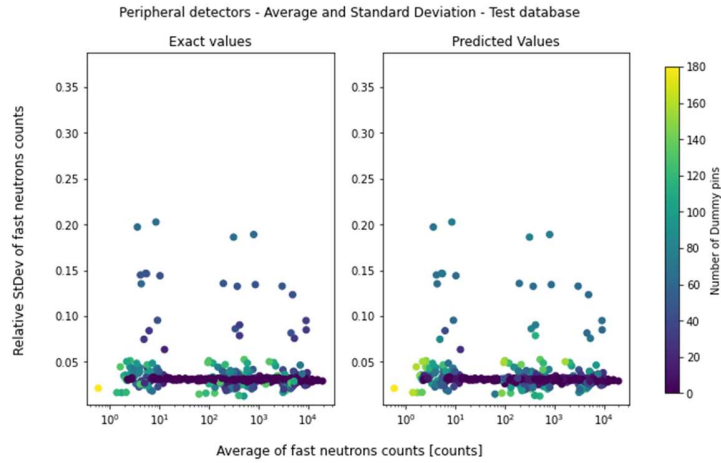


Figure 4.2. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Fast neutrons counts

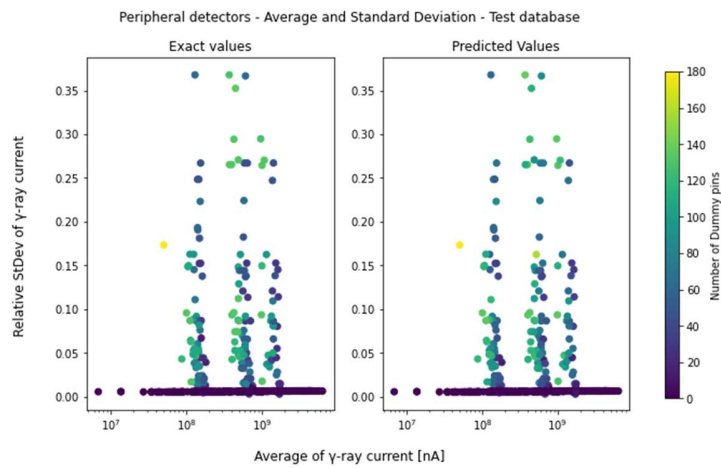


Figure 4.3. Comparison between exact and predicted values of number of dummy pins - Peripheral detectors - Average and standard deviation - Photons current

The previous plots have been produced only for illustrative aims and were not foreseen with the intention of presenting meaningful statistical inferences. Indeed, those outcomes are the ones predicted by the first of the 25 runs introduced in the cross validation procedure and refer only to the comparison between the exact and predicted outputs of the testing database, i.e., one fifth of the original database.

Hence, comparisons among evaluation parameters are here presented, as results of the cross validation process. First of all, the graphs representing the comparisons among RMSE of all the different combinations of kinds and groups of detectors are reported in *Figure 4.4, 4.5 and 4.6*, for the case of Sum, Ave&Std and Sin operations within the values, respectively. In the figures, the mean value of the RMSE, depicted with the empty mark, is represented with the respective standard deviation, depicted as the vertical error bar. Among the RMSE values retrieved by the different models, considering the same preprocessing operations on the values, generally better performances were encountered when all groups of detectors were investigated in the same model, and photons current (P) detectors responses were involved.

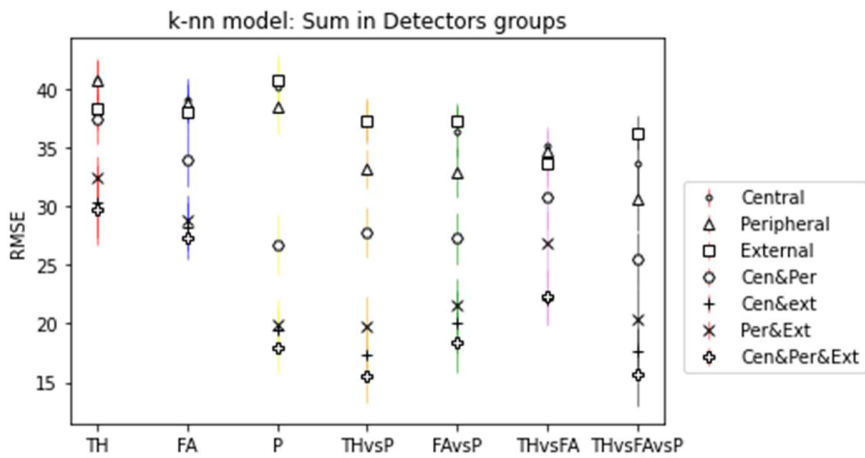


Figure 4.4. k-NN model RMSE - Sum in detectors groups for all combinations of kinds of detectors responses

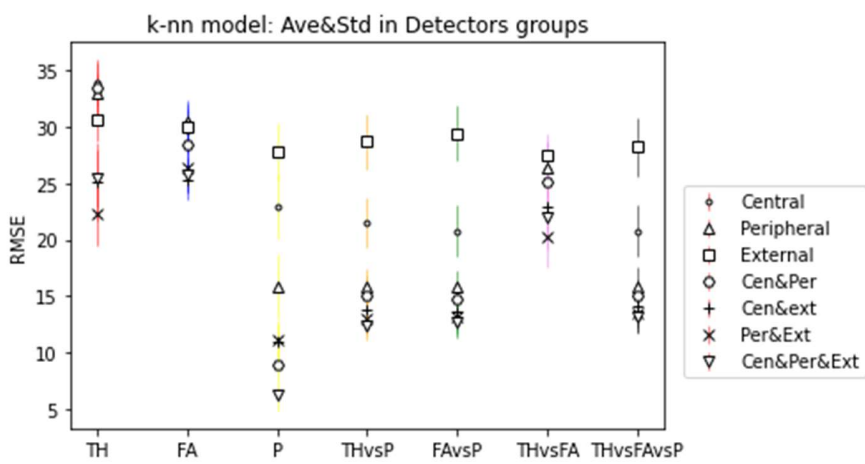


Figure 4.5. k-NN model RMSE - Average and standard deviation in detectors groups for all combinations of kinds of detectors responses

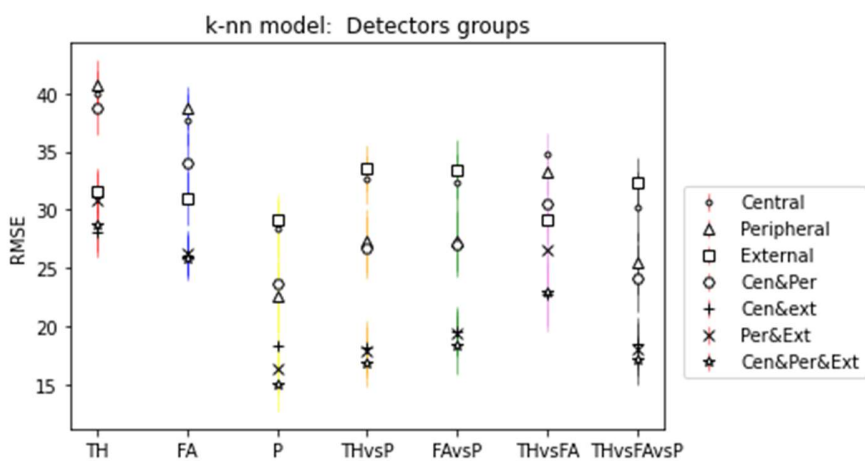


Figure 4.6. k-NN model RMSE - Single inputs of detectors groups for all combinations of kinds of detectors responses

A further investigation on the trend of the RMSE values for models considering all groups of detectors is presented in *Figure 4.7*, comparing different operations and different combinations of kind of detectors responses. Again, the higher performances of the models which supplied photons current as input features can be inferred, related to the greater sensitivity of the gamma-ray detectors responses with respect to the diversions of the fuel assembly configuration. Moreover, the models based on the average and standard deviation as preprocessing operations within the values of the detector groups produced better results with respect to the other operations investigated. This trend can be referred to the investigation performed on the values of average and relative standard deviation of the photons detectors responses in the database visualization section, where the dispersion of the relative standard deviation values in the defect scenarios cases was illustrated as a consequence of the higher sensitivity for the defect scenarios, with respect to thermal neutrons or fast neutrons counts.

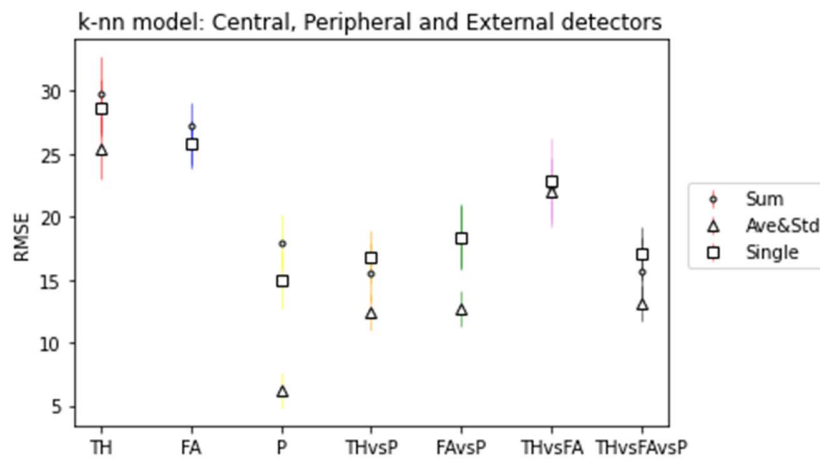


Figure 4.7. k-NN model RMSE - Operations on values within all detectors groups for all combinations of kinds of detectors responses

Moreover, the Delta, i.e., the difference between the predicted and exact values of number of dummy pins, of the models accounting for all the groups of detectors in the case of the most accurate operation on input features (Ave&Std) are presented in *Figure 4.8*. The comparison has been conducted on the percentage of the predictions belonging to each bin presented in the histogram. In details, the difference between actual and predicted values of number of dummy pins was computed for all the scenarios of the original dataset. The results were grouped in classes, which covers a range of 10 pins of discrepancies between true and predicted outputs, except for the middle interval, which covers a discrepancy between -1 and 1 Delta. Each resulting Δ fell in the bin of competence, which at end returned the percentage of predictions belonging to each range. The bin covering the largest number of predictions is the one ranging in a Delta from -1 to 1, implying that the majority of the misinterpreted predictions were actually small discrepancies. More examples, for the different combinations of detectors groups, are reported in [Appendix B](#).

k-nn model: Central, Peripheral and External detectors - Ave&Std

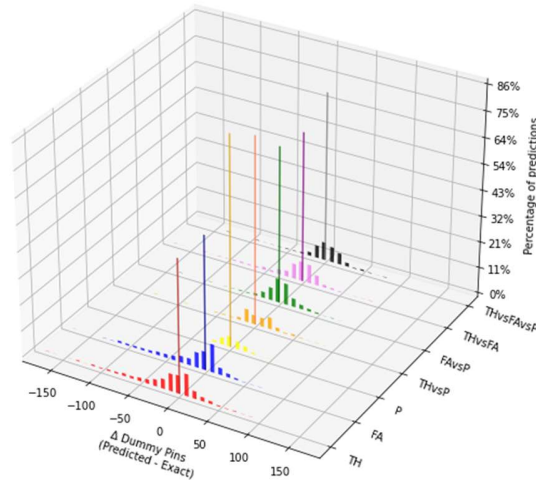


Figure 4.8. k-NN model Δ - Ave&Std within all detectors groups for all combinations of kind of detectors responses

In Table 4.1, Table 4.2, and Table 4.3 are reported the mean values and the standard deviation of RMSE for the different models used, considering Sum, Ave&Std and Sin models, respectively. The models which best predict within each group or combination of groups are highlighted, and the best one for each preprocessing operation is then selected below. Overall, the best prediction model is the one referring to the average and standard deviation of photons detectors responses values considering all the detectors locations, which returned a mean RMSE equal to 6.2 and a standard deviation of 1.4.

Eventually, a conclusion can be assessed comparing the results of the models for the different preprocessing operations, fixing the other conditions. Although more information were provided to the models which deal with the values of the detectors responses as they present in the database, the best performances were encountered for the Ave&Std models. It may be deduced that increasing the number of features, it would increase the statistical noise associated to them, since feeding the values for all locations could be meaningful when predicting the localized diversions on the lattice grid, but not when averaged on the whole lattice. Therefore, providing all the detailed information could be misleading for a measure of diversion which is averaged on the lattice, unless the single inputs would have been feed to the models in order to predict the precise location of the diversions, to be later summed up to retrieve the total number of dummy pins, as it was indeed implemented in the predictions of the replacements, later presented in this work.

	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT	
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std
SUM	TH	40.8	1.7	40.8	1.7	38.3	1.8	37.4	2.1	30.4	3.1	32.4	1.9	29.7	3.0
	FA	39.1	1.8	38.8	1.7	38.0	1.7	34.0	2.3	28.3	2.0	28.9	2.0	27.3	1.7
	P	40.2	2.1	38.5	2.2	40.8	2.1	26.8	2.5	19.4	2.0	19.9	2.1	18.0	2.1
	THvsP	37.3	1.9	33.2	1.7	37.3	1.7	27.8	2.1	17.4	2.2	19.8	2.5	15.5	2.3
	FAvsP	36.4	2.1	33.0	2.1	37.4	1.4	27.3	2.2	20.1	2.6	21.7	2.2	18.4	2.6
	THvsFA	35.2	1.7	34.7	1.9	33.6	1.7	30.7	2.7	22.1	2.3	26.9	2.7	22.3	2.3
	THvsFAvsP	33.7	2.5	30.7	2.8	36.3	1.5	25.5	2.4	17.6	2.0	20.4	2.7	15.7	2.7
	min in kinds:	33.7	2.5	30.7	2.8	33.6	1.7	25.5	2.4	17.4	2.2	19.8	2.5	15.5	2.3
min in operations:	15.5						2.3								

Table 4.1. k-NN models: Mean RMSE and standard deviation – Operation on values: Sum

	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT	
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std
AVERAGE & STANDARD	TH	34.0	2.0	33.0	1.8	30.6	1.9	33.6	2.2	25.1	3.5	22.4	2.9	25.5	2.5
	FA	29.6	2.0	30.5	1.9	30.0	2.1	28.4	2.1	25.2	1.7	26.4	2.2	25.8	1.6
	P	22.9	2.9	15.9	2.8	27.8	2.5	8.9	1.3	10.9	1.8	11.0	1.4	6.2	1.4
	THvsP	21.5	2.3	15.9	1.6	28.7	2.4	15.1	1.7	13.8	1.8	13.0	1.4	12.4	1.4
	FAvsP	20.8	2.4	15.8	1.6	29.4	2.4	14.7	1.6	13.6	1.7	13.1	1.5	12.7	1.4
	THvsFA	26.2	2.4	26.4	2.3	27.4	2.0	25.1	3.1	22.9	3.0	20.3	2.8	22.0	2.8
	THvsFAvsP	20.8	2.3	15.8	1.8	28.2	2.6	15.0	1.4	14.1	1.9	13.4	1.8	13.1	1.4
	min in kinds:	20.8	2.3	15.8	1.6	27.4	2.0	8.9	1.3	10.9	1.8	11.0	1.4	6.2	1.4
min in operations:	6.2						1.4								

Table 4.2. k-NN models: Mean RMSE and standard deviation – Operation on values: Average and Standard Deviation

	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT	
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std
SINGLE	TH	40.0	1.9	40.7	2.1	31.6	2.0	38.8	2.3	28.1	2.2	30.8	2.4	28.6	2.3
	FA	37.7	2.2	38.8	1.8	30.9	2.3	34.0	2.7	26.1	1.8	26.2	2.0	25.8	1.9
	P	28.4	2.4	22.6	3.0	29.1	2.2	23.6	3.2	18.4	2.1	16.3	2.1	15.0	2.3
	THvsP	32.6	2.1	27.3	2.8	33.6	1.9	26.7	2.6	18.2	2.2	17.8	2.2	16.9	2.1
	FAvsP	32.3	2.5	27.3	2.6	33.5	2.5	27.0	2.8	19.6	2.1	19.3	2.0	18.4	2.5
	THvsFA	34.9	1.7	33.2	2.0	29.1	2.0	30.5	2.3	22.7	2.8	26.5	3.5	22.9	3.4
	THvsFAvsP	30.2	3.0	25.4	2.7	32.4	2.1	24.1	2.9	18.5	2.2	18.0	2.3	17.1	2.1
	min in kinds:	28.4		22.6	3.0	29.1	2.0	23.6	3.2	18.2	2.2	16.3	2.1	15.0	2.3
min in operations:	15.0						2.3								

Table 4.3. k-NN models: Mean RMSE and standard deviation – Operation on values: Single input

4.1.2 Location of dummy pins

As in the previous section, dedicated to the presentations of the models results, a qualitative example of the working procedure of the prediction models is shown. A real configuration of a defect scenario is taken as an example, and it is represented in *Figure 4.9*, where the respective photons current values are superimposed on the diverted fuel assembly geometry.

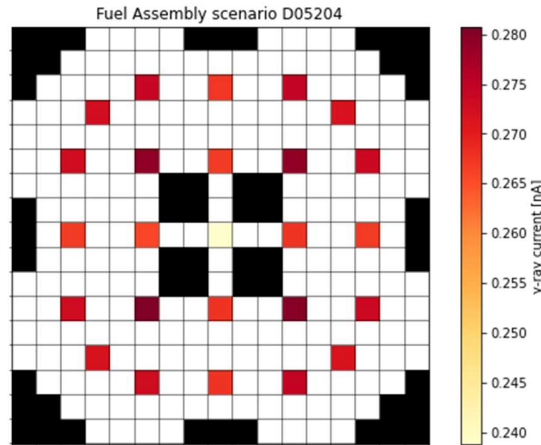


Figure 4.9. Defect scenario fuel assembly configuration - Real locations of dummy pins - Photons current

For the same defect scenario, the outcomes of the prediction models are instead reported in *Figure 4.10*, *4.11* and *4.12*, for the SIN, ADI and STD models respectively. The locations of the detectors are reported in bordeaux, whereas the probability of having a replaced pin in a certain location relies on a dashed colorbar going from white (representing “0”, i.e., pin present) to black (representing “1”, i.e., pin replaced). Thus, the darker the square on the lattice, the higher the predicted probability of having a replaced pin.

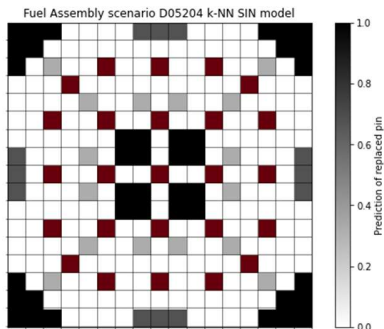


Figure 4.10. k-NN SIN model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

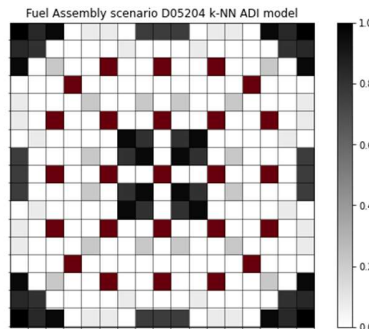


Figure 4.11. k-NN ADI model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

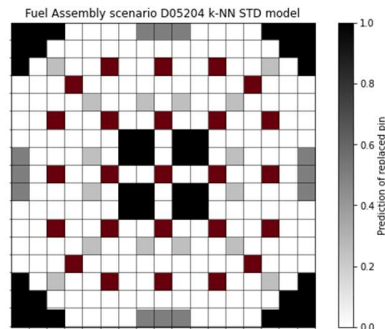


Figure 4.12. k-NN STD model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

An analysis on the overall predictions for the location of replaced pins is here presented. First, the percentage of correct and wrong predictions of the locations of the replaced pins for each model are reported. As previously mentioned, the probability of prediction to encounter a replaced pin in a certain location has been rounded to the nearest integer value, i.e., considering a threshold of 0.5. The rounded outcome has been then compared to the real outcome, to verify the correctness of the predictions for

each position on the fuel assembly. For each scenario, the number of wrong positions predicted were therefore summed up, reported on the abscissa of the following graphs. On the ordinate is instead reported the percentage of the scenarios for which has been predicted the defined number of wrong positions, with respect to all scenarios. In *Figure 4.13* the results for the SIN model are presented, which focus only on the wrong predicted scenarios, from which it can be inferred the percentage, out of all scenarios, of the number of wrong positions predicted in each scenarios investigated. A brief summary of the number of predicted scenarios belonging to each category is listed, in the case of SIN model:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 225/963 (23.4%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 738/963 (76.6%);

having a total percentage of accuracy of 69.4%.

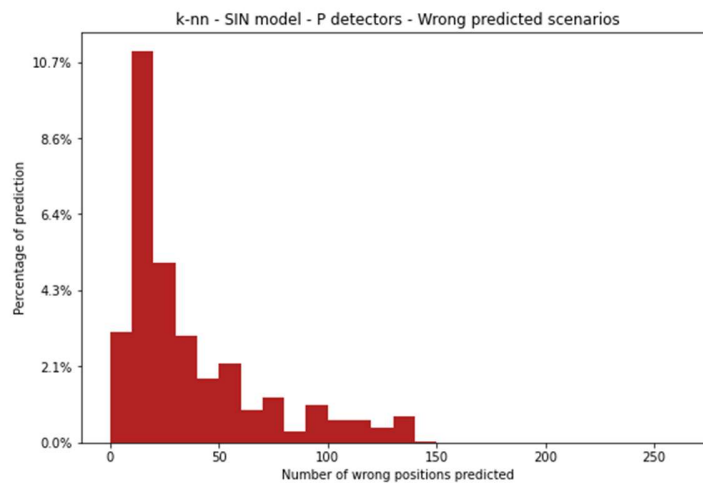


Figure 4.13. k-NN SIN model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted scenarios

The same analysis has been performed also for the ADI model, whose summarized results are reported in *Figure 4.14* for the wrong predicted scenarios. For this model, the number of predicted scenarios belonging to each category is listed below:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 537/963 (55.8%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 426/963 (44.2%);

having a total percentage of accuracy of 81.7%.

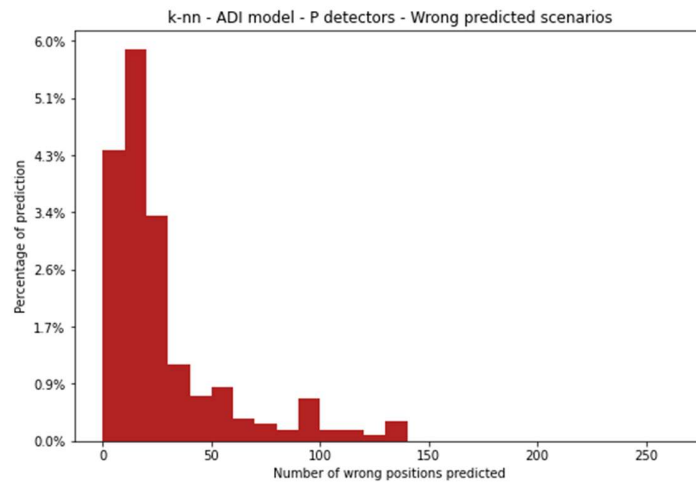


Figure 4.14. k-NN ADI model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios

Finally, same observations were made on the STD model, which analyses are presented in Figure 4.15. In this last case, the number of predicted scenarios pertaining to each category are:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 255/963 (26.5%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 708/963 (73.5%);

having a total percentage of accuracy of 69.7%.

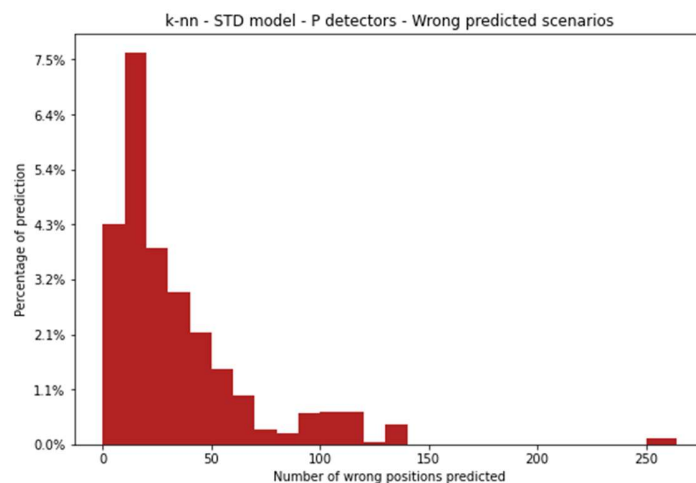


Figure 4.15. k-NN STD model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios

The summary of the results above described, for the three different k-NN models, is reported in Table 4.4.

	SIN model	ADI model	STD model
Complete scenarios well predicted [%]	100	100	100
Complete scenarios wrong predicted [%]	0	0	0
Defect scenarios well predicted [%]	23.4	55.8	26.5
Defect scenarios wrong predicted [%]	76.6	44.2	73.5
Total accuracy [%]	69.4	81.7	69.7

Table 4.4. *k*-NN models: Location of dummy pins - Summarized results

A further investigation on the model which led to the higher accuracy has been integrated. In this case, the method of using the single input detectors and the difference of the single values with respect to the average of the pertaining group (the ADI model) has reached the highest accuracy. This model was indeed particularly sensitive to the locations of the replaced pin if the diversion geometry faced some asymmetrical distribution of the dummy pins, as in the case reported in *Figure 3.7, 3.8 and 3.9*. In similar diverted cases, the discrepancies among the single detectors and the average of the group of detectors are higher in the area of diverted pins, with respect to the untouched locations, leading to a higher contribution of the input features to the final prediction. The STD model has as well led to better results with respect to the raw input features provided by the original database, but since the standard deviation is related to all the detectors in a certain group, it may lose some of the additional information that the heterogeneity of the data could provide.

Considering the results from the ADI model, the configurations of the fuel assembly which have been misinterpreted more frequently, have been inspected. In detail: each configuration of defect scenario has 9 different subcases relying on the combinations of the values of IE and BU. In the *Figures from 4.16 to 4.35* are reported the defect scenario configurations which had the highest frequency of wrong positions predictions for a certain geometry configuration, i.e., 9/9 (the 9 subcases which combines IE and BU values, pertaining to a certain geometrical configuration, for a certain number of dummy pins). On the left-hand side the figures are reporting the real configurations, while on the right-hand side one of the 9 wrong predicted configurations is reported. On each fuel assembly configuration are superimposed the detectors responses for the photons current.

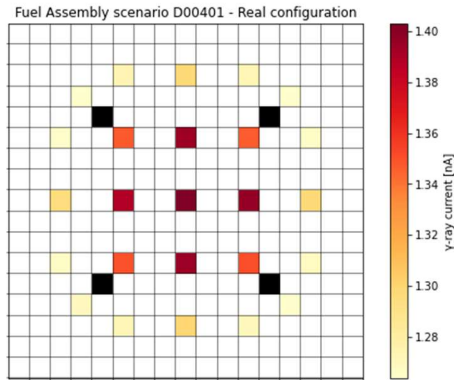


Figure 4.16. Defect scenario fuel assembly configuration – 1/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

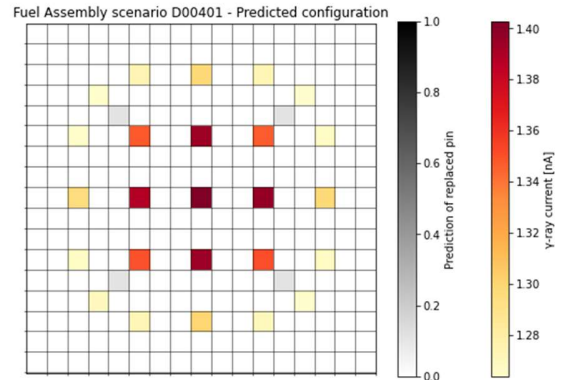


Figure 4.17. Defect scenario fuel assembly configuration – 1/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

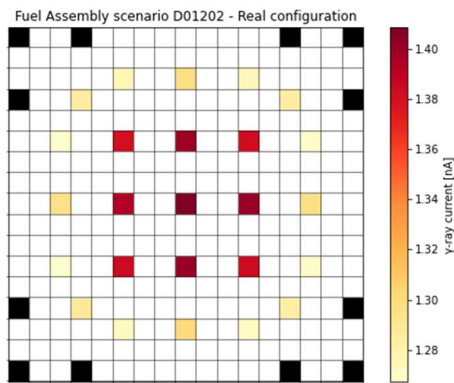


Figure 4.18. Defect scenario fuel assembly configuration – 2/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

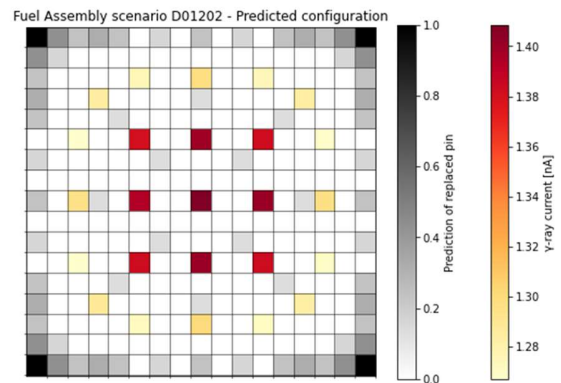


Figure 4.19. Defect scenario fuel assembly configuration – 2/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

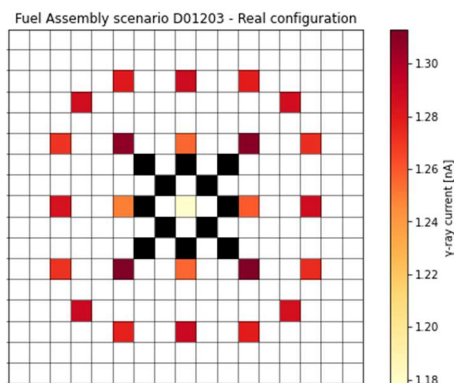


Figure 4.20. Defect scenario fuel assembly configuration – 3/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

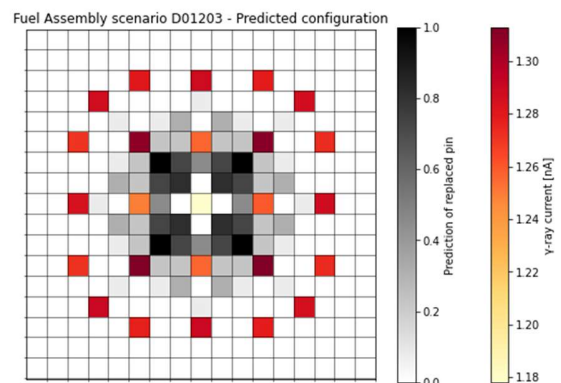


Figure 4.21. Defect scenario fuel assembly configuration – 3/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

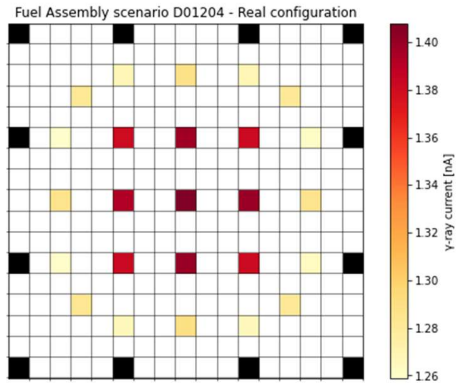


Figure 4.22. Defect scenario fuel assembly configuration – 4/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

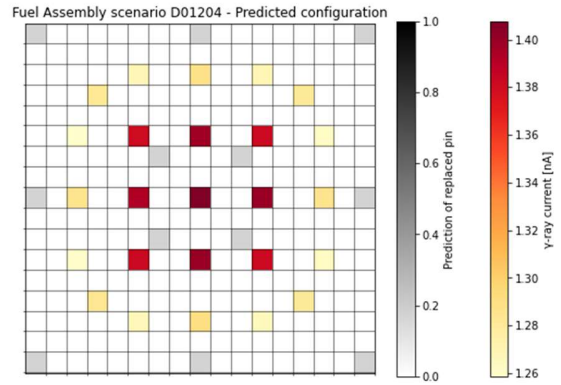


Figure 4.23. Defect scenario fuel assembly configuration – 4/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

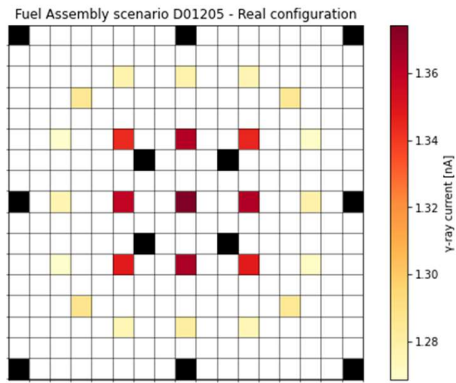


Figure 4.24. Defect scenario fuel assembly configuration – 5/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

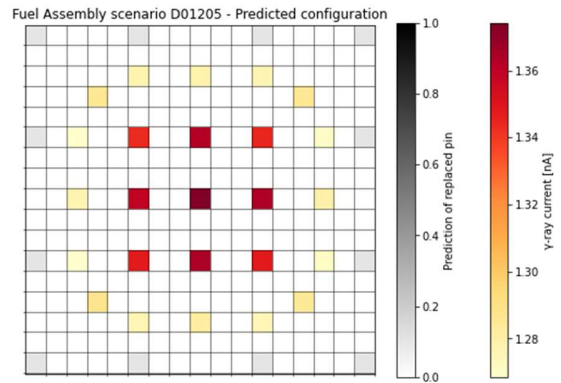


Figure 4.25. Defect scenario fuel assembly configuration – 5/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

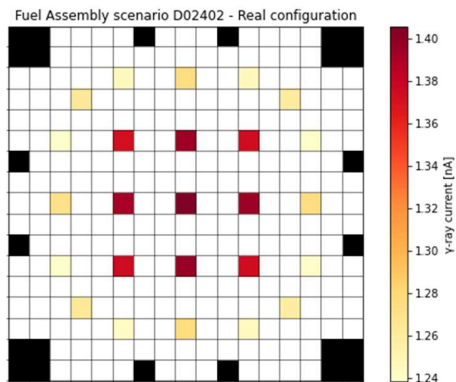


Figure 4.26. Defect scenario fuel assembly configuration – 6/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

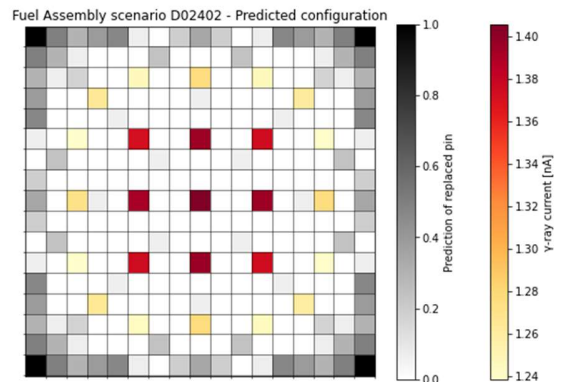


Figure 4.27. Defect scenario fuel assembly configuration – 6/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

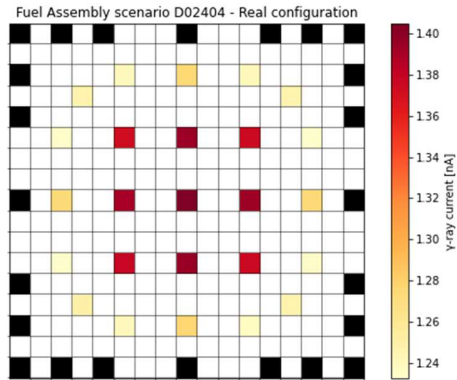


Figure 4.28. Defect scenario fuel assembly configuration – 7/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

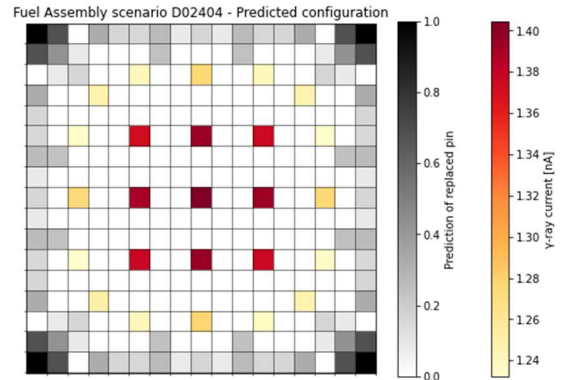


Figure 4.29. Defect scenario fuel assembly configuration – 7/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

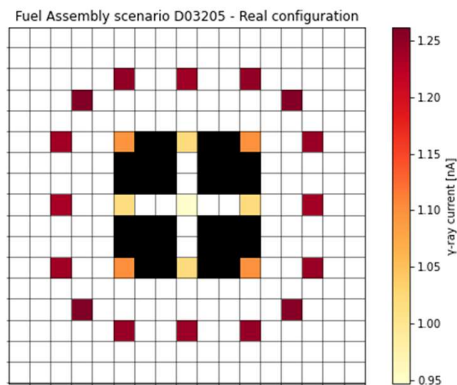


Figure 4.30. Defect scenario fuel assembly configuration – 8/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

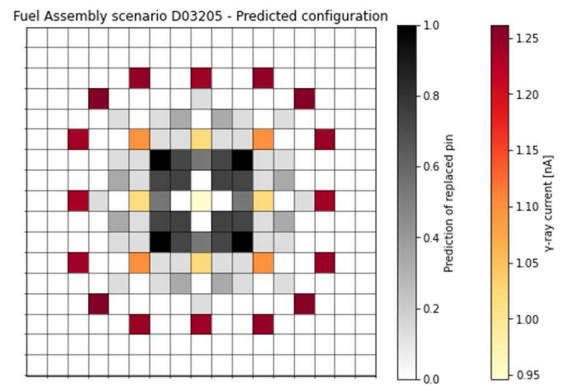


Figure 4.31. Defect scenario fuel assembly configuration – 8/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

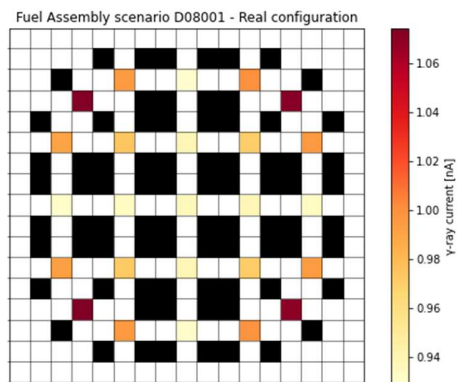


Figure 4.32. Defect scenario fuel assembly configuration – 9/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

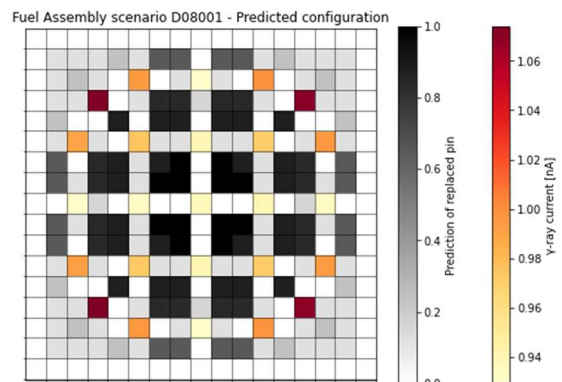


Figure 4.33. Defect scenario fuel assembly configuration – 9/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

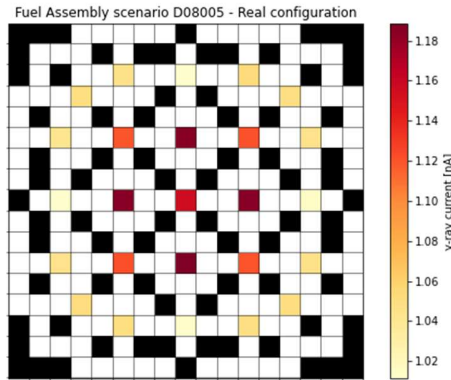


Figure 4.34. Defect scenario fuel assembly configuration – 10/10 Most frequent wrong interpreted scenarios - Real locations of dummy pins - Photons current

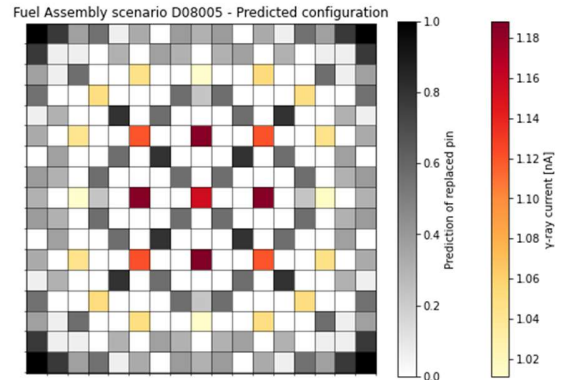


Figure 4.35. Defect scenario fuel assembly configuration – 10/10 Most frequent wrong interpreted scenarios - Predicted locations of dummy pins - Photons current

Form the worst predicted configurations above reported, it can be generally inferred that the most difficult lattices to predict are the ones presenting a chess-board distribution of the diverted pins on the geometry of the fuel assembly. Moreover, other difficulties of prediction were encountered more where diversions are performed on the outwards perimeter, and with a low number of replaced pins. This is an important insight for those which are the purposes of safeguards, since they give a hint on which could be the more challenging scenarios to predict, in a potential proliferative situation. Furthermore, considering the worst cases of 2nd, 3rd and from the 6th one to the 10th one, the graphs show that some diversions of the pins were actually predicted, but the locations of the replaced pins were wrongly established, meaning that eventually diverted situations would have been correctly identified by the model.

4.2 Neural Networks regression models

Concerning the neural networks regression algorithms, the outcomes of the analyses performed for the predictions of the number of dummy pins in the simulated scenarios and the locations of the dummy pins in the fuel assembly geometrical configurations are presented.

4.2.1 Number of dummy pins

Analogously to the precedent chapter, involving the results of the k-NN methods for the prediction of number of dummy pins, comparisons among evaluation metrics are here reported, retrieved at the end of the cross validation procedure. In the first place, the graphs representing the comparisons among RMSE of all different combinations of kinds and groups of detectors are reported in *Figure 4.36*, *Figure 4.37*, and *Figure 4.38*, for the operations on values: Sum, Ave&Std and Sin, respectively. Again, in the figures the mean value of the RMSE, depicted with the empty mark, is represented with the respective standard deviation, depicted as the vertical error bar. Generally, considering the same preprocessing operations on values, better results were achieved considering all groups of detectors in the same model. Additionally, as previously stated, the models which appear more sensitive to the pin diversions, i.e., implying lower values of RMSE, are the ones involving the photons current values, singularly or in combination with other detectors responses.

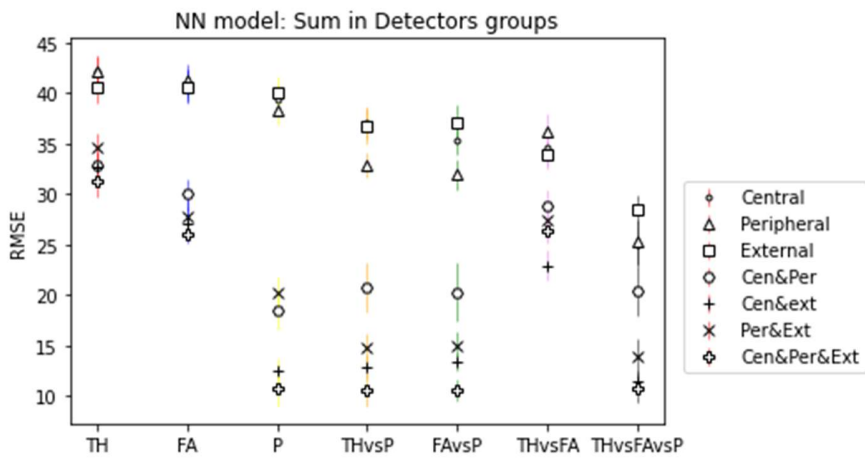


Figure 4.36. NN model RMSE - Sum in detectors groups for all combinations of kinds of detectors responses

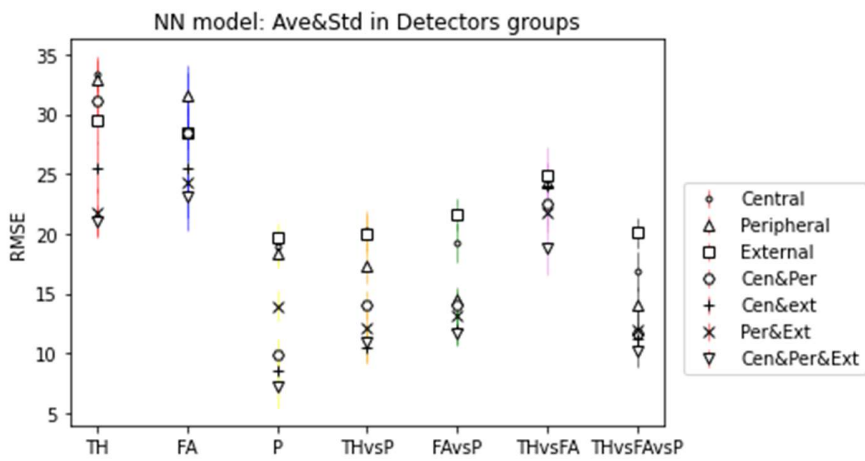


Figure 4.37. NN model RMSE - Average and Standard Deviation in detectors groups for all combinations of kinds of detectors responses

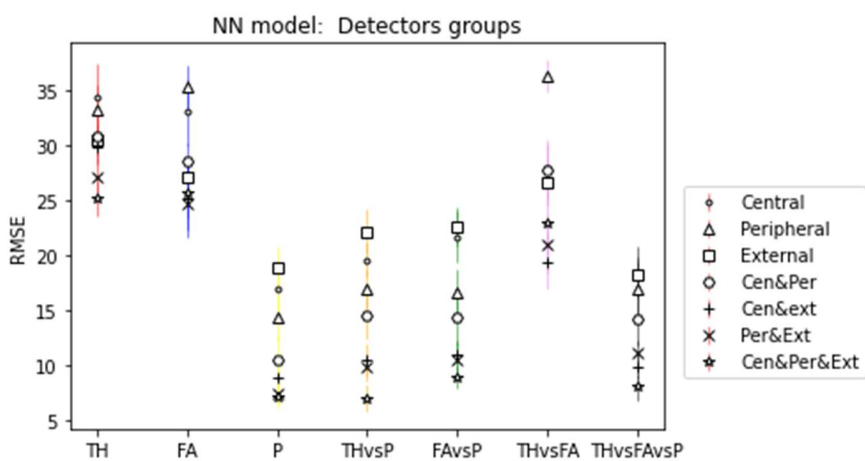


Figure 4.38. NN model RMSE - Single inputs of detectors groups for all combinations of kinds of detectors responses

A further investigation on the trend of the RMSE values for models considering all groups of detectors is presented in *Figure 4.39*, comparing different operations and different combinations of kinds of detectors responses. Again, the higher performance of the models which supplied photons current as input features can be inferred, beyond those in combination with other detectors. Focusing instead on the outcomes related to different operations on values, different trends were faced in case photons current were indeed included as inputs or not. Specifically, both thermal and fast neutrons counts inputs reached lower mean values of RMSE when the values were treated considering their average and standard deviation, while all the combinations including photons current reacted better RMSE when their values were considered singularly. The attitude of the latter could be an effect of the architecture of the neural networks, where the weights associated to the photons current values were enhanced with respect to the ones of the neutrons counts, such that the overall structure was following more closely the γ -ray responses. On the other hand, involving the neutrons counts, less sensitive to the diversions, consider the standard deviation of the values may help in the overall estimation of the diverted pins.

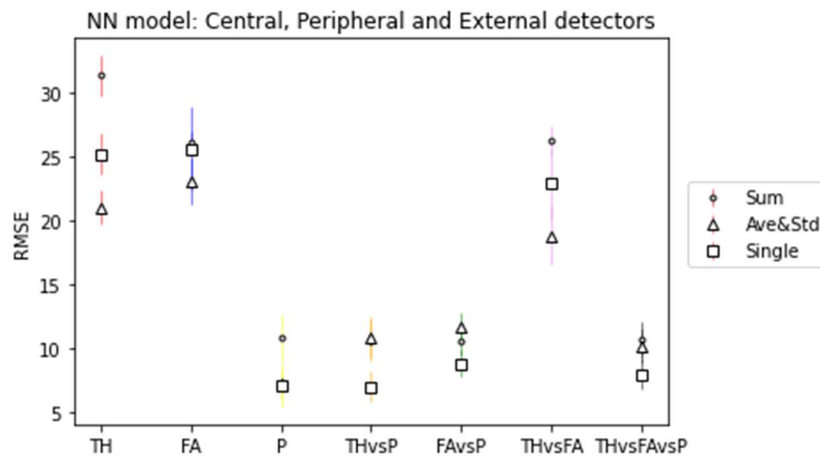


Figure 4.39. NN model RMSE - Operations on values within all detectors groups for all combinations of kinds of detectors responses

Moreover, the Delta, i.e., the difference between the predicted and exact values of number of dummy pins, of the models accounting for all the groups of detectors in the case of the most accurate operations on input features (Ave&Std and Sin) are presented in *Figure 4.40* and *Figure 4.41* respectively. The histograms have been built as introduced in the corresponding paragraph for the k-NN models. Even in these reported cases, the bin covering the largest number of predictions is the one ranging in a Delta from -1 to 1, implying that the majority of the misinterpreted predictions were actually small discrepancies. Dissimilarities on the outcomes can be inferred between Ave&Std and Sin operations for the input values of the neutrons counts, where the Ave&Std operation conducted to better estimations. On the other side, when the latter are compared to the results obtained with the involvement of the photons current, their lower performances are evident. More examples, for the different combinations of detectors groups, are reported in [Appendix B](#).

NN model: Central, Peripheral and External detectors - Ave&Std

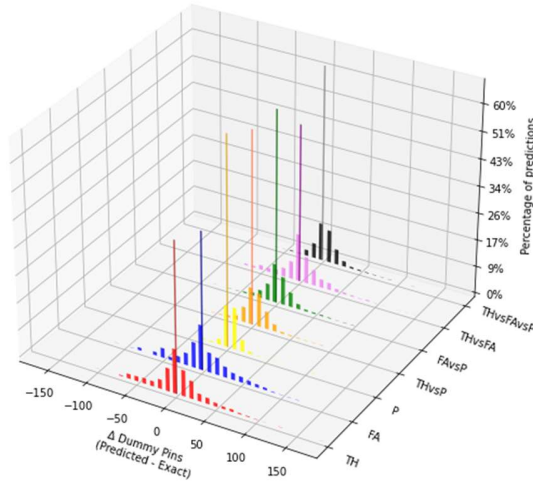


Figure 4.40. NN model Δ - Ave&Std within all detectors groups for all combinations of kinds of detectors responses

NN model: Central, Peripheral and External detectors

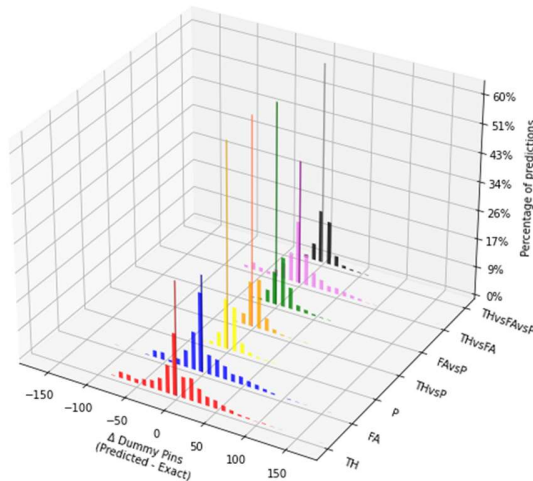


Figure 4.41. NN model Δ - Single central, peripheral, and external detectors values for all combinations of kinds of detectors responses

Eventually, the mean values and the standard deviation of RMSE for the different implemented models, considering Sum, Ave&Std and Sin models, are reported in *Table 4.5*, *Table 4.6*, and *Table 4.7* respectively. The models which best predict within each group or combination of groups are highlighted, and the best one for each preprocessing operation is then selected below. Generally, better performances are encountered in the models involving Ave&Std and Sin pre-processing operations, which values of mean RMSE are indeed comparable. Besides, the best prediction model is the one referring to the single inputs of photons current and thermal neutrons counts values considering all detectors locations, which returned a mean RMSE equal to 7.0 and a standard deviation of 1.2.

S U M	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT		
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	
		TH	42.2	1.4	42.2	1.5	40.6	1.7	32.9	1.4	32.7	1.4	34.5	1.5	31.3	1.6
FA	40.8	1.6	41.3	1.6	40.7	1.7	30.0	1.5	27.1	1.6	27.8	1.7	26.1	1.0		
P	39.3	1.4	38.4	1.4	40.1	1.5	18.5	2.0	12.5	1.2	20.3	1.5	10.8	1.8		
THvsP	37.1	1.6	32.9	1.3	36.8	1.8	20.8	2.4	12.9	1.1	14.7	1.5	10.6	1.6		
FAvsP	35.4	1.5	32.0	1.5	37.1	1.7	20.3	3.0	13.4	0.9	15.0	1.4	10.6	1.1		
THvsFA	34.7	1.7	36.2	1.8	33.9	1.4	28.8	1.5	22.9	1.5	27.4	1.9	26.3	1.1		
THvsFAvsP	25.0	2.0	25.2	2.3	28.4	1.5	20.5	2.5	11.4	1.1	13.8	1.9	10.7	1.4		
min in kinds:	25.0	2.0	25.2	2.3	28.4	1.5	18.5	2.0	11.4	1.1	13.8	1.9	10.6	1.6		
min in operations:	10.6								1.6							

Table 4.5. NN models: Mean RMSE and standard deviation – Operation on values: Sum

A V & S T D	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT		
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	
		TH	33.3	1.4	32.9	1.7	29.5	2.0	31.1	1.3	25.5	2.1	21.8	2.0	21.0	1.3
FA	31.4	2.7	31.6	1.9	28.4	1.2	28.5	2.4	25.4	5.1	24.3	1.5	23.1	1.8		
P	19.0	1.3	18.4	1.2	19.6	1.3	9.8	1.4	8.6	1.4	14.0	1.3	7.2	1.7		
THvsP	20.3	1.6	17.3	1.5	20.0	1.6	14.1	1.2	10.4	1.3	12.1	1.8	10.9	1.6		
FAvsP	19.2	1.6	14.4	1.1	21.6	1.4	14.1	1.1	12.0	1.2	13.1	1.3	11.7	1.1		
THvsFA	24.0	3.3	24.3	1.5	24.9	1.1	22.5	1.5	24.0	1.9	21.8	1.6	18.8	2.2		
THvsFAvsP	16.8	1.6	14.1	1.4	20.1	1.3	11.7	1.0	11.1	1.3	11.9	1.8	10.2	1.3		
min in kinds:	16.8	1.6	14.1	1.4	19.6	1.3	9.8	1.4	8.6	1.4	11.9	1.8	7.2	1.7		
min in operations:	7.2								1.7							

Table 4.6. NN models: Mean RMSE and standard deviation – Operation on values: Average and Standard Deviation

S I N G L E	Kind of detectors	CENTRAL		PERIPHERAL		EXTERNAL		CENTRAL&PERIPHERAL		CENTRAL&EXTERNAL		PERIPHERAL&EXTERNAL		CEN&PER&EXT		
		RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	RMSE_mean	RMSE_std	
		TH	34.4	3.0	33.3	2.2	30.5	1.4	30.8	2.3	29.8	1.5	27.2	1.5	25.2	1.6
FA	33.0	3.0	35.3	2.0	27.1	1.3	28.5	1.6	25.2	3.6	24.7	1.2	25.6	3.3		
P	16.9	2.2	14.3	2.1	18.9	2.0	10.4	2.3	8.9	1.0	7.4	0.9	7.1	1.0		
THvsP	19.6	1.6	16.9	1.7	22.2	2.0	14.4	2.0	10.5	1.5	9.7	1.3	7.0	1.2		
FAvsP	21.7	2.3	16.7	2.1	22.6	1.7	14.3	2.5	10.9	1.4	10.4	1.6	8.8	1.0		
THvsFA	27.9	2.3	36.3	1.4	26.7	2.1	27.8	2.7	19.4	2.5	21.0	2.7	22.9	2.7		
THvsFAvsP	18.4	2.3	17.0	2.4	18.2	1.6	14.2	2.4	9.7	1.1	11.1	1.2	8.0	1.2		
min in kinds:	16.9	2.2	14.3	2.1	18.2	1.6	10.4	2.3	8.9	1.0	7.4	0.9	7.0	1.2		
min in operations:	7.0								1.2							

Table 4.7. NN models: Mean RMSE and standard deviation – Operation on values: Single input

4.2.2. Location of dummy pins

First, a qualitative representation of the outcomes of the models are presented. Referring to the same real configuration of replaced dummy pins introduced in *Figure 4.9*, the corresponding predicted outcomes of the three examined models are reported in *Figure 4.42*, *Figure 4.43* and *Figure 4.44* for the SIN, ADI and STD models respectively. Again, the locations of the detectors are reported in bordeaux, whereas the probability of having a replaced pin in a certain location relies on a dashed colorbar going from white (representing “0”, i.e., pin present) to black (representing “1”, i.e., pin replaced). Thus, the darker the square on the lattice, the higher the predicted probability of having a replaced pin.

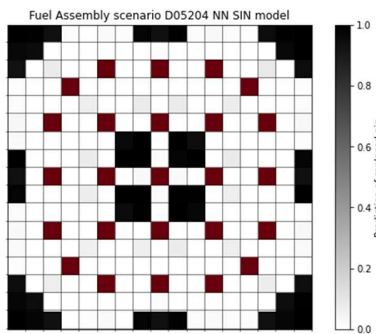


Figure 4.42. NN SIN model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

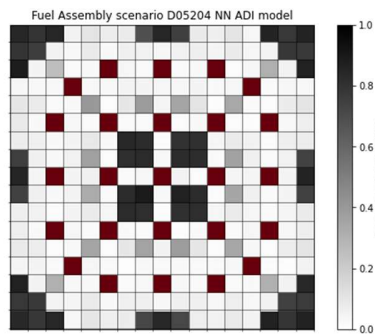


Figure 4.43. NN ADI model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

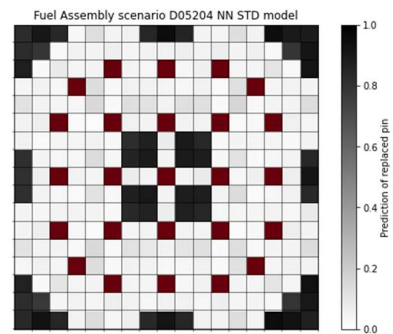


Figure 4.44. NN STD model: Defect scenario fuel assembly configuration - Predicted locations of dummy pins

Following the same procedure reported in the analogous paragraph involving the k-NN models, an analysis on the outcomes of these three models were investigated, and the results are discussed on the consequently graphs. On the abscissa, the number of wrong positions predicted for each scenario is reported, while on the ordinate the percentage of scenarios for which has been predicted the correspondent number of wrong positions, percentage calculated with respect to all scenarios.

In *Figure 4.45* the results for the SIN model are presented, which focus only on the wrong predicted scenarios, from which it can be inferred the percentage, out of all scenarios, of the number of wrong positions predicted in each scenarios investigated. A brief summary of the number of predicted scenarios belonging to each category is listed, in the case of SIN model:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 843/963 (87.5%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 120/963 (12.5%);

having a total percentage of accuracy of 94.7%.

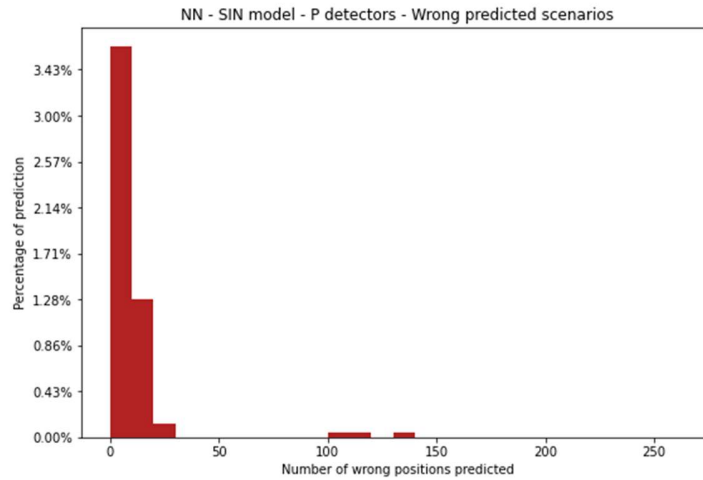


Figure 4.45. NN SIN model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted scenarios

The same analysis has been performed also for the ADI model, whose summarized results are reported in Figure 4.46 for the wrong predicted scenarios. For this model, the number of predicted scenarios belonging to each category is listed below:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 609/963 (63.2%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 354/963 (36.8%);

having a total percentage of accuracy of 84.8%.

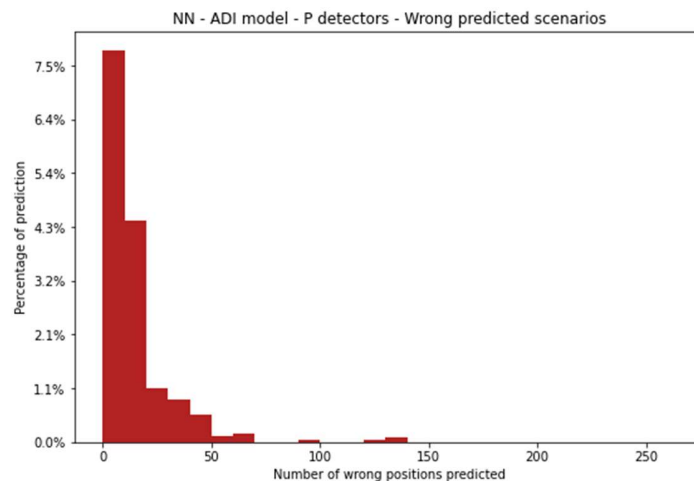


Figure 4.46. NN ADI model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios

Finally, same observations were made on the STD model, which analyses are presented in Figure 4.47. In this last case, the number of predicted scenarios pertaining to each category are:

- Number of complete scenarios whose dummy pin locations have been well predicted: 1372/1372;
- Number of complete scenarios whose dummy pin locations have been wrong predicted: 0/1372;
- Number of defect scenarios whose dummy pin locations have been well predicted: 917/963 (95.2%);
- Number of defect scenarios whose dummy pin locations have been wrong predicted: 46/963 (4.8%);

having a total percentage of accuracy of 97.5%.

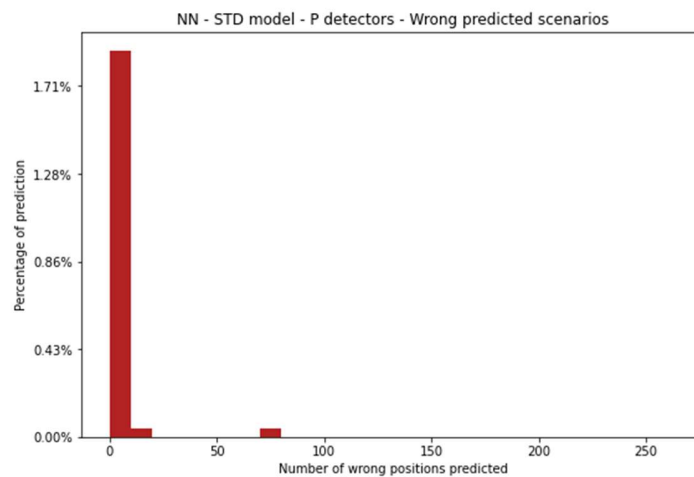


Figure 4.47. NN STD model: Percentage of prediction VS Number of wrong position predicted – Wrong predicted Scenarios

The summary of the results above described, for the three different NN models, is reported in Table 4.8.

	SIN model	ADI model	STD model
Complete scenarios well predicted [%]	100	100	100
Complete scenarios wrong predicted [%]	0	0	0
Defect scenarios well predicted [%]	87.5	63.2	95.2
Defect scenarios wrong predicted [%]	12.5	36.8	4.8
Total accuracy [%]	94.7	84.8	97.5

Table 4.8. NN models: Location of dummy pins - Summarized results

A further investigation on the model which led to the higher accuracy has been integrated. In this case, the method which considered the single input and the standard deviation within each group has reached the highest accuracy, besides also the other methods provided a good correctness. The STD model eventually joined the precision provided by the SIN model, having already high performance, and introduce some further information on the values which are more related to the distortions per sè. On the other hand, having provided 130 input features to the ADI model, could have led to an increase of the statistical noise of the intrinsic features of the dataset, inside a network which is already quite complex.

Considering the results from the STD model, frequently mis-interpreted geometrical configurations were not encountered, if it is considered the maximum number of 9 wrong predicted configurations out of 9, as reported in the previous correspondent paragraph. On the other hand, lowering the constraint of the maximum frequency and considering the geometrical configurations which have been incorrectly predicted half of the times (i.e., more than 4 configurations out of 9), some further information may be retrieved. In the *Figure* from 4.48 to 4.53 are reported the diverted scenarios which had a frequency of misinterpretation of more than half of the total number of scenarios for a certain geometrical configuration. On the left-hand side the figures are reporting the real configurations, while on the right-hand side one of the wrong predicted configurations is reported, with the number of times they have been misinterpreted, out of 9. On each fuel assembly configuration are superimposed the detectors responses for the photons current.

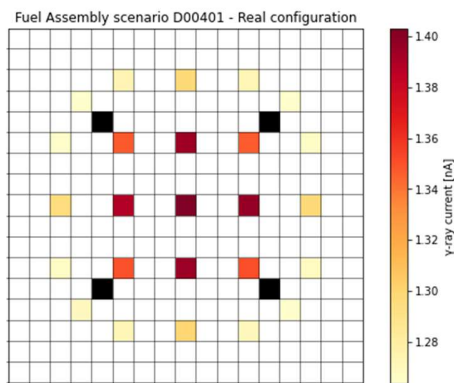


Figure 4.48. Defect scenario fuel assembly configuration – Wrong interpreted scenario 5/9 - Real locations of dummy pins - Photons current

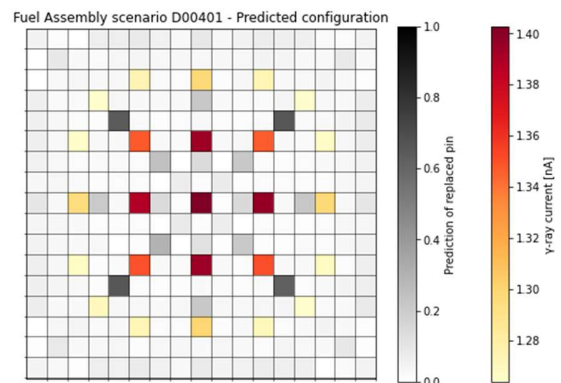


Figure 4.49. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9- Predicted locations of dummy pins - Photons current

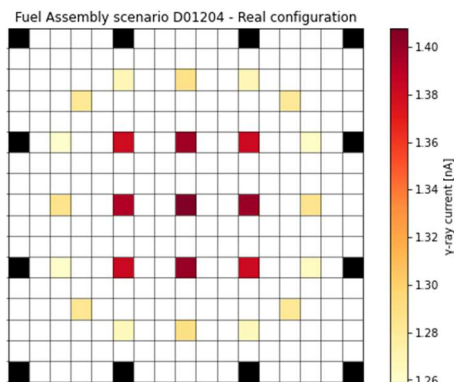


Figure 4.50. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 8/9- Real locations of dummy pins - Photons current

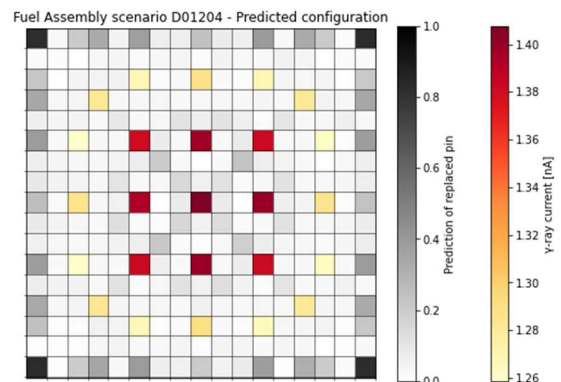


Figure 4.51. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 8/9- Predicted locations of dummy pins - Photons current

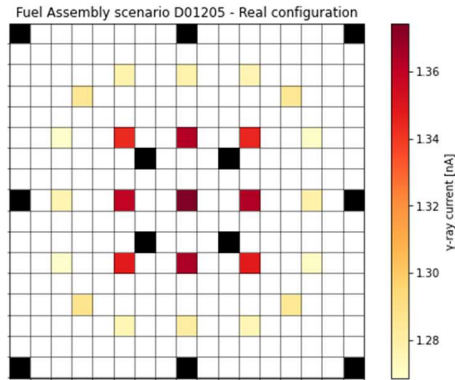


Figure 4.52. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9 - Real locations of dummy pins - Photons current

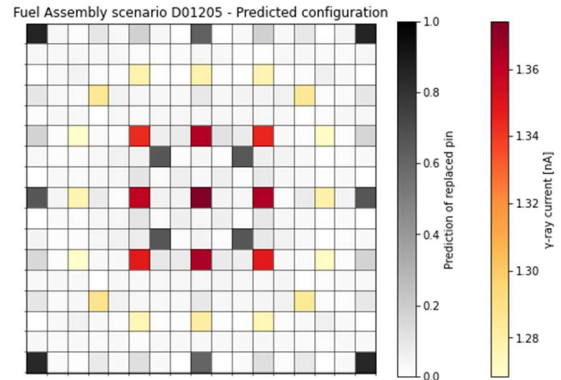


Figure 4.53. Defect scenario fuel assembly configuration – Wrong interpreted scenarios 5/9 - Predicted locations of dummy pins - Photons current

Even in this case, from the worst predicted configurations above reported, it can be generally inferred that the most difficult lattices to predict are the ones presenting a chess-board distribution of the diverted pins on the geometry of the fuel assembly, especially if their number is low and the replacements are located on the outwards perimeter. Moreover, the graphs show that some diversions of the pins were actually predicted, but the locations of the replaced pins were wrongly established, meaning that eventually diverted situations would have been correctly identified by the model.

5. CONCLUSIONS

The objective of this study was to develop machine learning models to detect the diversions of spent nuclear fuel assemblies, both investigating the number of pins replaced by steel dummies and the specific locations of the replacements on the 17x17 lattice, in a safeguards perspective. In particular, the thesis had the aim to further examine models which have been already investigated in a previous similar work, for extending the predictions using another prospective: regression models instead of classification ones.

The starting point was a database constituted by thousands of Monte Carlo simulated scenarios, calculating different detectors responses for both complete and defect fuel assemblies. The former covered a wide range of combination of radiation history parameters of the fuel, such as burnup, initial enrichment and cooling time, whereas the defect scenarios were tuned on reduced number of factors, but on extensive range of numbers of dummy pins and geometrical configurations of the replacements on the lattice. The detectors responses were simulated on the base of the prototype of the NDA technique, the so-called PDET, which employed detectors sensitive to thermal and fast neutrons, and γ -rays in several locations on a spent fuel assembly.

The simulated detectors responses were used in different combinations as input features of the machine learning algorithms, for the predictions of some outcomes of interest, namely the number of dummy pins and their location on the grid. For both purposes were considered two ML models: the k-nearest neighbors and the neural networks, configured on the same input features to eventually confront their performances. Comparisons at different levels could be asserted on the processing, the methodologies and the databases employed.

Focusing on the detectors responses supplied in the input phase of the procedure, better predictions were reached when all detectors locations were considered simultaneously, for any method and expected outcome. Moreover, the detectors which appeared more sensitive to the diversions, thus leading to larger accuracy of the models results for the prediction of the number of dummy pins, were the γ -ray detectors. When only photons current values were used in k-NN models, the best results were generally obtained, whereas as input of NN models better performances were reached in combination with other detectors responses values. The structures of the two methods are indeed intrinsically different: whether the k-NN uses just some k observations for training the models, the NN relies basically on all provided information, giving different relevance to the samples depending on their contribution on the prediction. This mostly results on the amount of the information a model can retain, since the former has to choose which data preserve and which discharge, even if they might be useful on the final prediction, whereas the latter could press the advantage of considering all the features provided, associating suitable weights.

Regarding the preprocessing methods that were integrated in the input phase, preserving the heterogeneity of the values of the dataset typically conducted to better accuracies of the predictions. Specifically, considering the sum within groups of detectors performed the worst estimations, although the single detectors inputs and average within groups and their standard deviation conducted to similar outcomes. This effect was more visible in the models based on NN with respect to k-NN, probably due to

the amount of data that the networks could handle, and properly tune, against k-nearest neighbors methods. On the other hand, the complex structure of neural networks could lead to problems of overfitting of the data, if the configuration is not properly tuned and adjusted on the input features. This could be one of the reasons why the best predictive model has been encountered using the k-NN, even if better overall estimations were encountered by means of the neural networks.

Preserving the heterogeneity of the values was effective also in the phase concerning the prediction of the location of the dummy pins. In this case, the greater the amount and the diversification of the information provided to the model, the more proficient was the latter, where focusing on the enhancement of the discrepancies proper of the diversions was efficient. In this case, better results were achieved in the configurations relying on neural networks methods, reaching almost the perfect superimposing of data and predictions, nevertheless a tradeoff between amount of information and model complexity must be foreseen.

An additional comment should be introduced on the number of models investigated in this work. In order to further improve and increase the efficiency of the models, the idea of first investigate a large number of models and then select the best ones, for further and more precise tuning, could be an efficacious option. Indeed, in this work it has been decided to investigate a wide range of models in the stage of prediction of number of dummy pins, while just the best configuration was later used for the investigation of the location of the replacements. This has resulted in a lower computational expense of the second stage, and the possibility to deeply tune the hyperparameters required, on the data selected, especially in the development of the neural networks algorithms, which has then reached 97.5% of accuracy.

In conclusion, the machine learning models implemented in this study showed very promising results for the detection of partial defects of spent fuel assemblies. Very accurate and precise predictions were achieved developing neural networks algorithms in the replacement locations, whereas overall good estimations of the diversions were encountered for the purpose of revealing proliferation actions even implementing the less complex k-nearest neighbors architectures. Future works could introduce further investigation on the preprocessing phase of the input features, to focalize on the source of distortions with respect to the complete scenarios, providing more effective information to the models.

REFERENCES

- Bottou, L. (1991). *Stochastic gradient learning in neural networks*. Proceedings of Neuro-Nimes, 91.8: 12.
- Chollet, F., et al. (2015). *Keras*.
- ESARDA (2008). *Nuclear Safeguards and Non-Proliferation – Syllabus of the ESARDA Course*.
- Giani, N. (2019). *Application of machine learning algorithms to the safeguards verification of spent nuclear fuel*. Master thesis, Politecnico di Torino.
- Ham, Y., Kerr, P., Sitaraman, S., Swan, R., Rossa, R., & Liljenfeldt, H. (2015). *Partial defect verification of spent fuel assemblies by PDET: Principle and field testing in Interim Spent fuel Storage Facility (CLAB) in Sweden*. 4th International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA).
- Hargraves, R., & Moir, R. (2010). *Liquid Fluoride Thorium Reactors*. American Scientist, Volume 98, 304-313.
- Hastie, T., Tibshirani, R., Friedman, J. (2017). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Editor.
- Hyndman, R., Koehler, A. (2006). *Another look at measures of forecast accuracy*. International Journal of Forecasting, 22, 679-688.
- IAEA (2002). *IAEA Safeguards Glossary. 2001 Edition*.
- IAEA (2011). *Safeguards techniques and equipment: 2011 Edition*. International Nuclear Verification Series No. 1 (Rev. 2).
- IAEA (2018). *International Safeguards in the design of facilities for long term spent fuel management*. IAEA Nuclear Energy Series No. Nf-T-3.1.
- IAEA (2022). *Artificial Intelligence for accelerating Nuclear Applications, Science and Technology*. In I. A. E. Agency (Ed.). Vienna: IAEA.
- INFCIR/153 (1972). *The structure and content of agreements between the Agency and States required in connection with the treaty on the Non-Proliferation of Nuclear weapons*.
- IAEA (1989). IAEA Statute.
- Jagtap, A. D., Karniadakis, G.E. (2022). *How important are activation functions in regression and classification? A survey, performance comparison, and future directions*. arXiv preprint arXiv:2209.02681.
- Kingma, D.P., Ba, J.L. (2015). *Adam: A method for stochastic optimization*. ICLR 2015.
- Li, L., Jamieson K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. (2018). *Hyperband: A novel Bandit- Based Approach to Hyperparameter Optimization*. Journal of Machine Learning Research, 185, 1-52.
- Murphy, K. P. (2012). *Machine Learning: a probabilist prespective*. Editor: MIT press Ltd.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.
- Relly, D., Enselin, N., Smith, H.Jr. (1991). *Gamma-Ray Detectors*. In S. Kreiner (Ed.), *Passive Nondestructive Assay of Nuclear Materials*. Office of nuclear regulatory research U.S. Nuclear Regulatory Commission, 43-64.
- Rossa, R., Borella, A., Giani, N. (2020). *Comparison of machine learning models for the detection of partial defects in spent nuclear fuel*. Annals of Nuclear Energy, 147.

- Rossa, R., Borella, A., Van Der Meer, K. (2013). *Development of a reference spent fuel library of 17x17 PWR fuel assemblies*. ESARDA Bulletin No 49, 27-39.
- Rossa, R., Borella, A. (2019). *Use of machine learning models for the detection of fuel pin replacement in spent fuel assemblies*. ESARDA Bulletin No 58, 22-34.
- Rossa, R., Borella, A. (2021). *Detection of partial defects in spent fuel assemblies with the help of machine learning*. INMM Annual Meeting Proceedings.
- Smith L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1—learning rate, batch size, momentum, and weight decay*. arXiv preprint arXiv:1803.09820.

APPENDIX A: DATABASE VISUALIZATION

In this section are reported the additional graphs related to the database visualization presented in [Paragraph 2.3](#). Central, peripheral, and external detectors are treated separately, for each kind of detector response. The plots produced for the sum of the values within each group are reported in the first part, while the second is dedicated to the average within each group. The sequence of the figure follows the structure presented in the body text.

Sum in central detectors

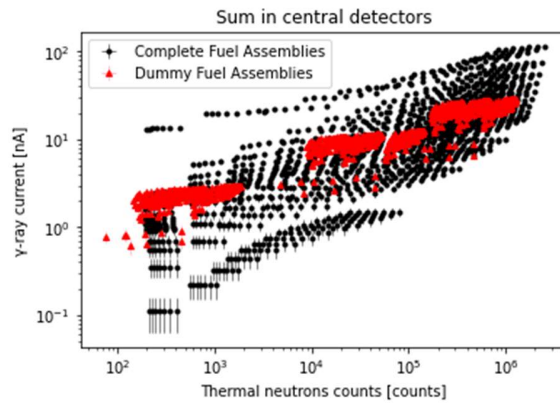


Figure A.1. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Thermal neutrons counts and photons current

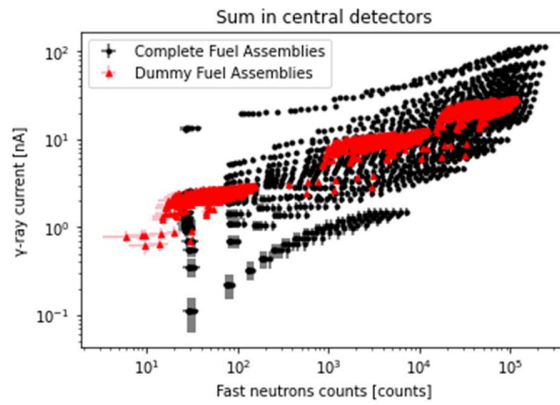


Figure A.2. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Fast neutrons counts and photons current

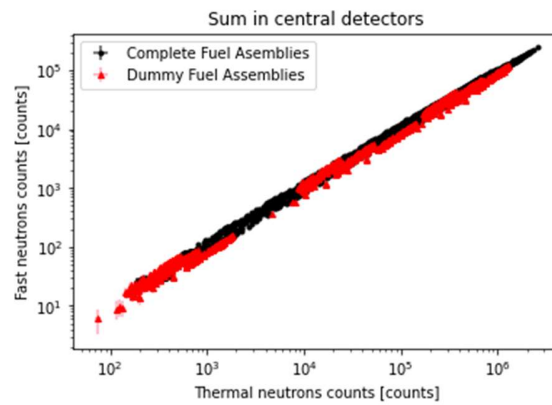


Figure A.3. Database visualization: Complete and defect scenarios - Sum in central detectors and relative uncertainties – Thermal and fast neutrons counts

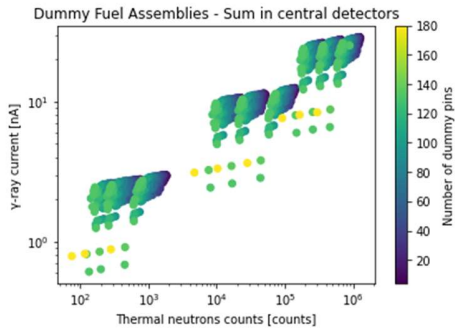


Figure A.4. Database visualization: Defect scenarios - Sum in central detectors - Thermal neutrons counts and photons current - Number of dummy pins

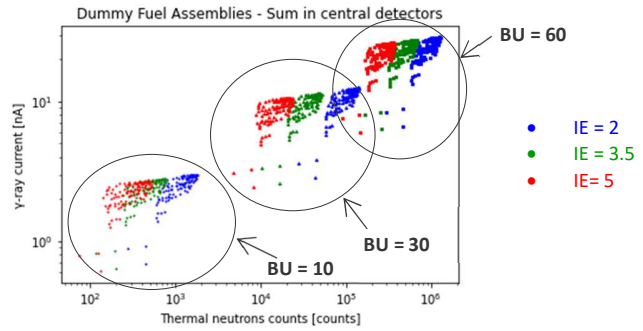


Figure A.5. Database visualization: Defect scenarios – Sum in central detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp

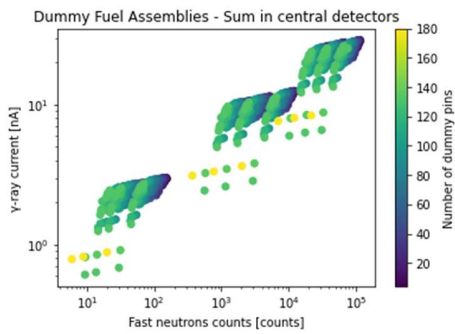


Figure A.6. Database visualization: Defect scenarios – Sum in central detectors - Fast neutrons counts and photons current - Number of dummy pins

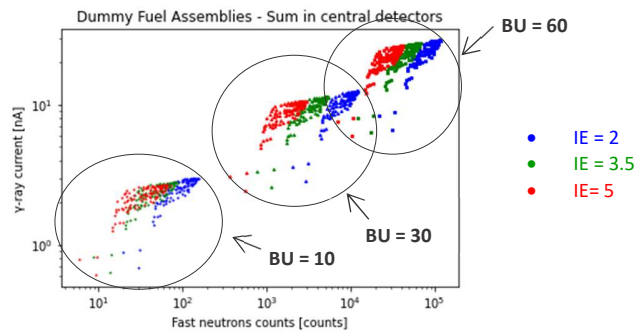


Figure A.7. Database visualization: Defect scenarios – Sum in central detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp

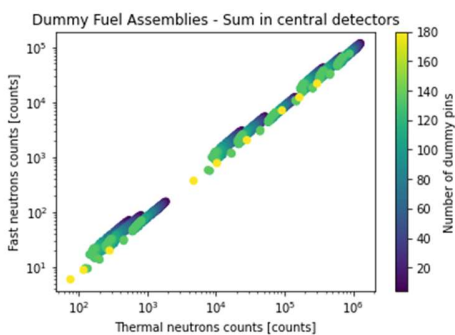


Figure A.8. Database visualization: Defect scenarios – Sum in central detectors - Thermal and fast neutrons counts - Number of dummy pins

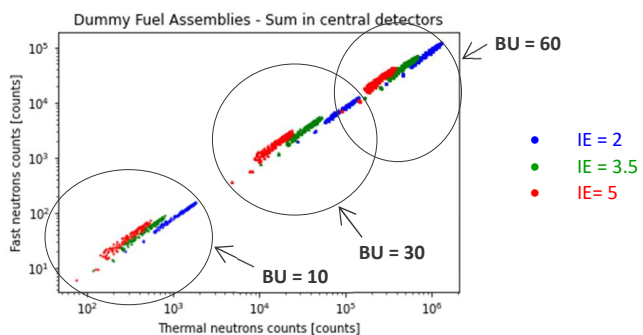


Figure A.9. Database visualization: Defect scenarios – Sum in central detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp

Sum in peripheral detectors

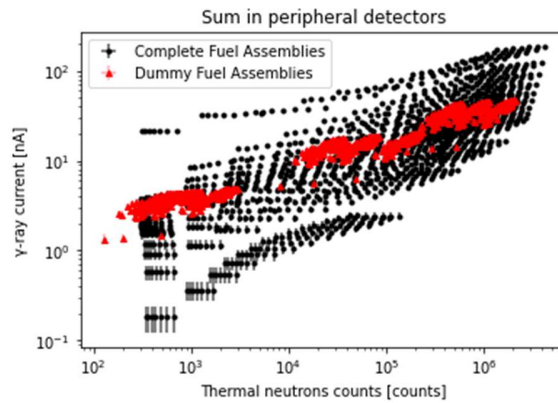


Figure A.10. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Thermal neutrons counts and photons current

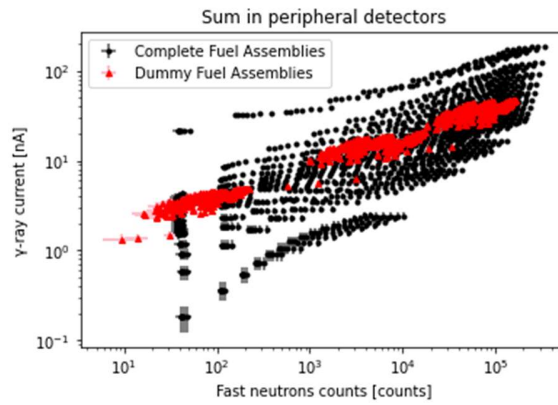


Figure A.11. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Fast neutrons counts and photons current

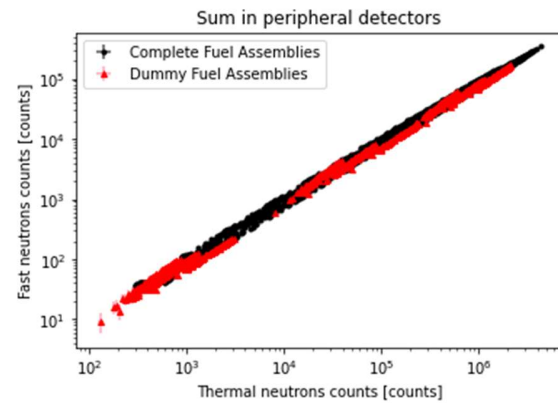


Figure A.12. Database visualization: Complete and defect scenarios - Sum in peripheral detectors and relative uncertainties – Thermal and fast neutrons counts

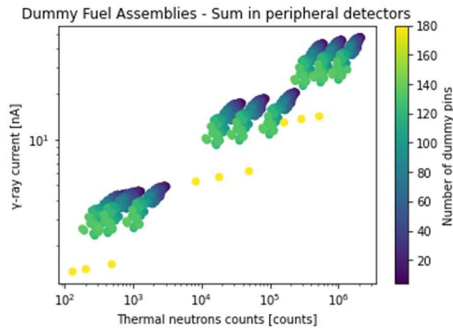


Figure A.13. Database visualization: Defect scenarios - Sum in peripheral detectors - Thermal neutrons counts and photons current - Number of dummy pins

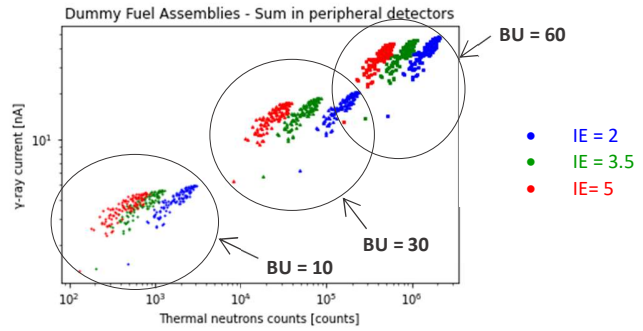


Figure A.14. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp

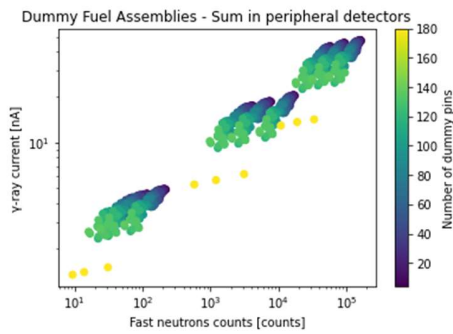


Figure A.15. Database visualization: Defect scenarios – Sum in peripheral detectors - Fast neutrons counts and photons current - Number of dummy pins

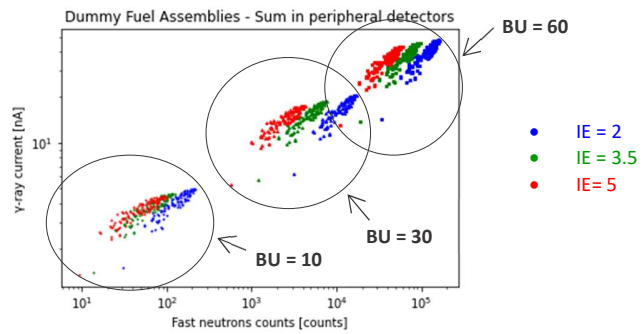


Figure A.16. Database visualization: Defect scenarios – Sum in peripheral detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp

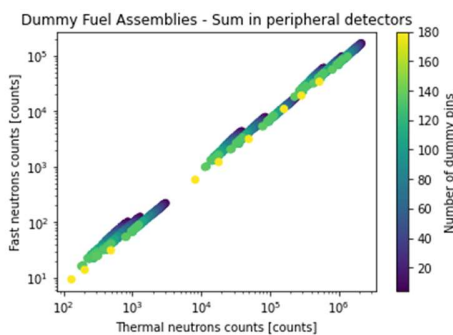


Figure A.17. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal and fast neutrons counts - Number of dummy pins

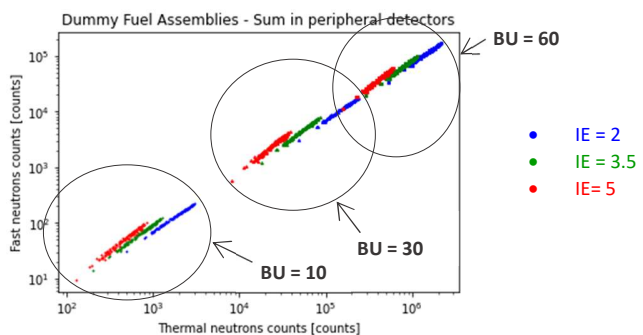


Figure A.18. Database visualization: Defect scenarios – Sum in peripheral detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp

Sum in external detectors

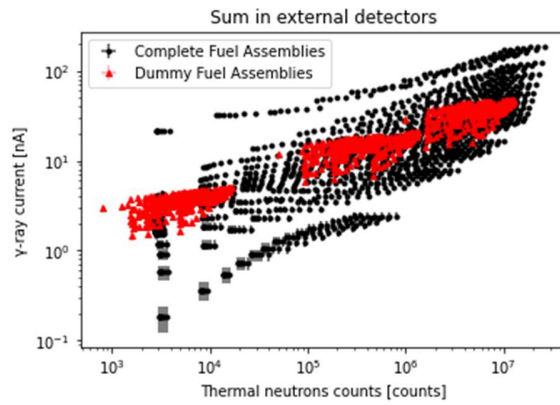


Figure A.19. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Thermal neutrons counts and photons current

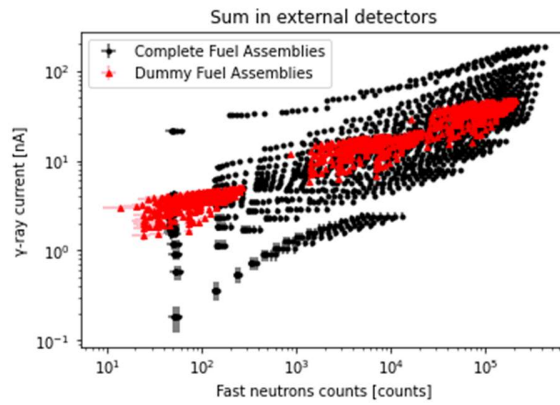


Figure A.20. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Fast neutrons counts and photons current

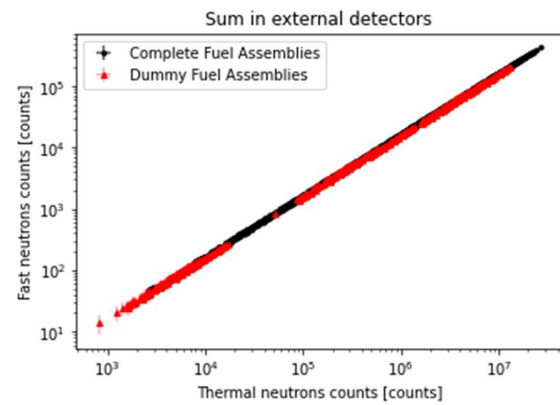


Figure A.21. Database visualization: Complete and defect scenarios - Sum in external detectors and relative uncertainties – Thermal and fast neutrons counts

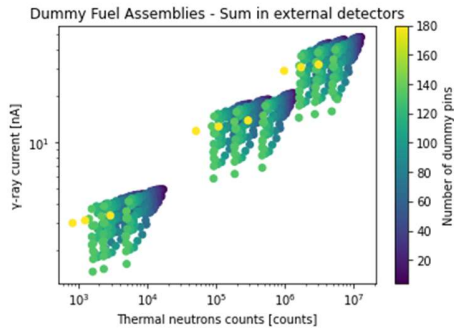


Figure A.22. Database visualization: Defect scenarios - Sum in external detectors - Thermal neutrons counts and photons current - Number of dummy pins

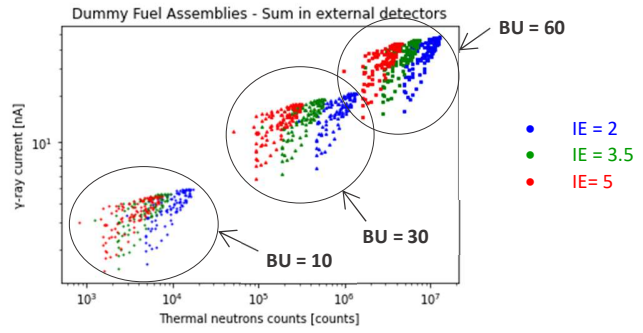


Figure A.23. Database visualization: Defect scenarios – Sum in external detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp

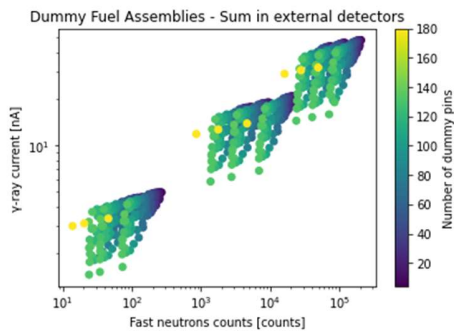


Figure A.24. Database visualization: Defect scenarios – Sum in external detectors - Fast neutrons counts and photons current - Number of dummy pins

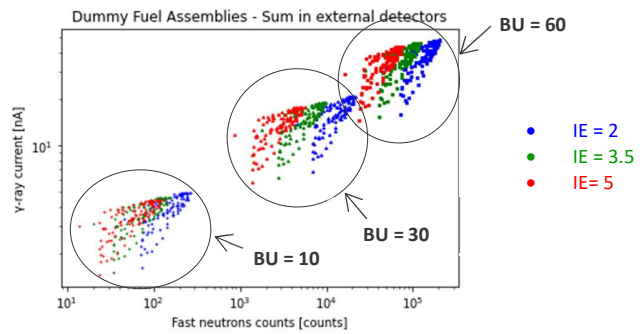


Figure A.25. Database visualization: Defect scenarios – Sum in external detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp

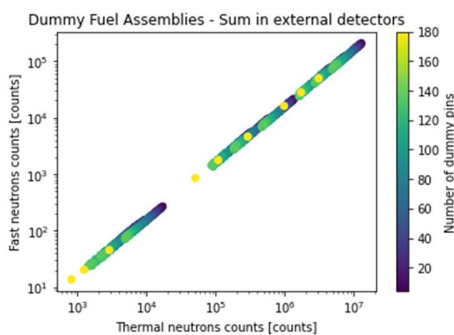


Figure A.26. Database visualization: Defect scenarios – Sum in external detectors - Thermal and fast neutrons counts - Number of dummy pins

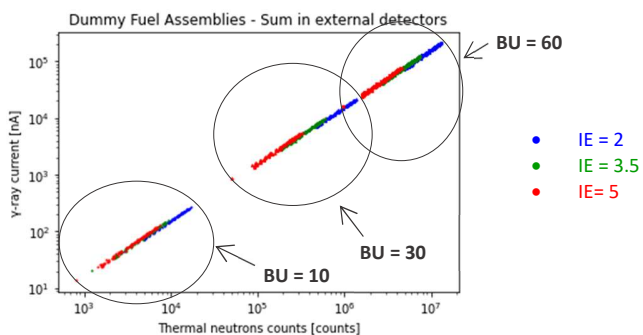


Figure A.27. Database visualization: Defect scenarios – Sum in external detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp

Average in central detectors

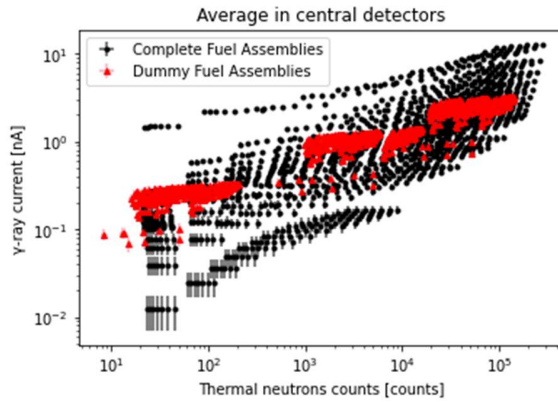


Figure A.28. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Thermal neutrons counts and photons current

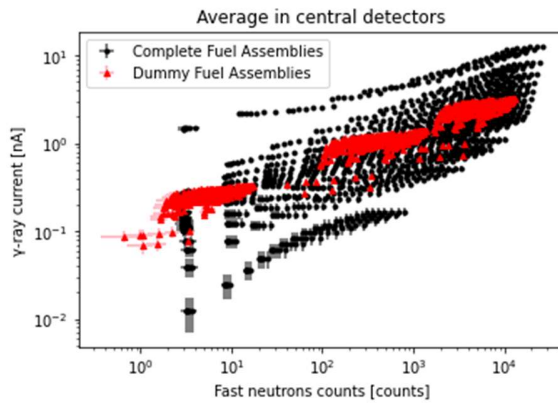


Figure A.29. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Fast neutrons counts and photons current

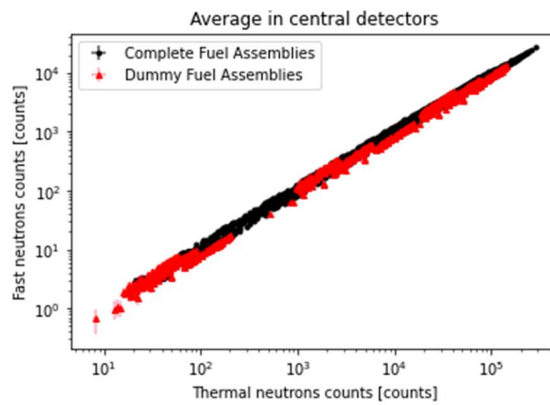


Figure A.30. Database visualization: Complete and defect scenarios - Average in central detectors and relative uncertainties – Thermal and fast neutrons counts

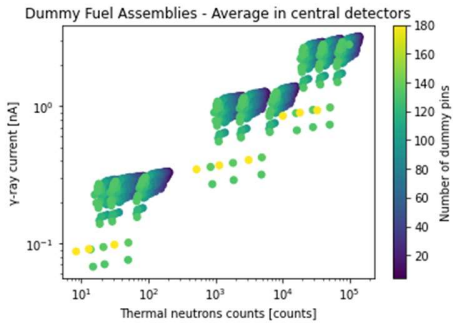


Figure A.31. Database visualization: Defect scenarios - Average in central detectors - Thermal neutrons counts and photons current - Number of dummy pins

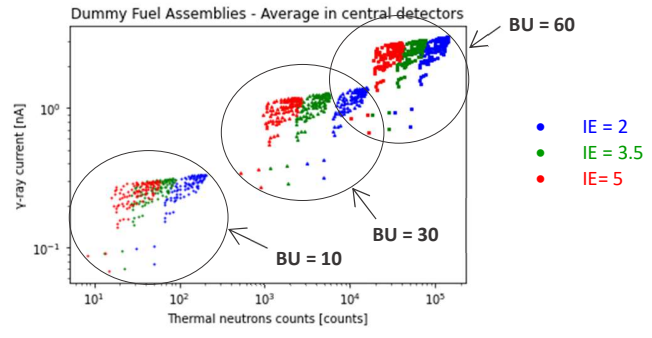


Figure A.32. Database visualization: Defect scenarios - Average in central detectors - Thermal neutrons counts and photons current - Initial Enrichment and BurnUp

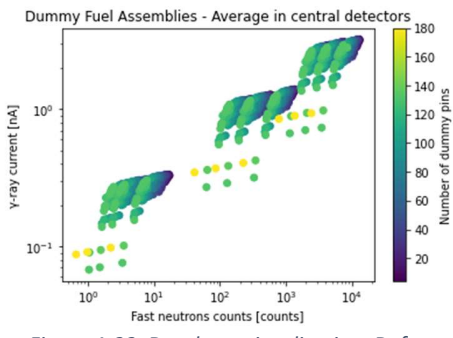


Figure A.33. Database visualization: Defect scenarios - Average in central detectors - Fast neutrons counts and photons current - Number of dummy pins

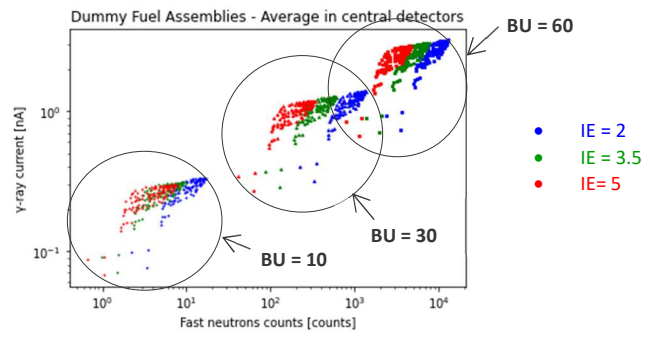


Figure A.34. Database visualization: Defect scenarios - Average in central detectors - Fast neutrons counts and photons current - Initial Enrichment and BurnUp

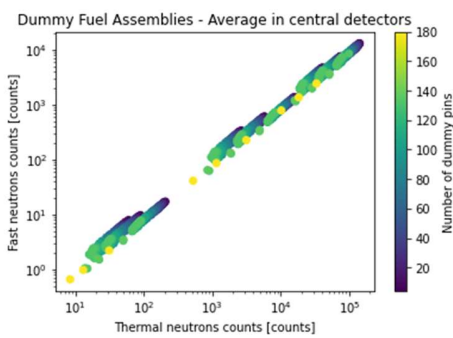


Figure A.35. Database visualization: Defect scenarios - Average in central detectors - Thermal and fast neutrons counts - Number of dummy pins

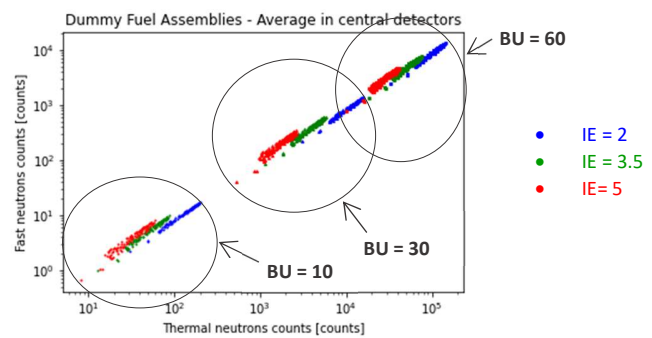


Figure A.36. Database visualization: Defect scenarios - Average in central detectors - Thermal and fast neutrons counts - Initial Enrichment and BurnUp

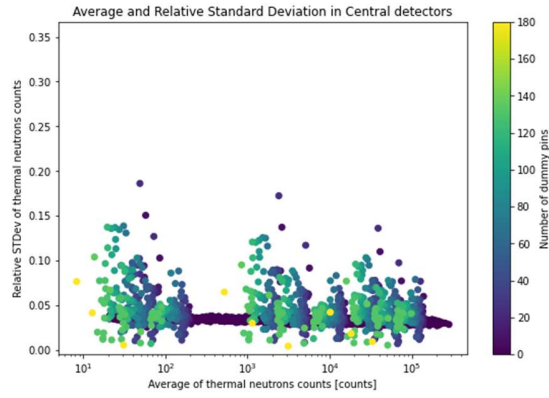


Figure A.37. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Thermal neutrons counts - Number of dummy pins

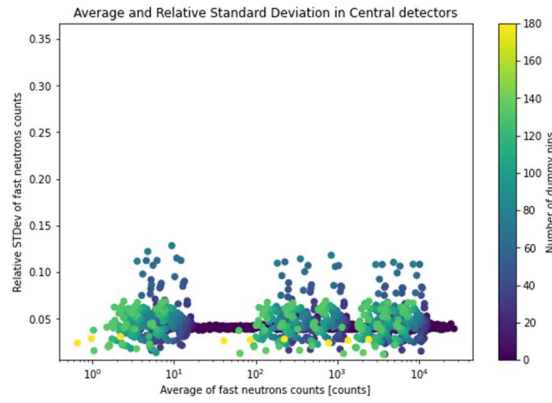


Figure A.38. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Fast neutrons counts - Number of dummy pins

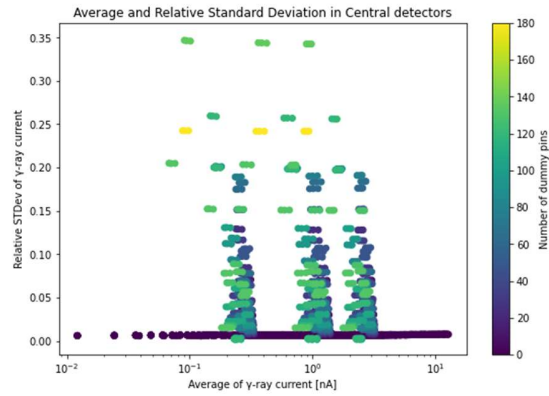


Figure A.39. Database visualization: Complete and defect scenarios - Average and relative standard deviation in central detectors - Photons current - Number of dummy pins

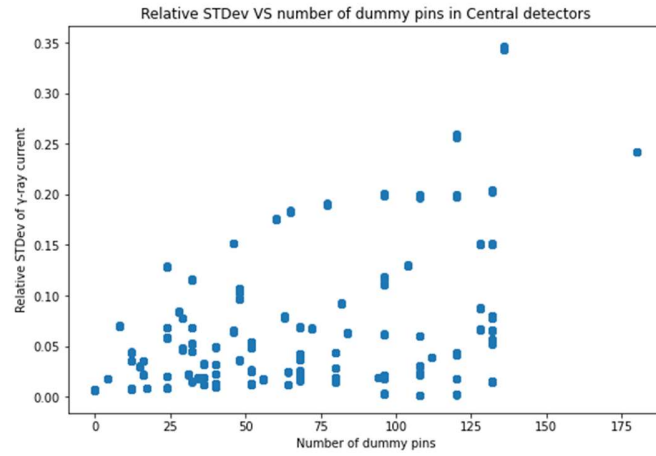


Figure A.40. Database visualization: Complete and defect scenarios - Relative standard deviation of photons current in central detectors - Number of dummy pins

Average in external detectors

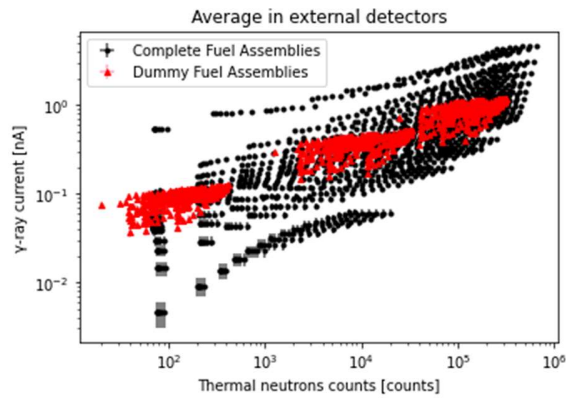


Figure A.41. Database visualization: Complete and defect scenarios – Average in external detectors and relative uncertainties – Thermal neutrons counts and photons current

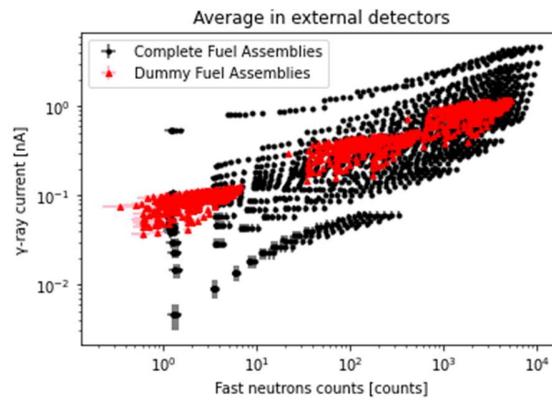


Figure A.42. Database visualization: Complete and defect scenarios - Average in external detectors and relative uncertainties – Fast neutrons counts and photons current

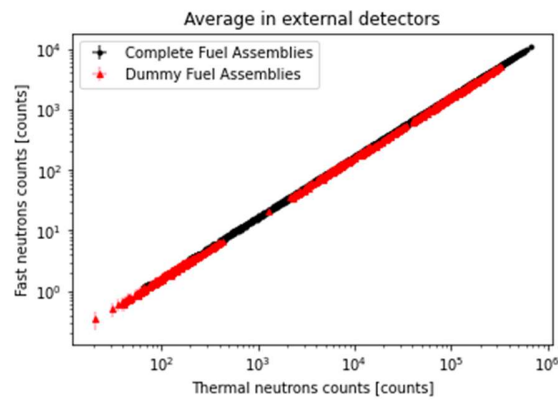


Figure A.43. Database visualization: Complete and defect scenarios - Average in external detectors and relative uncertainties – Thermal and fast neutrons counts

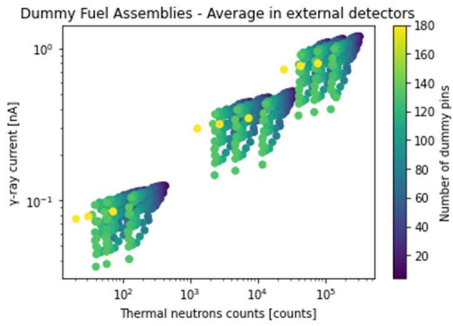


Figure A.44. Database visualization: Defect scenarios - Average in external detectors - Thermal neutrons counts and photons current - Number of dummy pins

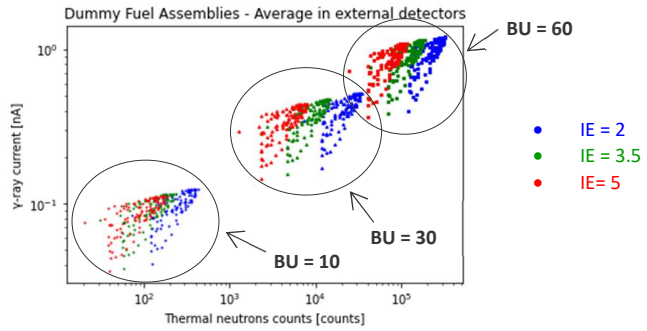


Figure A.45. Database visualization: Defect scenarios – Average in external detectors - Thermal neutrons counts and photons current – Initial Enrichment and BurnUp

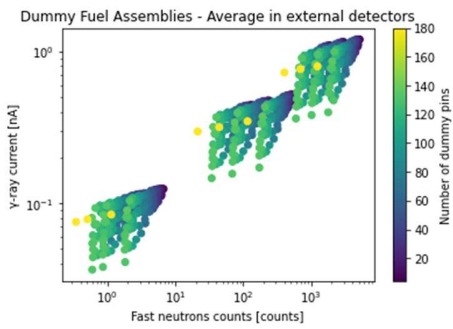


Figure A.46. Database visualization: Defect scenarios – Average in external detectors - Fast neutrons counts and photons current - Number of dummy pins

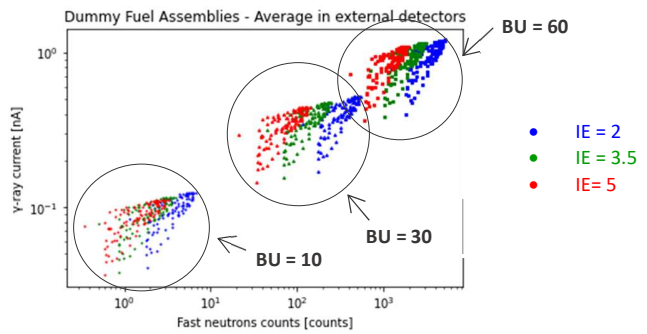


Figure A.47. Database visualization: Defect scenarios – Average in external detectors - Fast neutrons counts and photons current – Initial Enrichment and BurnUp

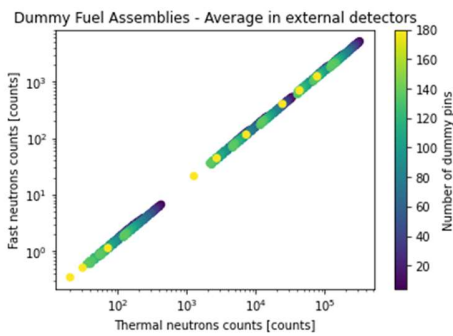


Figure A.48. Database visualization: Defect scenarios – Average in external detectors - Thermal and fast neutrons counts - Number of dummy pins

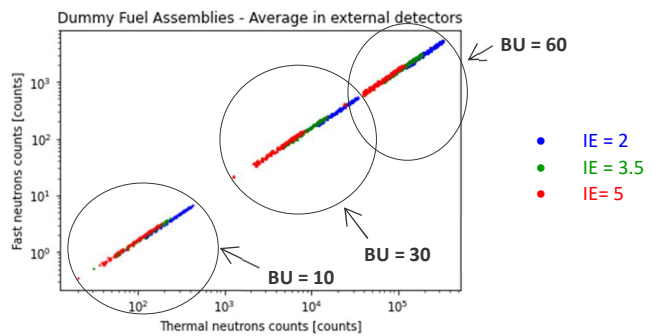


Figure A.49. Database visualization: Defect scenarios – Average in external detectors - Thermal and fast neutrons counts – Initial Enrichment and BurnUp

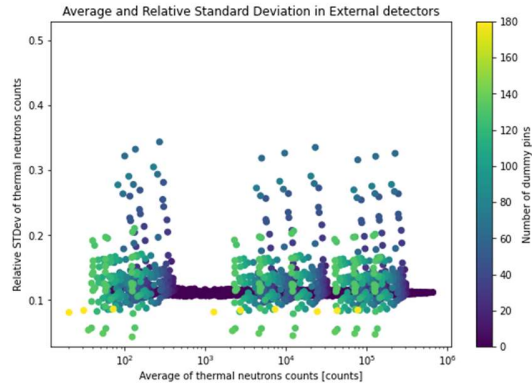


Figure A.50. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Thermal neutrons counts - Number of dummy pins

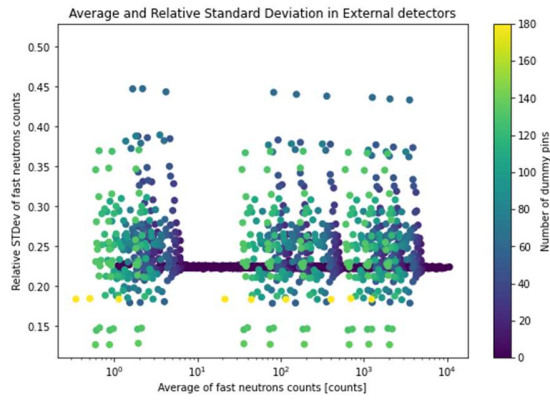


Figure A.51. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Fast neutrons counts - Number of dummy pins

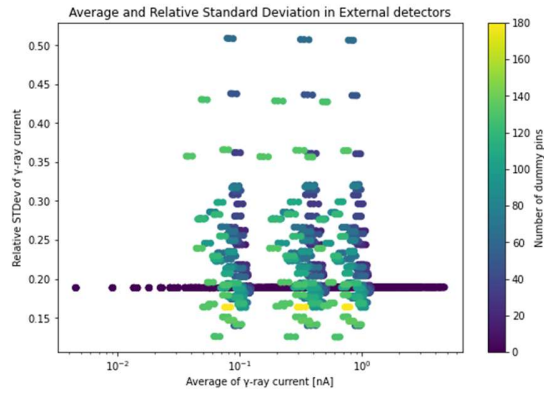


Figure A.52. Database visualization: Complete and defect scenarios - Average and relative standard deviation in external detectors - Photons current - Number of dummy pins

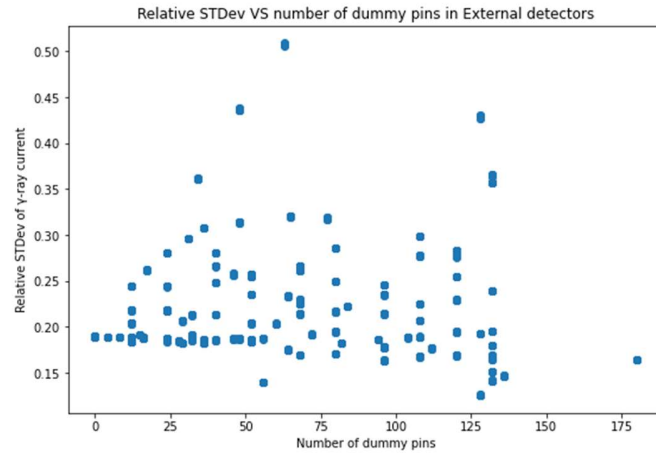


Figure A.53. Database visualization: Complete and defect scenarios - Relative standard deviation of photons current in external detectors - Number of dummy pins

APPENDIX B: RESULTS

In this section are reported the additional graphs related to the results of the machine learning models developed in this work, presented in [Chapter 4](#). Specifically, the following graphs are presenting the delta extracted by each model, i.e., the difference between the predicted and the exact number of dummy pins, for each combination of input features investigated. The plots produced from the outcomes of the k-NN models are reported in the first part, while the second is dedicated to the outputs of the NN models.

k-NN regression models

Number of dummy pins

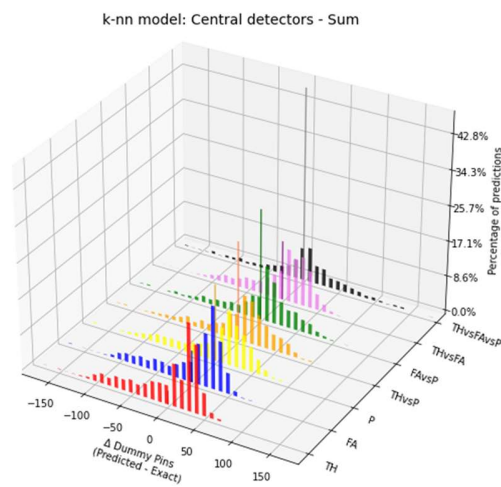


Figure B.1. k-NN model Δ - Sum within central detectors for all combinations of kinds of detectors responses

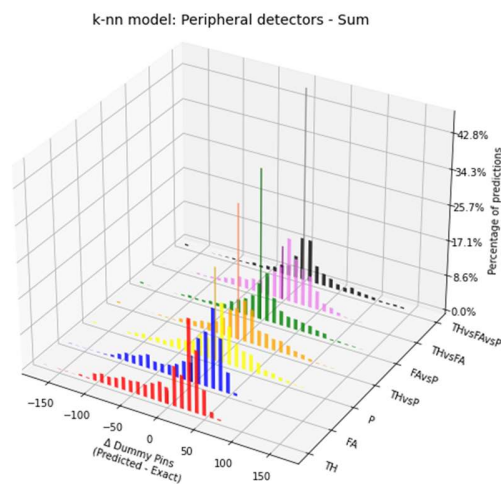


Figure B.2. k-NN model Δ - Sum within peripheral detectors for all combinations of kinds of detectors responses

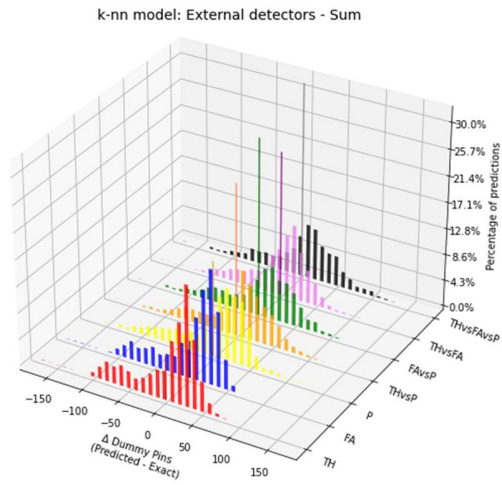


Figure B.3. k-NN model Δ - Sum within external detectors for all combinations of kinds of detectors responses

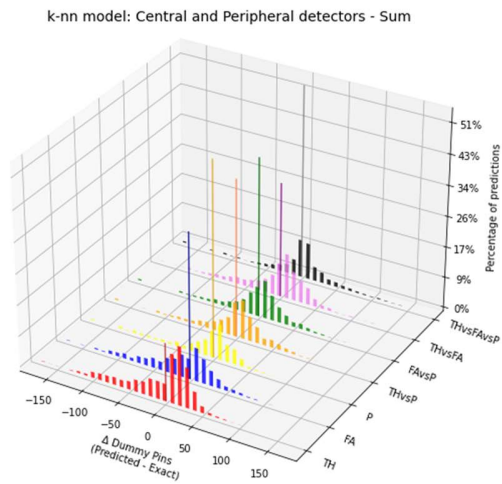


Figure B.4. k-NN model Δ - Sum within central and peripheral detectors for all combinations of kinds of detectors responses

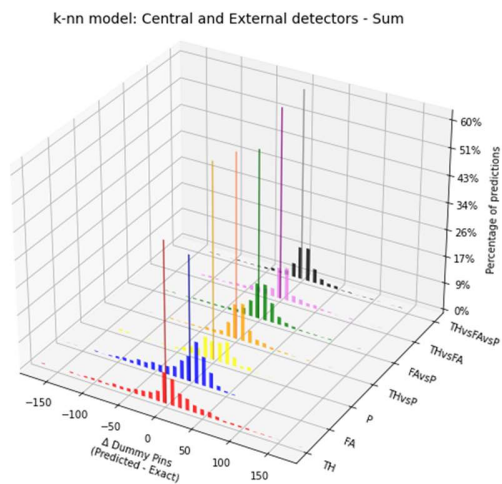


Figure B.5. k-NN model Δ - Sum within central and external detectors for all combinations of kinds of detectors responses

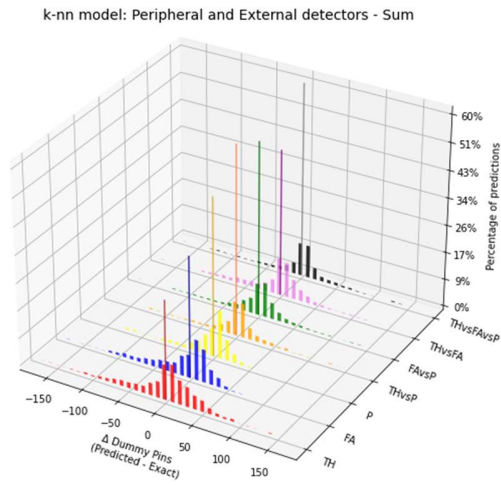


Figure B.654. k-NN model Δ - Sum within peripheral and external detectors for all combinations of kinds of detectors responses

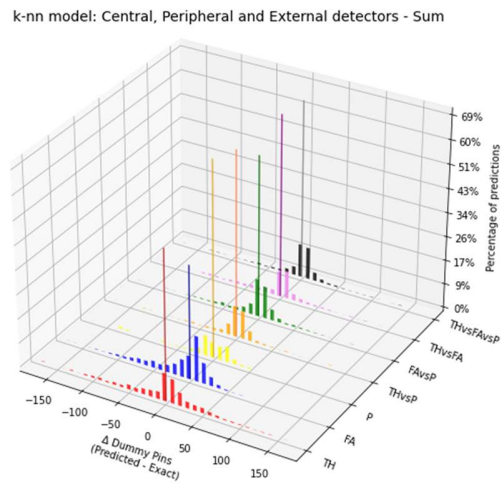


Figure B.755. k-NN model Δ - Sum within central, peripheral and external detectors for all combinations of kinds of detectors responses

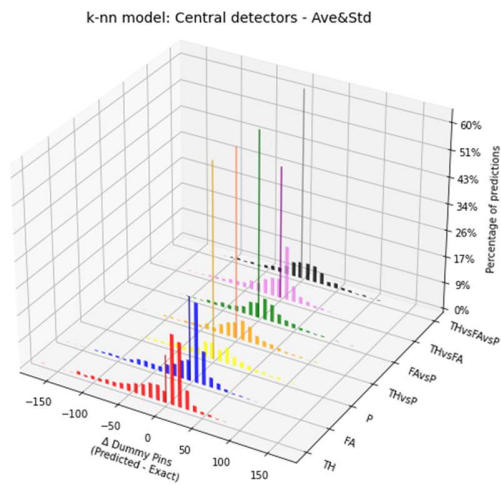


Figure B.8. k-NN model Δ - Ave&Std within central detectors for all combinations of kind of detectors responses

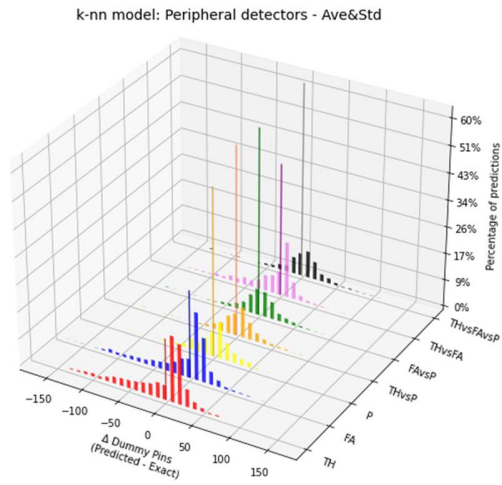


Figure B.9. *k-NN model Δ - Ave&Std within peripheral detectors for all combinations of kinds of detectors responses*

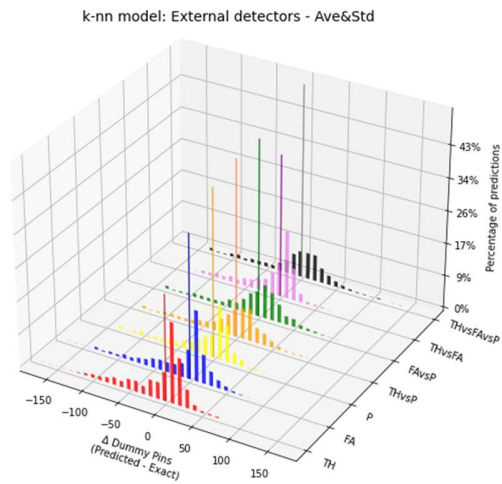


Figure B.10. *k-NN model Δ - Ave&Std within external detectors for all combinations of kinds of detectors responses*

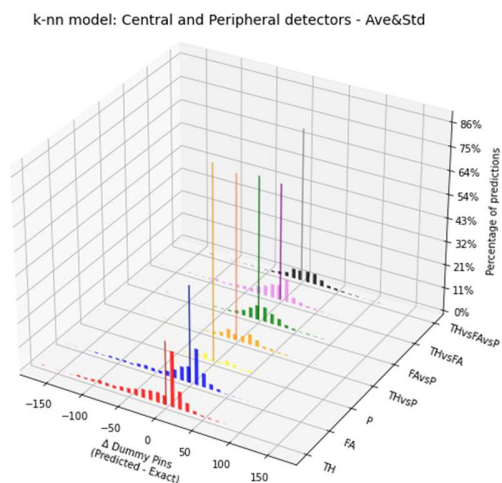


Figure B.11. *k-NN model Δ - Ave&Std within central and peripheral detectors for all combinations of kinds of detectors responses*

k-nn model: Central and External detectors - Ave&Std

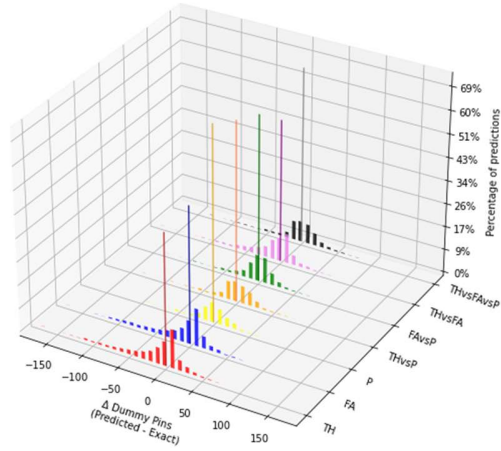


Figure B.12. k-NN model Δ - Ave&Std within central and external peripheral detectors for all combinations of kinds of detectors responses

k-nn model: Peripheral and External detectors - Ave&Std

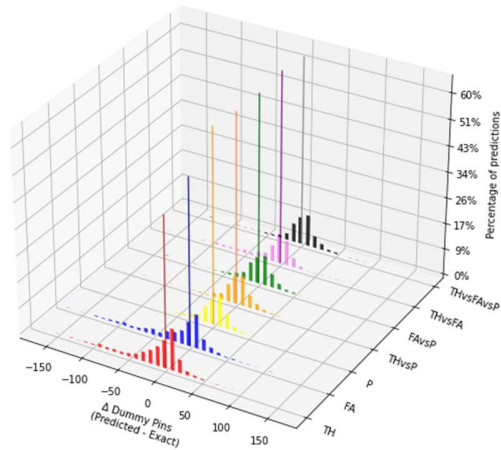


Figure B.13. k-NN model Δ - Ave&Std within peripheral and external detectors for all combinations of kinds of detectors responses

k-nn model: Central detectors

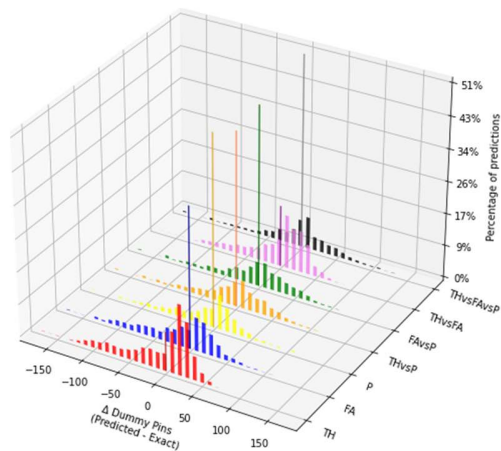


Figure B.14. k-NN model Δ – Single central detectors values for all combinations of kinds of detectors responses

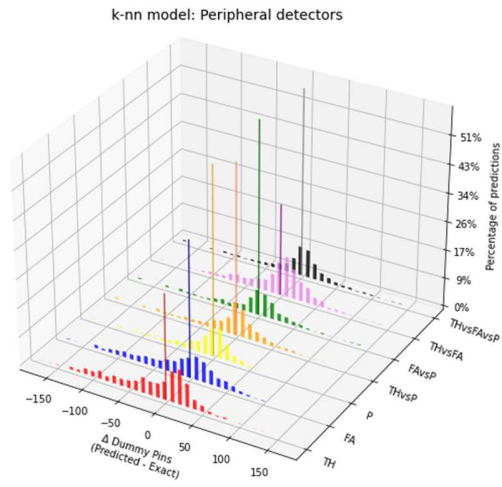


Figure B.15. *k-NN model Δ – Single peripheral detectors values for all combinations of kinds of detectors responses*

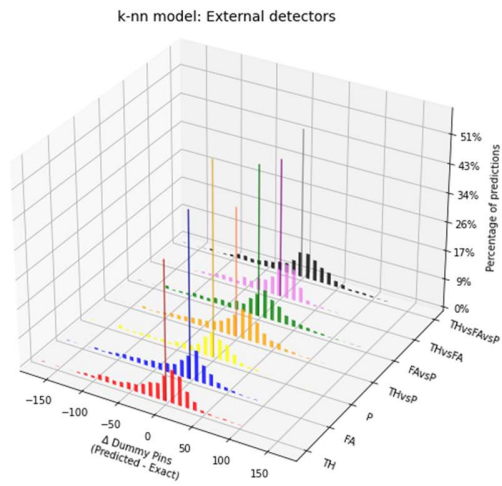


Figure B.16. *k-NN model Δ – Single external detectors values for all combinations of kinds of detectors responses*

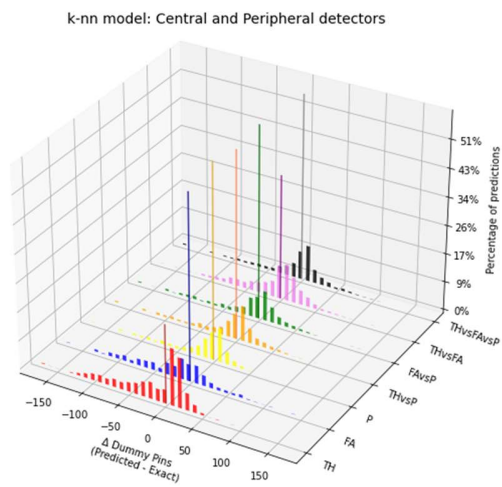


Figure B.17. *k-NN model Δ – Single central and peripheral detectors values for all combinations of kinds of detectors responses*

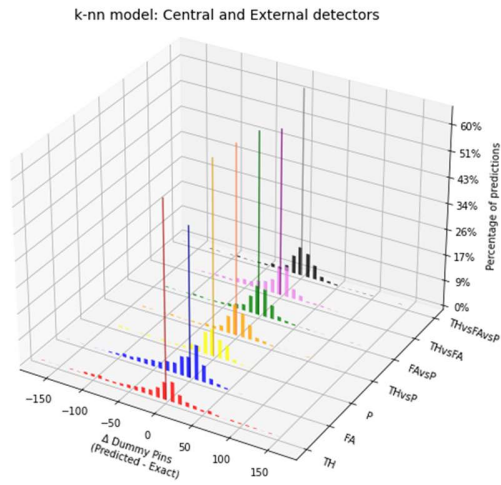


Figure B.18. k-NN model Δ – Single central and external detectors values for all combinations of kinds of detectors responses

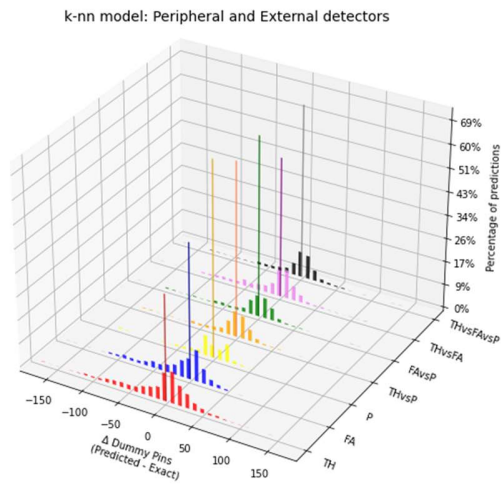


Figure B.19. k-NN model Δ – Single peripheral and external detectors values for all combinations of kinds of detectors responses

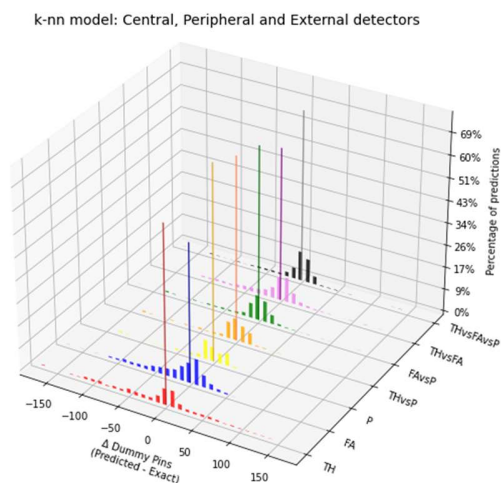


Figure B.20. k-NN model Δ – Single central, peripheral and external detectors values for all combinations of kinds of detectors responses

NN regression models

Number of dummy pins

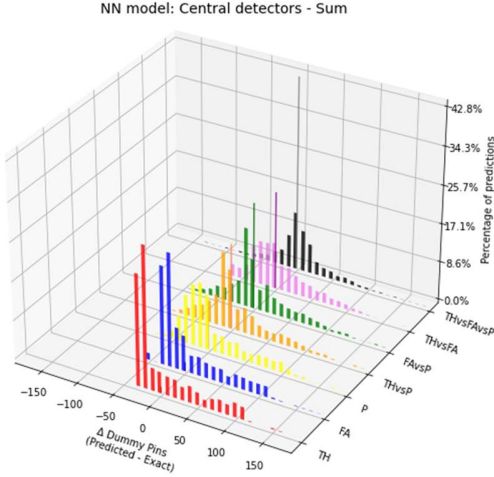


Figure B.21. NN model Δ - Sum within central detectors for all combinations of kinds of detectors responses

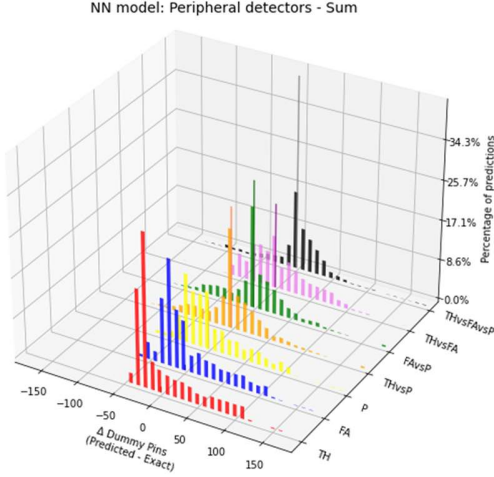


Figure B.22. NN model Δ - Sum within peripheral detectors for all combinations of kinds of detectors responses

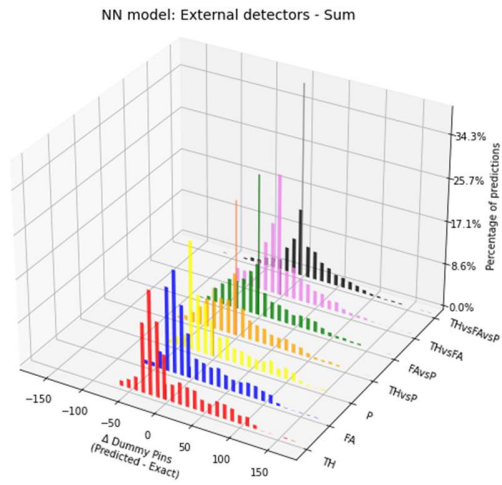


Figure B.23. NN model Δ - Sum within external detectors for all combinations of kinds of detectors responses

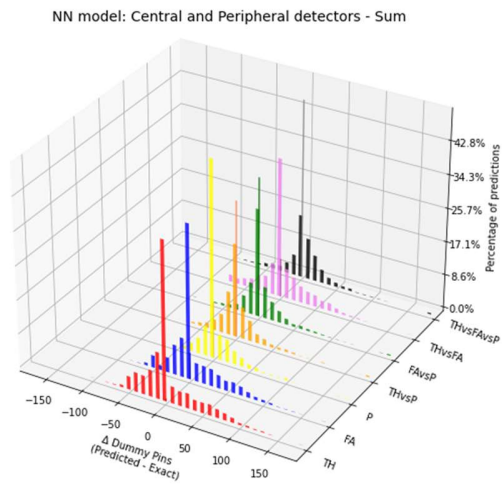


Figure B.24. NN model Δ - Sum within central and peripheral detectors for all combinations of kinds of detectors responses

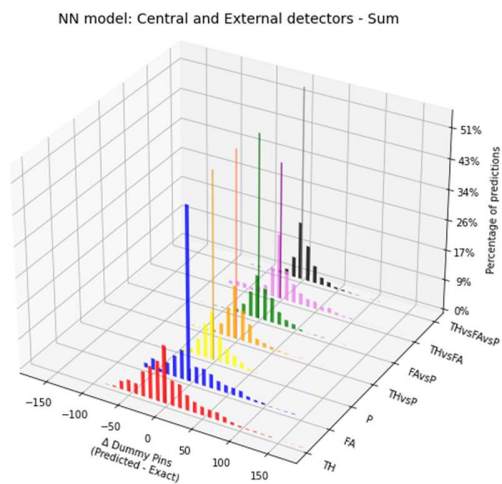


Figure B.25. NN model Δ - Sum within central and external detectors for all combinations of kinds of detectors responses

NN model: Peripheral and External detectors - Sum

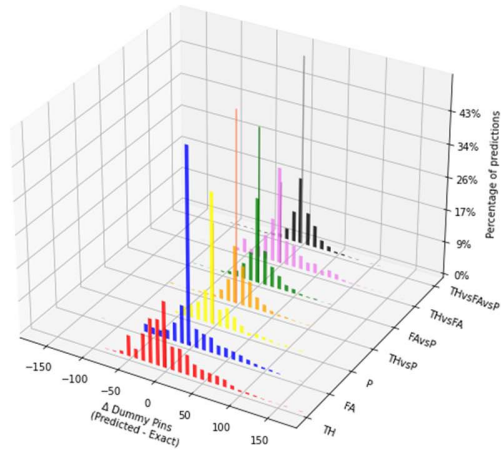


Figure B.26. NN model Δ - Sum within peripheral and external detectors for all combinations of kinds of detectors responses

NN model: Central, Peripheral and External detectors - Sum

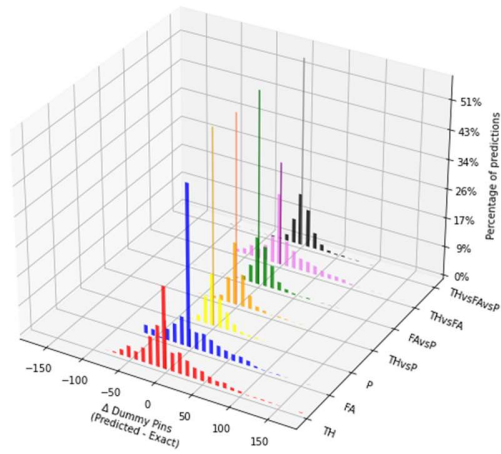


Figure B.27. NN model Δ - Sum within central, peripheral and external detectors for all combinations of kinds of detectors responses

NN model: Central detectors - Ave&Std

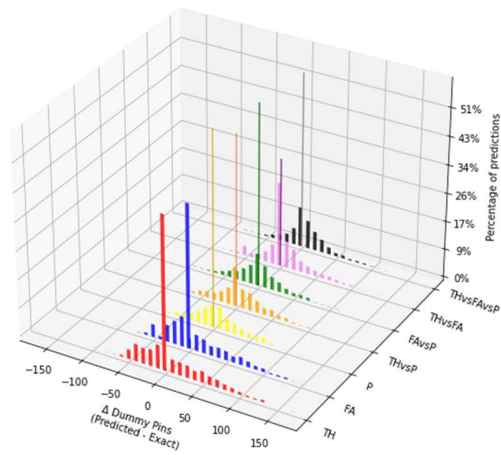


Figure B.28. NN model Δ - Ave&Std within central detectors for all combinations of kinds of detectors responses

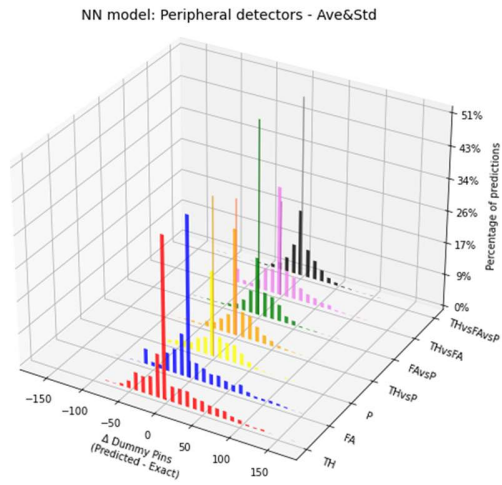


Figure B.29. NN model Δ - Ave&Std within peripheral detectors for all combinations of kinds of detectors responses

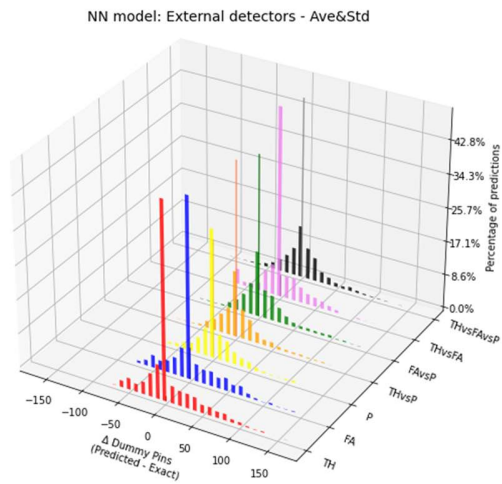


Figure B.30. NN model Δ - Ave&Std within external detectors for all combinations of kinds of detectors responses

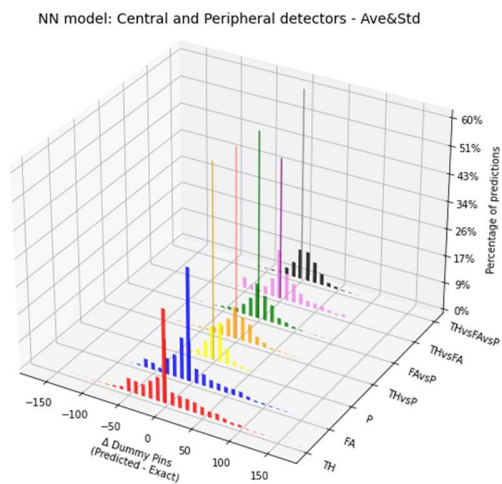


Figure B.31. NN model Δ - Ave&Std within central and peripheral detectors for all combinations of kinds of detectors responses

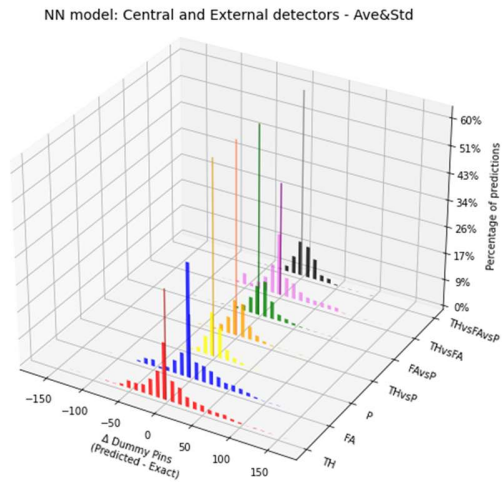


Figure B.32. NN model Δ - Ave&Std within central and external detectors for all combinations of kinds of detectors responses

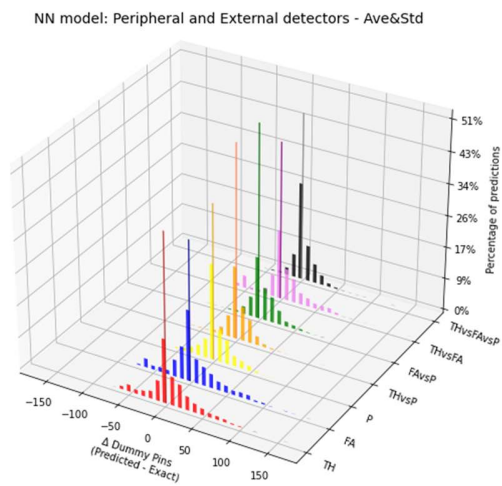


Figure B.33. NN model Δ - Ave&Std within peripheral and external detectors for all combinations of kinds of detectors responses

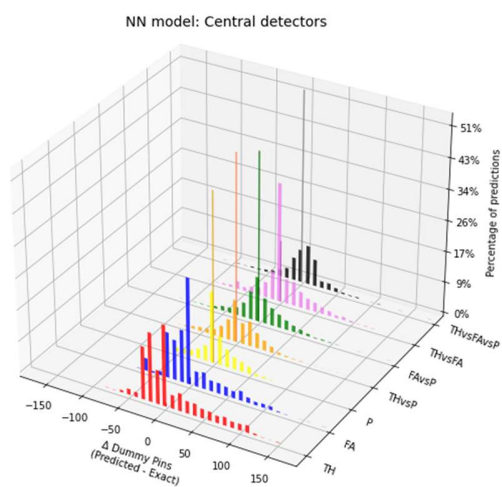


Figure B.34. NN model Δ - Single central detectors values for all combinations of kinds of detectors responses

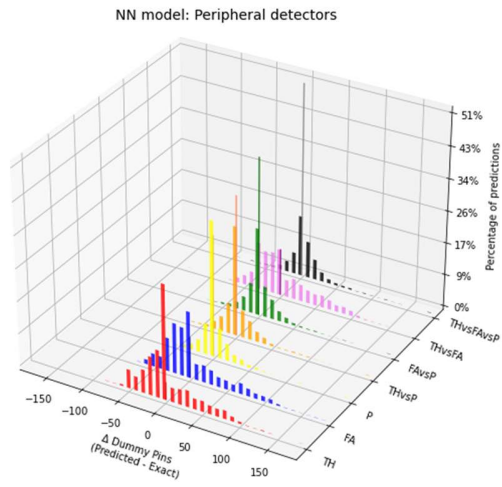


Figure B.35. NN model Δ – Single peripheral detectors values for all combinations of kinds of detectors responses

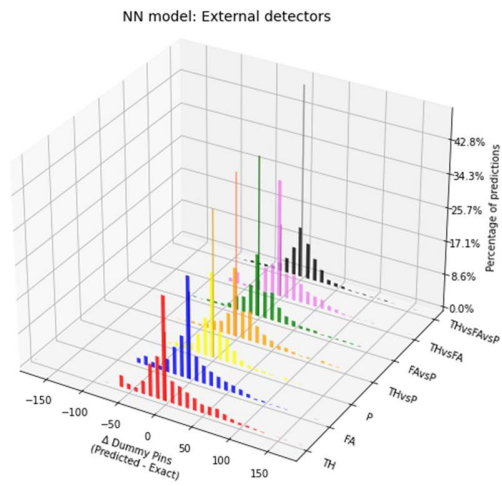


Figure B.36. NN model Δ – Single external detectors values for all combinations of kinds of detectors responses

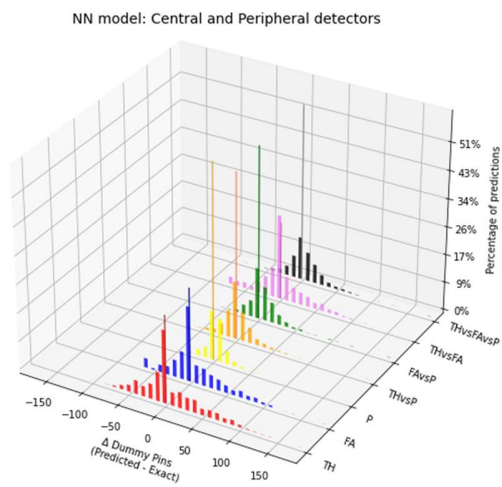


Figure B.37. NN model Δ - Single central and peripheral detectors values for all combinations of kinds of detectors responses

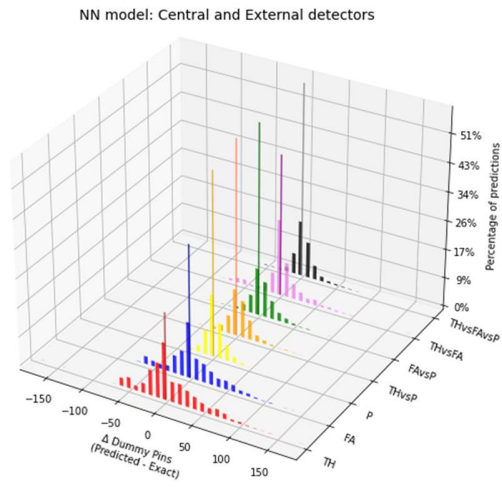


Figure B.38. NN model Δ - Single central and external detectors values for all combinations of kinds of detectors responses

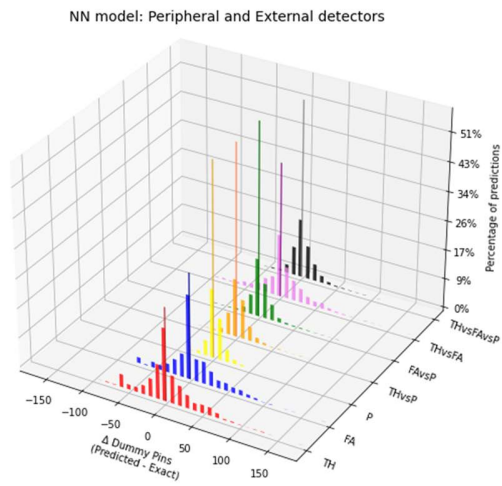


Figure B.39. NN model Δ - Single peripheral and external detectors values for all combinations of kinds of detectors responses