

Politecnico di Torino

Dipartimento di Ingegneria Energetica
Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare
A.A. 2023/2024
Sessione di Laurea: marzo 2024

Tesi di Laurea Magistrale

Improving validation of Instadose dosimeters using machine learning algorithms



**Politecnico
di Torino**

sck cen

Relatore: Dr. Raffaella Testoni

Mentor: Roberto Federica

Co-Mentor: Dr. Vanhavere Filip

Candidato: Eugeni Luca

Acknowledgments

First, a huge thank you goes to my mentor Federica Roberto. Her kindness and empathy led me putting all the effort and passion in this thesis, even in moments marked by big changes for my personal and professional career. I would also like to thank my co-mentor Filip Vanhavere for the technical comments and for believing in this project, as well as my promoter Raffaella Testoni for all the support concerning the revision of this thesis and the bureaucracy process needed.

Second, I would like to thank the many people that supported me during this master thesis and mainly in these last months. To my “Belgian” friends: I will miss a lot the moments passed together, I have found new shades of my character that i was not expecting and for that I am grateful. Hopefully, I have managed to spread with you some of the joy I have been lucky enough to experience in these months!

Switching briefly to Italian... un grazie va ai miei compagni e amici che ho avuto la fortuna di incontrare a Torino. Siete riusciti ad allietare degli anni importantissimi contribuendo a momenti indimenticabili. Nonostante il futuro sembri incerto e decisamente poco delineato ho la fortuna di poter sempre contare su di voi.

Un altro grandissimo grazie è indirizzato alla mia famiglia, per avermi sempre sostenuto nelle mie scelte e per essere sempre pronta a darmi nuovi stimoli per migliorare e trovare il mio posto nel mondo.

Coming back to English, no names or other words are needed to express how grateful I am for all of you. You made me a better person, love you all!

This thesis has been co-funded by the European Union
by means of ENEN2plus mobility grants.



**Co-funded by
the European Union**

Abstract

Dosimetry represents a key aspect in the radioprotection field for nuclear related activities: people working with ionising radiation need to monitor their exposure to be able to optimise their doses and to keep them below the dose limits. Such dosimeters are provided by approved dosimetry services, like SCK CEN. These dosimetry services need to prove that the doses they measure are correct. In this process, the validation of the personal dosimeter results is of major importance. The validation process ensures a reliable measurement of the personal dose equivalent and guarantees the correct operation of the dosimeter. In this project the focus is on Instadose dosimeters, which is a novel type of hybrid personal Dosimeters.

SCK nuclear research centre provides currently ~ 2000 Instadose dosimeters to different costumers, such as hospitals and companies. These Instadose dosimeters transfer the measured doses automatically to the dosimetry services once per week. Once per month a validation process is performed manually for each dosimeter by means of Microsoft Excel sheets. The latter contain information on the status of the device, details on the last measurements, and graphs with dose trends during the last months. In particular, these graphs permit an easy control of measurements that deviate from expected values or that present anomalies. Thanks to this modality, it is straightforward to classify these dosimeters in 10 different classes, based on the actions that have to be adopted for the dosimeter. This process is remarkably time consuming and the need to fasten it up is required since an increase of the number of these dosimeters is foreseen in the incoming years.

In order to allow a faster validation, Machine Learning algorithms have been developed and tested in this project. The main idea is to identify all dosimeters results that are in line with the expected performances, for which no actions need to be taken. Although the remaining results will still require manual evaluation, the automatic validation would reduce the workload significantly. The implementation process starts from a careful cleaning and rearranging of the data containing all the measurements of the previous year. The latter has been later used to create a unique database for training and testing of the models. Information collected inside this database are based on the validation process performed manually. Four different supervised classification models have been implemented: Decision Trees, Random Forest, K-Nearest Neighbours and Neural Networks. This first implementation was based on a binary classification approach to distinguish well-performing dosimeters from faulty ones.

For each model, many combinations of hyperparameters have been implemented for the tuning of the algorithms. This approach has been followed in order to find a subset of combinations that can be easily applied on new incoming data, providing confidence intervals for the dosimeter's classification. Although Decision Tree model for classification is rather straightforward, with respect to other developed models, it provides slightly better results concerning precision for binary classification. It is difficult to declare an overall best model based on the results obtained: small changes can be observed and no explicit correlations are always present. Therefore, a set of hyperparameter combinations has been identified for each model, in order to be implemented on new unseen data for monthly validations.

Contents

Acknowledgments	III
Abstract	V
Contents	VII
List of acronyms	IX
List of figures	XI
List of tables	XV
1. Introduction	1
1.1. Radiation protection: standards and regulations	1
1.2. Dosimetry	3
1.2.1. Dosimetric and protection quantities	3
1.2.2. Operational quantities	7
1.2.3. Dosimeters	8
1.3. Machine Learning	13
1.3.1. Unsupervised learning	13
1.3.2. Reinforcement learning	14
1.3.3. Supervised learning	14
1.3.4. Deep Learning	16
2. Validation process and database configuration	17
2.1. Instadose dosimeters	17
2.2. SCK CEN validation process	18
2.2.1. SQL database	19
2.2.2. Microsoft Excel sheets	21
2.3. Configuration of final database for ML implementation	23
2.3.1. Data rearranging and grouping	24
2.3.2. Dimensionality reduction	26
2.3.3. Pre-processing of the database	33
3. Machine Learning algorithms	35
3.1. K-Nearest Neighbours	36
3.1.1. Hyperparameter definition	36
3.2. Decision Trees	38
3.2.1. Hyperparameter definition	40
3.3. Random forest	41
3.3.1. Hyperparameters definition	42
3.4. Neural Networks	42
3.4.1. Hyperparameters definition	44

4. Results	49
4.1. <i>Metric description</i>	49
4.2. <i>Cross validation</i>	50
4.3. <i>Binary classification</i>	51
4.4. <i>K-Nearest Neighbour implementation</i>	52
4.4.1. Results of K-Nearest Neighbours models	55
4.5. <i>Decision Tree implementation</i>	56
4.5.1. Results of Decision Tree models	62
4.6. <i>Random Forest implementation</i>	62
4.6.1. Results of Random Forest models	67
4.7. <i>Neural Networks implementation</i>	68
4.7.1. Results of Neural Networks models	74
5. Discussion and conclusions	77
Bibliography	81

List of acronyms

<i>AI</i>	Artificial Intelligence
<i>ALARA</i>	As Low As Reasonably Achievable
<i>APD</i>	Active Personal Dosimeters
<i>BLE</i>	Bluetooth Low Energy
<i>BLP</i>	Begin Linking Period
<i>BP</i>	Back Propagation
<i>CV</i>	Cross Validation
<i>DIS</i>	Direct-Ion-Storage
<i>DL</i>	Deep Learning
<i>DT</i>	Decision Tree
<i>ED</i>	Exposure Date
<i>EEPROM</i>	Electrically Erasable Programmable Read-Only Memory
<i>ELP</i>	End Linking Period
<i>EPD</i>	Electronic Personal Dosimeters
<i>EURATOM</i>	EUROpean ATOMIC energy community
<i>FA</i>	False <i>A</i>
<i>FL</i>	False <i>L</i>
<i>FN</i>	False Negative
<i>FP</i>	False Positive
<i>IAEA</i>	International Atomic Energy Agency
<i>ICRP</i>	International Commission on Radiological Protection
<i>ICRU</i>	International Commission on Radiation Units and measurements
<i>ID</i>	IDentification
<i>IEC</i>	International Electrotechnical Commission
<i>ISO</i>	International Organization for Standardization
<i>KD</i>	K-Dimensional
<i>KERMA</i>	Kinetic Energy Released to MATter
<i>KNN</i>	K-Nearest Neighbours
<i>LET</i>	Linear Energy Transfer
<i>ML</i>	Machine Learning
<i>MLP</i>	Multi-Layer Perceptron
<i>NaN</i>	Not a Number
<i>NN</i>	Neural Network
<i>OSL</i>	Optically Stimulated Luminescence
<i>PCA</i>	Principal Component Analysis
<i>RBE</i>	Relative Biological Effectiveness
<i>ReLU</i>	Rectified Linear Unit
<i>RF</i>	Random Forest
<i>RPL</i>	Radio Photo Luminescence
<i>SGD</i>	Stochastic Gradient Descent

<i>SI</i>	International System of unit
<i>SQL</i>	Structured Query Language
<i>TA</i>	True <i>A</i>
<i>TL</i>	True <i>L</i>
<i>TLD</i>	Thermo-Luminescence Dosimeters
<i>TN</i>	True Negative
<i>TP</i>	True Positive

List of figures

Figure 1.1: 5 th congress of International Society of Radiology [4]	2
Figure 1.2: weighting factor function for neutron radiation [2]	6
Figure 1.3: dosimeter patent by Arthur Mutscheller n° US-2036072-A [18].....	8
Figure 1.4: whole-body and extremities dosimeters provided by and used at SCK CEN [Photo courtesy of SCK CEN].....	9
Figure 1.5: film-badge holder and processed film [23]	10
Figure 1.6: thermoluminescent glow curve for <i>LiF: Mg, Ti</i> [26]	11
Figure 1.7: (a) EEPROM memory cell (b) DIS memory cell [29]	12
Figure 1.8: scheme for description of machine learning subgroups [32]	13
Figure 1.9: general scheme for distinction of supervised and unsupervised learning [34].....	14
Figure 1.10: example of decision tree [36].....	15
Figure 1.11: training and testing error related to model complexity [37].....	16
Figure 1.12: an illustration of a deep learning neural network [38]	16
Figure 2.1: differences between Instadose + and Instadose 2 dosimeters [39].....	17
Figure 2.2: table with hard reads in Microsoft Excel sheet	21
Figure 2.3: main information regarding dosimeter in Microsoft Excel sheet.....	21
Figure 2.4: real dose graph with no filters for dosimeter validation.....	22
Figure 2.5: dose graph filtered from outliers and background contribution.....	23
Figure 2.6: dose graph filtered with jumps higher than $150 \mu Sv$	23
Figure 2.7: correlation matrix for grouped database evaluated with Sweetviz library	28
Figure 2.8: Spearman's correlation matrix for grouped database.....	29
Figure 2.9: correlation matrix for grouped database evaluated by Sweetviz library after feature selection ...	31
Figure 2.10: Spearman's correlation matrix for grouped database after feature selection	32
Figure 2.11: histogram with classes distribution in grouped database	33
Figure 2.12: histogram with classes distribution in grouped database after class balancing process.....	34
Figure 3.1: comparison of KNN decision boundaries with $K = 1$ and $K = 100$ [32].....	37
Figure 3.2: example of a basic decision tree [48].....	39
Figure 3.3: decision tree implemented in early stages during this thesis project	39
Figure 3.4: example of a basic random forest classifier [50]	41
Figure 3.5: schematization of a single layer MLP [51]	43
Figure 3.6: schematization of a multi-layer MLP [51].....	44
Figure 3.7: activation functions shape [57]	46
Figure 4.1: schematic confusion matrix for binary classification [60].....	49
Figure 4.2: scatter plot of KNN models with accuracy, precision and K hyperparameter	53
Figure 4.3: scatter plot of KNN models with accuracy, precision and K hyperparameter	53
Figure 4.4: box plot of KNN models with precision and weight hyperparameter.....	54

Figure 4.5: box plot of KNN models with precision and algorithm hyperparameter	55
Figure 4.6: scatter plot of KNN models with accuracy, precision and metric hyperparameter	55
Figure 4.7: scatter plot of DT models with accuracy, precision and tree depth hyperparameter	57
Figure 4.8: line plot with highest precision for DT models related to tree depth hyperparameter	58
Figure 4.9: scatter plot of DT models with accuracy, precision and maximum leaf nodes hyperparameter...	58
Figure 4.10: scatter plot of DT models with accuracy, precision and criterion hyperparameter.....	59
Figure 4.11: box plot of DT models with precision and criterion hyperparameter	59
Figure 4.12: scatter plot of DT models with accuracy, precision and minimum samples per leaf hyperparameter	60
Figure 4.13: line plot with highest precisions for DT models related to minimum samples per leaf hyperparameter	60
Figure 4.14: scatter plot of DT models with accuracy, precision and max features hyperparameter.....	61
Figure 4.15: box plot of DT models with precision and max features hyperparameter	61
Figure 4.16: scatter plot of RF models with accuracy, precision and number of estimators hyperparameter .	63
Figure 4.17: violin plot of RF models with precision and number of estimators hyperparameter	64
Figure 4.18: line plot with mean precisions for RF models related to number of estimators hyperparameter	64
Figure 4.19: scatter plot of RF models with accuracy, precision and tree depth hyperparameter.....	65
Figure 4.20: box plot of RF models with precision and tree depth hyperparameter	65
Figure 4.21: box plot of RF models with precision and bootstrap hyperparameter	66
Figure 4.22: scatter plot of RF models with accuracy, precision and bootstrap hyperparameter	66
Figure 4.23: scatter plot of RF models with accuracy, precision and warm start hyperparameter.....	67
Figure 4.24: violin plot of RF models with precision and warm start hyperparameter	67
Figure 4.25: scatter plot of NN models with accuracy, precision and number of layers and nodes hyperparameter	69
Figure 4.26: box plot of NN models with precision and number of layers and nodes hyperparameter	70
Figure 4.27: scatter plot of NN models with accuracy, precision and activation function hyperparameter....	70
Figure 4.28: box plot of NN models with precision and activation function hyperparameter	71
Figure 4.29: scatter plot of NN models with accuracy, precision and solver hyperparameter	71
Figure 4.30: box plot of NN models with precision and solver hyperparameter.....	72
Figure 4.31: scatter plot of NN models with accuracy, precision and learning rate hyperparameter	72
Figure 4.32: box plot of NN models with precision and learning rate hyperparameter	73
Figure 4.33: scatter plot of NN models with accuracy, precision and batch size hyperparameter	73
Figure 4.34: box plot of NN models with precision and batch size hyperparameter	74
Figure 5.1: precision comparison between all the 4 ML models based on selected subset of hyperparameter combinations.....	78
Figure 5.2: accuracy comparison between all the 4 ML models based on selected subset of hyperparameter combinations.....	78
Figure 5.3: weighted CV precision comparison between all the 4 ML models based on selected subset of hyperparameter combinations	79

Figure 5.4: weighted CV accuracy comparison between all the 4 ML models based on selected subset of hyperparameter combinations 79

List of tables

Table 1.1: radiation weighting factors for equivalent dose evaluation [2]	5
Table 1.2: tissue weighting factor for effective dose evaluation values [2]	6
Table 2.1: information retrieved by the SQL database for validation process	20
Table 3.1: main hyperparameters values for KNN classifier.....	37
Table 3.2: main hyperparameters values for DT classifiers	40
Table 3.3: main hyperparameters values for RF classifiers.....	42
Table 3.4: main hyperparameters values for NN classifiers	45
Table 4.1: folds percentage of the whole database for cross validation	51
Table 4.2: percentages of <i>A</i> and <i>L</i> classes in the datasets.....	52
Table 4.3: results summary concerning number of combination, highest accuracy and precision	52
Table 4.4: best KNN models from all hyperparameter combinations	52
Table 4.5: hyperparameter subset for KNN classifiers.....	56
Table 4.6: 5 best KNN models from selected subset of hyperparameter combinations	56
Table 4.7: best DT models from all hyperparameter combinations	57
Table 4.8: hyperparameter subset for DT classifiers	62
Table 4.9: 5 best DT models from selected subset of hyperparameter combinations	62
Table 4.10: best RF models from all hyperparameter combinations	63
Table 4.11: hyperparameter subset for RF classifiers	68
Table 4.12: 5 best RF models from selected subset of hyperparameter combinations.....	68
Table 4.13: best NN models from all hyperparameter combinations	69
Table 4.14: hyperparameter subset for NN classifiers.....	74
Table 4.15: 5 best NN models from selected subset of hyperparameter combinations	75
Table 5.1: highest precisions and relative accuracies for all 4 models after CV process	80

1. Introduction

Since the discovery of radioactivity of uranium in 1896 by Henri Becquerel [1] and from further exploration of this phenomena thanks to Marie and Pierre Curie, ionizing radiations have been adopted for multiple purposes, spacing from medicine to energy production. Since the beginning of the 20th century, it has been clear that exposure to radiation posed potential health risks. For this reason, many studies have been conducted in order to categorize, standardize and report quantities and phenomena that belongs to this branch of nuclear physics. In 1928 the International Commission on Radiological Protection (ICRP) published the first document that provides recommendations and guidance regarding radiological protection, titled "Recommendations of the International Commission on Radiological Protection". Updated documents have been published by ICRP from that moment forward, combining latest findings and new information, latest document dates back to 2007 [2]. In June of 1962, Basic Safety Standards have been approved for the first time by International Atomic Energy Agency (IAEA) Board of Governors. These standards guarantee fundamental principles, requirements and recommendations to ensure nuclear safety. Latest safety standards for occupational radiation protection have been published in 2018 [3]. All the states that are currently members of the IAEA take these principles as global reference for the aspects that concern nuclear safety.

The need to understand and measure the effects of radiation exposure on living organisms led to the establishment of the Dosimetry field, defined as the measurement of radiation doses. The latter is fundamental in order to ensure the safety of individuals working with radioactive materials, as well as with diagnostic procedures. Over time, advancements in technology led to the parallel improvement in dosimeters manufacture. These advancements have enhanced precision and accuracy of radiation dose measurements, enabling better monitoring and regulation of exposure levels. This is the case of new Electronic Personal Dosimeters (EPDs), and one important example is represented by the new Direct Ion Storage dosimeters, which will be further analysed in detail since it has been extensively used as reference for this thesis work.

The introduction of electronic devices in the Dosimetry field launches new possibilities regarding the treatment of the data resulting from the exposure measurements. Indeed, data regarding the equivalent dose can be used in combination with Machine Learning (ML) algorithms. Nowadays, Artificial Intelligence is spreading above a large portion of research fields, in order to ease computations and tasks. In this thesis, ML algorithms have been implemented in order to simplify the validation process of innovative electronic dosimeters.

1.1. Radiation protection: standards and regulations

The term radiation protection indicates the practices, principles and measures that are taken in order to ensure protection of individuals, communities and environment from effect of ionizing and non-ionizing radiations. In first instance, the use of ionizing radiations was related to clinical purposes: John Hall-Edwards was the first to use X-rays for medical purposes creating a new profession called radiologist, specialized in medical imaging. However, first patients and radiologists suffered from hair loss, skin burns and other physical effects, later justified as related to ionizing radiation exposition. In the early stage of X-ray imaging, just superficial layers of the human body were assumed to present biological side-effects. Once the radiological community realised that a considerable amount of radiation could be absorbed by the deeper tissues, even without showing superficial damages, deep effects were contemplated, finding a correlation between ionizing exposition and induction of carcinoma.

It became of primary importance to set an exposure level that did not cause any biological effect. For this purpose, the first International Congress of Radiology took place in 1925, in order to standardize measurements and units related to radiations. Figure 1.1 shows the official photo taken for the 5th International Congress of radiology. During this congress, the later on called International Commission on Radiation Units and Measurements (ICRU) was founded. Consequently, the ICRP was founded in 1928, during the second International Congress of Radiology. Contextually, a limit, for the ionizing rate exposure was established for a person in normal health, based on absence of reported biological effects.

This event marked a milestone in radioprotection. In order to combine radiation deposition to the tissue and damage to the tissue itself, absorbed dose concept has been adopted afterwards. In paragraph 1.2, the concept of absorbed dose will be described.



FOREIGN DELEGATES TO THE FIFTH INTERNATIONAL CONGRESS OF RADIOLOGY

Guests of the Eastman Kodak Company, Rochester, N. Y., September 9, 1937

- | | | | |
|---|--|---|---|
| <p><i>First (Bottom) Row—Left to Right</i></p> <ol style="list-style-type: none"> 1. Dr. OCSAK SOTO, Lima, Peru 2. Dr. GEORGE SCHMITZER, Bucharest, Roumania 3. Dr. HERMANN L. WINTZ, Erlangen, Germany 4. Dr. GILBERT H. BUSH, Bristol, England 5. Dr. JAVONKA MARKA, Moravská-Ostava, Czechoslovakia 6. Dr. ARTHUR PICKMAN, Berlin-Dahlem, Germany 7. Dr. BRUNO HOFFMANN, Leipzig, Germany 8. Dr. ATHANASIOS L. LAMBRANOS, Athens, Greece 9. Dr. KARL FAHNE, Berlin, Germany 10. Dr. EMMIL LAZARU, Bucharest, Roumania 11. EMMIL SCHWARZ, Zürich, Switzerland 12. Mrs. CARL A. SCHLUDENNER, Frankfurt am Main, Germany 13. Mrs. FERMINO VON DÖBLINGER DA GRAÇA, Rio de Janeiro, Brazil 14. Dr. FERMINO VON DÖBLINGER DA GRAÇA, Rio de Janeiro, Brazil 15. EMMETT W. G. WELSON, London, England 16. Dr. RUSSELL J. RAYMOND, London, England 17. Dr. HAROLD COURTNEY GAGE, London, England 18. Dr. ERICH VON SCHUBERT, Berlin, Germany 19. Dr. H. M. HANSEN, Copenhagen, Denmark 20. Dr. HANS R. SCHNITZ, Zürich, Switzerland 21. Dr. RUDOLF GRABNER, Cologne, Germany 22. Dr. PAUL FLEMING-NIELSEN, Copenhagen, Denmark <p><i>Second Row</i></p> <ol style="list-style-type: none"> 1. Dr. FRANTISEK BLONK, Prague, Czechoslovakia 2. Not identifiable 3. Dr. F. D. ANDREW, Rochester, N. Y., U. S. A. 4. Not identifiable 5. Dr. HANS HEINRICH BEAG, Hamburg, Germany 6. Dr. WERNER R. G. BLOHM, Rostock/Mecklenberg, Germany 7. Dr. ROBERT CAHILL, Paris, France 8. Dr. H. LA CHARITÉ, Montreal, Que., Canada 9. Dr. ROBERTO IREGUI, Bogota, Colombia | <ol style="list-style-type: none"> 10. Dr. JULIETTE BAUD, Paris, France 11. Dr. PAUL GRELET, Paris, France 12. Dr. KRAT LEISTNER, Dresden, Germany 13. Dr. LEONHARD GERTZ, Bonn, Germany 14. Dr. GERHARD KOSCHMANN, Oldenburg, Germany 15. Dr. WOLFGANG FAIS, Berlin, Germany 16. Dr. WERNER TESCHENDORF, Cologne, Germany 17. Dr. ARSIVUS HELM, Berlin, Germany 18. Dr. IOAN CHIRITESCU, Roumania 19. Mrs. EMMIL LAZARU, Bucharest, Roumania 20. Dr. CARLOS T. VENTURA, Bogota, Colombia 21. Dr. OSKAR WEIL, Prague, Czechoslovakia 22. Dr. BORIS RAJEWSKY, Frankfurt am Main, Germany 23. Dr. HANS SCHUBERT, Düsseldorf, Germany 24. MASAMICHI TANAKA, Kawasaki, Kanagawaken, Japan 25. EMMETT W. G. WELSON, London, England 26. Dr. DANIEL DES HONDS, Amsterdam, Holland 27. RAYMOND H. GREENMAN, Rochester, N. Y., U. S. A. 28. Dr. HANS VON BRAUNHORN, Freiburg, Germany 29. Dr. P. H. STEVENSON, Peiping, China <p><i>Third Row</i></p> <ol style="list-style-type: none"> 1. Dr. SOL C. DAVIDSON, Rochester, N. Y., U. S. A. 2. Dr. BENJAMIN J. SLATER, Rochester, N. Y., U. S. A. 3. Dr. G. H. RAMSEY, Rochester, N. Y., U. S. A. 4. EDWARD J. PERRON, Quebec, Que., Canada 5. Mrs. EMMIL J. PERRON, Quebec, Que., Canada 6. Dr. PAUL HENRI COTTENOT, Paris, France 7. Dr. IER SOLOMON, Paris, France 8. Dr. A. I. SILVERMAN, London, England 9. Dr. WILLIAM TRENANT, Glasgow, Scotland 10. Dr. G. GANATHAN, Calcutta, India 11. GODFREY PEARCE, London, England 12. JOHN T. CALDWELL, Harrow, England | <ol style="list-style-type: none"> 13. Dr. FRANK L. HOPWOOD, London, England 14. Dr. GUSTAV SCHULTE, Rockinghamen, Germany 15. Dr. G. FREDERICK HAENTGEN, Hamburg, Germany 16. Dr. WALTER FRIEDRICH, Berlin, Germany 17. Dr. PAUL SCHLÖDER, Böttrop i. Westfalen, Germany 18. Dr. KARL MARHOLT, Vienna, Austria 19. Dr. GUSTAV HEINRICH, Prague, Czechoslovakia 20. Dr. FRAIN FELDSCHNEIDER, Vienna, Austria 21. Dr. ERWIN SCHLIEPMACK, Giessen, Germany 22. Dr. CARL EGON KOCH, Cologne, Germany 23. Dr. HERMANN BRONKHOF, Berlin-Charlottenburg, Germany 24. Dr. BUKHARD WEBER, Berlin-Friedenau, Germany 25. Dr. GÖRKE A. WELTZ, Munich, Germany 26. Dr. HERMANN HOLTREITER, Hamburg, Germany 27. Dr. F. J. FARR, Bristol, England 28. Dr. CARL A. SCHLUDENNER, Frankfurt am Main, Germany 29. Dr. LEWELYN J. SANDERS, Rochester, N. Y., U. S. A. <p><i>Fourth Row</i></p> <ol style="list-style-type: none"> 1. Dr. JAMES M. FURSH, Rochester, N. Y., U. S. A. 2. Dr. LEONWKA LANDES-LEISER, Lodow, Poland 3. Dr. EMIL MEISSEL, Iwów, Poland 4. Dr. MARY WIKKERTSIN, Warsaw, Poland 5. Dr. PAUL JACOB, Odense, Denmark 6. Dr. JEAN JOVIN, Bucharest, Roumania 7. Dr. JOHN B. KING, Edinburgh, Scotland 8. Dr. ROBERT McWHIRTER, Edinburgh, Scotland 9. Dr. CARL PEUS, Berlin-Dahlem, Germany 10. Dr. GEORGE BRUGGIER, Berlin, Germany 11. Dr. ZAKAR TARTER, Istanbul, Turkey 12. ALFRED UNOLDEN, Rudolstadt, Germany 13. Dr. JOSEF G. HERGEN, Würzburg, Germany 14. Dr. M. WELT, Berlin-Wittenau, Germany 15. Dr. ELISABETH M. SVYKES, Sheffield, England | <ol style="list-style-type: none"> 16. Dr. FRANCISCUS S. P. VAN BUCHEN, Tilburg, Holland 17. Dr. ANNA HAMANN, Hamburg, Germany <p><i>Fifth Row</i></p> <ol style="list-style-type: none"> 1. Dr. JOHANNES PITZOLD, Erlangen, Germany 2. WALTER KOLBAY, London, England 3. Dr. HANS F. M. MARTENSTEIN, Dresden, Germany 4. CARL NIEMANN, Buchenhof, Germany 5. Dr. KATOWITZ, Poland 6. Dr. OETA CYRORI, Budapest, Hungary 7. Dr. EMIL BERGMANN, Berlin-Halensee, Germany 8. Dr. HELLA BERGMANN-ROGGMANN, Berlin-Halensee, Germany 9. Dr. KOLJ B. EGGESTAD, Oslo, Norway 10. Dr. KARL ECKLMANN, Hamburg, Germany 11. Dr. ADALBERT LÖW-BEER, Prague, Czechoslovakia 12. GERMANY SCHWAB, Vienna, Austria 13. Dr. WALTER HESSE, Berlin-Halensee, Germany 14. Dr. WILHELM TRAUH, Berlin-Frohau, Germany <p><i>Sixth (Top) Row</i></p> <ol style="list-style-type: none"> 1. Dr. M. CLAUDIUS VON SICHERER, Munich, Germany 2. Not identifiable 3. Dr. ALBERT A. HEDFIELD, Magdeburg, Germany 4. Dr. ERICH SAUER, Dresden, Germany 5. Dr. JOHANNES THEISSER, Berlin, Germany 6. Dr. IMRE A. L. KOROMAI, Budapest, Hungary 7. Dr. OTTO DRES, Würzburg, Germany 8. Dr. HERMANN MEYER-BUCHGROFF, Lübeck, Germany 9. Dr. JENS JENSEN, Copenhagen, Denmark 10. Dr. HERBERT PRICHLAT, Freiburg, Germany 11. Dr. FELIKS STUMPF, Munich, Germany 12. Dr. THEOPHIL REES, Karlsruhe/Baden, Germany 13. Dr. GOTTFRIED MARKE, Cologne, Germany 14. Dr. CARL FRIED, Breslau, Germany 15. Dr. CURT WITTKOWSKY, Guatemala City, Guatemala |
|---|--|---|---|

Figure 1.1: 5th congress of International Society of Radiology [4]

In the years that follow the 2nd World War, radiation protection knowledge and techniques/tools experienced radical changes. New techniques have been discovered, allowing to retrieve more information on the type of radiation exposure.

From the 40s, thanks to the contribution of Louis Harold Gray and John Read, neutrons effect on tissue was studied. The main idea was to compare effects of α -particles and neutrons with respect to gamma and X rays. From these studies, the Relative Biological Effectiveness (RBE) of neutrons was retrieved and a new unit of measurement, called energy unit, was proposed. This definition represented a milestone in the dosimetry history since it was the precursors of the later called absorbed dose, based on the radiation interaction with matter instead of pure radiation exposure or intensity. This led to the awareness that biological effects were related to the equivalent dose.

In the following years, ICRP and ICRU have decided to standardize the unit of measures for absorbed dose adopting the [rad] unit. This measure has been later introduced in the recommendation document published by ICRP in 1954 [5]. Due to the rising concern regarding radiation-related effects, in 1959 the ICRP released a document in which eye lenses, gonads and blood forming organs have been described as critical organs due to their sensitivity to radiations [6]. Sensitivity of blood forming organs are related to a publication, back in 1906, by Jean Bergonié and Louis Tribondeau. This publication highlights that “cells are inasmuch radiosensitive as they grow fastly”. Although this empirical law has never been fully validated “it has made a significant contribution to the advances in radiation biology and the relationship between proliferation and radiosensitivity” [7]. With the previously cited publication of 1959, by ICRP, a list of permissible doses to these sensitive tissues for a large quantity of nuclides was included.

Until those years, a tolerance dose was adopted below which it was thought that no harmful effects could be detected. This attitude changed after the publication of an updated recommendation document by ICRP in 1966: the idea of a linear relationship between dose and somatic or genetic effects seemed to be a more reasonable option. This no-threshold model has been proved to avoid underestimation of harmful effects and led to a change from deterministic to stochastic effects. Consequently, in the recommendations released in 1977 by ICRP [8], the three main principles of radioprotection have been eventually coined:

- Justification: any exposure to radiation must bring greater benefits to the community with respect to damages;
- Optimization: any justified exposure must include the minimum reasonably possible doses to the population, according to the dictates of the good technique. This is also called the ALARA (As Low As Reasonably Achievable) principle;
- Dose Limit: any justified and minimised exposure must not result in any dose exceeding the legal limits. [9]

According to these principles a cost-benefit analysis should lead to the optimum level of protection.

After the first introduction in 1975 by Wolfgang Jacobi, the concept of effective dose was adopted [10]. The current knowledge about cancer risk was adequate to permit the calculation of a weighted whole-body dose, leading, in 1977, to the adoption in ICRP recommendations of the new quantities named dose equivalent and effective dose equivalent, evaluated in sievert [Sv].

Report number 39 and 43 provided by ICRU, respectively in 1985 and 1988, set a standard for the evaluation of the dose equivalents resulting from exposure by sources external to the body [11], [12]. The first report states the definitions of the quantities used for the monitoring including ambient, directional and individual dose equivalent, while the second report focuses on supporting the selection of these quantities and on providing the basis for their definition. In order to implement these dosimetric quantities for radiation protection, IAEA held a technical meeting in the following years: the International Organization for Standardization (ISO), as well as the International Electrotechnical Commission (IEC), whose objective is to release standardization regarding dosimetry and radioprotection. The first ISO standard (ISO 8963:1988) on dosimetry of X and gamma rays has been constantly updated during the years, leading to the latest standards, released in 2019 (ISO 4037-2:2019). The IEC 61066:1991 standard on test and performance criteria for dosimeters has been replaced in 2006. During the years, many standards have been published and updated from ISO and IEC, covering all the aspects regarding radiation protection.

In the 90s, concepts of Equivalent Dose and Effective Dose have been introduced to replace previous defined quantities with averaged ones, extensively explained in paragraph 1.2.

1.2. Dosimetry

Dosimetry represents a fundamental aspect of radiation protection. It is centred on measuring, assessing and understanding radiation doses, due to exposure to ionizing or non-ionizing radiations, in order to ensure safety for individuals, populations and environment. This practise is of crucial importance in many sectors including medicine, industry, environmental protection and research.

1.2.1. Dosimetric and protection quantities

Dosimetric quantities are established by ICRP recommendations, which latest version refers to 2007 [2]. The following list defines the main quantities used nowadays in personal dosimetry and explains the relationships between them. Dosimetric quantities are used to quantitatively assess the radiation exposures to humans, which are necessary to retrieve the biological effects related to a certain level of exposition by means of protection quantities. These are fundamental for the risk estimation and are used to specify exposure limits that ensure low levels of stochastic biological effects. All the quantities summarized afterwards are universally defined by ICRU in “Fundamental quantities and units for ionizing radiation” [13].

Exposure

“Exposure is a measure of the ability of an X-ray beam to ionize a mass of air; i.e., it expresses the amount of electrical charge of electrons (Q) generated per unit mass of air (m)” [14]. It is identified by X and has been defined as the ratio between the absolute value of the total charge of the ions, dQ , that are produced in air, and the mass dm :

$$X = \frac{dQ}{dm} \quad (1.1)$$

The International System of Unit (SI) unit is $[C/kg_{\text{air}}]$, while the conventional unit for exposure is the röntgen [R], equivalent to $2.58 * 10^{-4} C/kg$. This quantity can be measured but does not provide information regarding the biological damage provided to the individuals due to the exposition.

Absorbed dose

This quantity is identified by D and it has been defined as the following ratio:

$$D = \frac{d\bar{\epsilon}}{dm} \quad (1.2)$$

in which $d\bar{\epsilon}$ identifies the mean energy imparted to the mass dm by ionizing radiations. The unit of measure for the Absorbed dose in SI unit is $[J/kg]$, also called gray [Gy], equivalent to 100 rads (unit of measurement previously used). This quantity represents the basic dose evaluation, and it has been defined for all types of ionizing radiation and geometry.

For practical applications the averaged value of absorbed dose on a tissue is evaluated. In practise, for the purpose of biological effects evaluation, is worthless to take into account values of absorbed dose in specific points of the body. The mean absorbed dose in a specific region of organ or tissue T has been defined as following:

$$D_T = \frac{1}{m_T} \int_{m_T} D dm \quad (1.3)$$

In this definition m_T represents the mass of the organ or tissue.

Kinetic Energy Released to Matter (KERMA)

The KERMA (Kinetic Energy Released to Matter) is identified by K and it has been defined as:

$$K = \frac{dE_{tr}}{dm} \quad (1.4)$$

in which dE_{tr} identifies the mean sum of the initial kinetic energies of all the charged particles liberated, in a mass dm of a certain material, by the uncharged particles incident on that same mass. The unit of measure for the kerma is equal to the one of Absorbed dose: $[J/kg]$ also called gray [Gy]. Assuming some hypothesis, kerma can be used as an approximation of the absorbed dose.

Linear Energy Transfer (LET)

The LET has been defined as the mean energy lost by charged particles due to electronic interactions dE in a travelled distance dl . The definition is the following:

$$LET = \frac{dE}{dl} \quad (1.5)$$

The SI unit for LET is $[J/m]$, often given in $[keV/\mu m]$.

Relative Biological Effectiveness (RBE)

The latest definition of RBE outlined by ICRP in their annals is reported in publication 136, published in 2017. It defines RBE as “the ratio of absorbed dose of a low-linear-energy-transfer reference

radiation to absorbed dose of the radiation considered that gives an identical biological effect. RBE values vary with absorbed dose, dose rate, and biological endpoint considered" [15]. Accordingly, the RBE factor related to a radiation type R has been evaluated as:

$$RBE = \frac{D_X}{D_R} \quad (1.6)$$

Where D_X is the reference absorbed dose of radiation of standard type X and D_R is the absorbed dose of the radiation of type R that causes the same amount of biological damage. Different types of radiation have different biological effectiveness since they transfer energy to the tissue by different mechanisms.

Equivalent dose

Equivalent dose quantifies the biological damage that relates the absorbed dose and the type of ionizing radiation: same dose coming from different types of radiation can result in different effects. In order to evaluate the equivalent dose, the biological effectiveness of the radiation has been taken into account. This term depends on the type and energy of radiation. These information are contained in the so-called radiation weighting factor W_R . The equivalent dose for a specific tissue (H_T) is evaluated as:

$$H_T = \sum_R W_R D_{T,R} \quad (1.7)$$

The term $D_{T,R}$ represents the absorbed dose in tissue T by radiation type R . The SI unit for the equivalent dose is [J/kg], commonly known as sievert [Sv]. Weighting factors for different types of radiations have been defined by ICRP and ICRU [2], [16] and are summarized in Table 1.1.

Radiation type	Weighting factor (W_R)
Photons	1
Electrons and muons	1
Protons and charged pions	2
Neutrons	Continuous function of energy
α -particles, fission fragments and heavy ions	20

Table 1.1: radiation weighting factors for equivalent dose evaluation [2]

Weighting factors for photons (X and γ rays), electrons, muons and secondary particles generated by photons has been set equal to 1. For safety purposes a weighting factor equal to 2 has been given to protons of all energies. While, due to strongly dependence of biological effectiveness of neutrons with energy, a continuous function has been used for evaluation of neutron weighting factors, as reported in Figure 1.2. Further analytic details on the function can be found in the ICRP publication number 103 [2]. For what concern α -particles, heavy ions and fission fragments, since they can cause significant damage to tissue and organs in case of internal emitters, a higher weighting factor of 20 has been given.

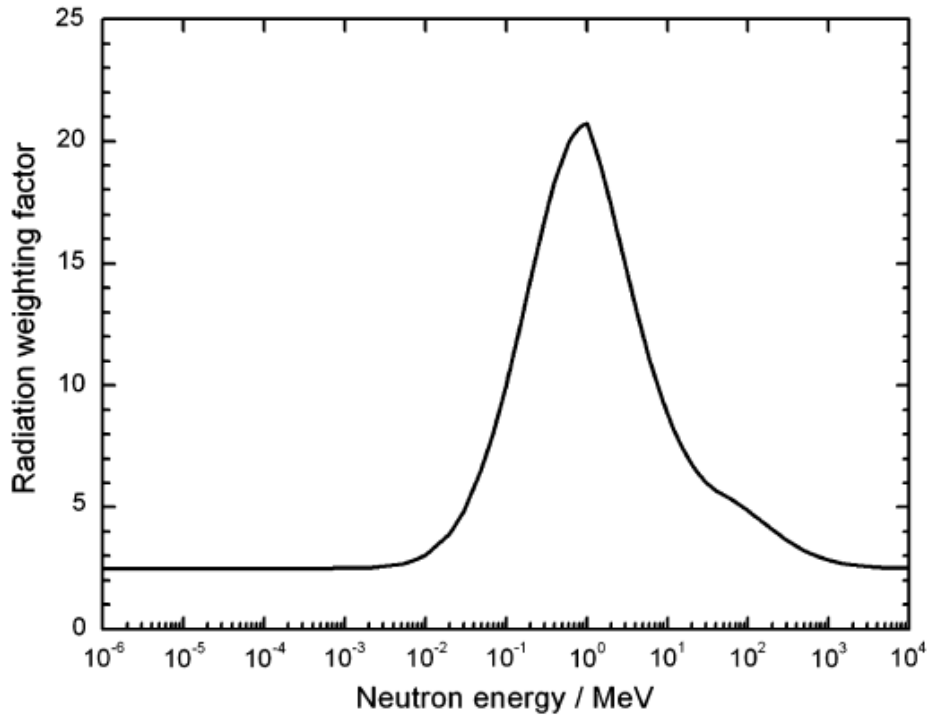


Figure 1.2: weighting factor function for neutron radiation [2]

Effective dose

The effective dose is identified by E and represent a tissue-weighted sum of the equivalent doses. According to ICRP the effective dose is obtained from equivalent dose applying defined weighting factors to take into account differences in biological effectiveness of various organs and tissues. Thanks to this approach, differences in sensitivities to stochastic health effects for different organs and tissues have been also considered. The expression for the evaluation of the effective dose is the following:

$$E = \sum_T W_T \sum_R W_R D_{T,R} = \sum_T W_T H_T \quad (1.8)$$

The factor W_T represents the sex-averaged tissue weighting factor: all the possible values for this term are listed in Table 1.2. The evaluation of the effective dose is performed over organs deemed sensitive to stochastic effects induction. The SI unit of measure is still [J/kg], commonly called sievert [Sv].

Organ or tissue	Weighting factor (W_T)	Sum of weighting factors ($\sum_T W_T$)
Red bone marrow, Colon, Lung, Stomach, Breasts, remainder tissues	0.12	0.72
Gonads	0.08	0.08
Bladder, Liver, Esophagus, Thyroid	0.04	0.16
Skin, Bone surface, Salivary glands, Brain	0.01	0.04
Total		1

Table 1.2: tissue weighting factor for effective dose evaluation values [2]

The remainder tissues cited in the table refer to: adrenals, extrathoracic region, gall bladder, heart, kidneys, lymphatic nodes, muscle, oral mucosa, pancreas, prostate (just for male individuals), small intestine, spleen, thymus, uterus/cervix (just for female individuals). The sum of the weighting factor W_T for these remainder tissues and organs is equal to 0.12. All weighting factors sum up to 1 in order to have effective dose equal to the equivalent one when the body is uniformly irradiated by penetrating external radiations. Higher weighting factors have been given to tissues and organs that are more sensitive to radiations, as the red bone marrow.

Dose limits

In 2007, publication 103 [2] has been released by ICRP containing an updated version of radiation and tissue weighting factors and a new limit on effective dose. The limits have been set, by this publication, to 20 mSv per year averaged over 5 years for occupational exposure. Limit for eye lenses have been set equal to 150 mSv, while, for what concern skin and extremities, a limit of 500 mSv has been adopted. Regarding public exposure, limit on effective dose is equal to 1 mSv per year, but in special circumstances a higher value could be allowed in a single year provided that the average over 5 years does not exceed 1 mSv/y. These limits apply only to doses received above the local natural background radiation and, in addition, these limits have to be intended for planned exposure radiations, in other cases reference levels have been specified. These levels assure the flexibility needed in emergency and existing exposure situations.

1.2.2. Operational quantities

Due to the fact that the previously defined quantities cannot be measured directly, the so-called operational quantities have been defined. In the reports released in 1985 and 1988 by the ICRU [11] [12], a set of operational measurements has been defined: these quantities are directly measurable in individual monitoring and can be used to retrieve the protection quantities defined in the previous paragraph. In order to achieve this purpose, Individual dose equivalent penetrating, $H_p(d)$ and Individual dose equivalent superficial, $H_s(d)$ have been defined.

These quantities are defined at an appropriate location and not as average quantities or over an extended mass. The estimation, for the protection quantities defined above, is meant to be conservative, related to exposures or potential exposures. They are calculated based on fluence knowledge at the locations of interest, allowing a calibration process based on these quantities.

Personal dose equivalent

Concerning individual monitoring, the established quantity is called personal dose equivalent, identified as $H_p(d)$. This represents the dose equivalent in soft tissue at an appropriate depth d , below a specified point on the individual body. Recommended depths are: 0.07 mm for superficial radiations, 3 mm for eye lens dose and 10 mm for penetrating radiations. $H_p(0.07)$ is used to evaluate skin effects and dose given to the extremities, $H_p(3)$ provide information about eye lens dose and $H_p(10)$ gives information about the effective dose intake by the individual. As for effective and equivalent dose, the SI unit of measure is [J/kg], commonly called sievert [Sv]. The specified point at which this dose is evaluated depends on the location at which the personal dosimeter is placed on the individual's body. Formulas for the evaluation of this personal dose equivalent will be explained in detail for the dosimeter of interest in paragraph 2.1.

Ambient dose equivalent

The ambient dose equivalent, identified as $H^*(d)$, is defined in the ICRP publication 103. It is defined as the “*dose equivalent at a point in a radiation field that would be produced by the corresponding expanded and aligned field in the ICRU sphere*” at a depth (d) on the radius vector “*opposing the direction of the aligned field*” [2].

The unit of measure for the ambient dose equivalent is [J/kg], but commonly called sievert [Sv].

1.2.3. Dosimeters

In order to prevent the occurrence of deterministic effects as well as reduce the probability of stochastic effects, a dose evaluation is needed for all the individuals that are exposed to radiations daily or occasionally. This dose assessment for external irradiation is commonly performed by means of individual monitoring. Major milestones for personal dosimetry are represented by ICRP publications 60 [17] and 103 [2], respectively published in 1991 and 2007.

Personal dosimeters are usually provided to workers in order to evaluate the dose by direct measuring. In some peculiar cases, computational dosimetry has been allowed by EURATOM (EUROPEAN ATOMIC energy community) directives for individual dose assessment, as for aircrew dose assessment. Computational tools such as Monte Carlo simulation or Machine Learning algorithms are nowadays used to evaluate the dose received from patients and medical personnel.

Briefly summarizing the history of dosimeters, the continuous development of instruments and quantities has been decisive along the history of dosimetry. First dosimeters were developed in the 20s and they were constituted by a pocket ionization chamber and a film, ancestors of the modern pencil dosimeters. In 1934, some years after the foundation of the ICRP, the “dose-meter” name was given to an instrument used to measure X-ray quantities, calibrated in röntgens. In Figure 1.3 one of the first dosimeters is represented.

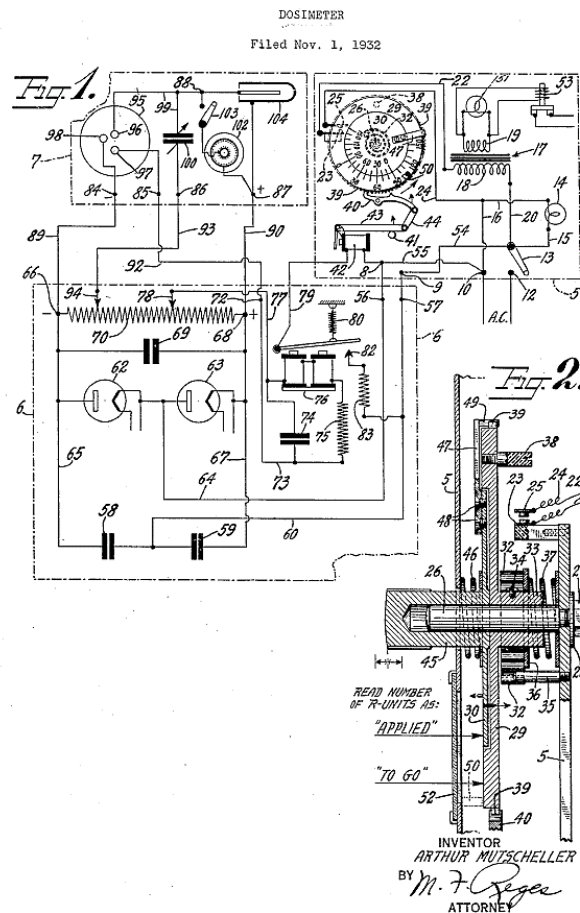


Figure 1.3: dosimeter patent by Arthur Mutscheller n° US-2036072-A [18]

During the Manhattan project, film dosimeters were adopted, although it was already clear that low energetic photons were difficult to detect. At the end of the project, previously used pocket ionization chamber have been substituted with direct-reading chambers, allowing the worker to evaluate the exposure received at any time [19]. After the 2nd World War, dosimeters based on new techniques were proposed, such as Thermo-Luminescence Detectors (TLDs), proposed in 1953 by F. Daniels [20].

In 1997, ICRP released publication 75 [21], in which the maximum permissible measurement deviation of a personal dosimeter, depending on dose, have been specified. From 1997 onwards, several new passive and electronic dosimeters have been developed alongside with standardized calibration procedures. Concerning direct reading dosimeters, in 1997 IEC have released their standard for design requirements and performances evaluation named IEC 61526:1997 (revised in 2017). Film-badges have been commonly used as personal dosimeters, however, from the beginning of the new century, they have been replaced. This was related to a no longer sufficient performance for complying with the new regulations, mainly due to energy dependence and sensitivity to humidity.

As far as the measurement is concerned, effective dose is assessed by means of $H_p(10)$ measurements on a representative location of the body. Personal dosimeters are usually worn around the chest to represent the “whole-body” dose, however they can also be worn on extremities or near the eyes in peculiar cases, as shown in Figure 1.4.



Figure 1.4: whole-body and extremities dosimeters provided by and used at SCK CEN [Photo courtesy of SCK CEN]

Dosimeters can be divided in two major groups: passive dosimeters, which accumulate the dose over a certain period of time, and active dosimeters, which instead provide a real-time measurement, useful to improve radiation-protection measurements. The following paragraphs briefly explain the operation of the main passive dosimeter technologies, focusing on new generation technologies such as the Direct Ion Storage (DIS) dosimeters.

Film-badge dosimeter

As previously mentioned, film-badge dosimeters have been introduced for the first time around the 30s. They are small portable devices based on the presence of photographic film sensitive to ionizing radiations. They are able to monitor cumulative radiation dose: radiations that struck the film cause a progressive darkening of the film itself due to the reduction of the silver halide contained in the material [21]. After a certain period of time, the dosimeter is withdrawn: thanks to the developing of the photographic film and its darkness it is possible to estimate the radiation dose received by the individual. Dosimeters based on these technologies are passive, and real-time measurements are not possible since the dosimeter has to be collected and processed. However, these film processing techniques are time consuming and requires specialized equipment.

Dosimeters of this kind were one of the most commonly used due, for example, to low cost of the utilized films and the ease of use and construction. In addition, they are sensitive to many different types of radiations, such as α , β and γ . Usually, the photographic film is covered with various filters in different positions: in this way a part of the film remains uncovered in order to measure the total exposure, while other part of the film manage to measure different contributions related to distinct kind of radiations [22]. A representative example of film-badge dosimeter is reported in Figure 1.5.

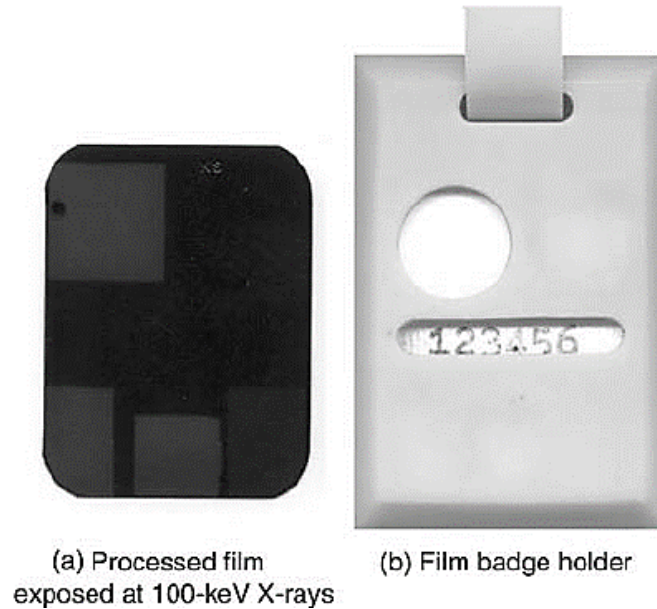


Figure 1.5: film-badge holder and processed film [23]

Nonetheless, many drawbacks have led to the substitution of these devices during the years. Main disadvantages are:

- Lack of real-time measurements: as specified before, these devices are not able to provide real-time measurements. The period of time that elapses from the shipment of the device back to the laboratory and the assessment of the dose received could be significant;
- Limitation on the measurement of high levels of radiations: in high radiation environments the film can saturate leading to problems in the evaluation of exposure levels;
- Susceptibility on environmental factors: photographic film is sensible to environmental factors such as humidity [24], temperature and light exposure. For this reason, it is of primary importance to store and use the dosimeter properly;
- Possibility of errors during the processing: film development procedure can be subjected to errors that impact the accuracy of radiation dose assessment leading to inaccuracies in the results;
- Single use of photographic films: in order to maintain reasonable levels of detection, new unexposed films must be used each time [22].

Thermo-Luminescence Detectors (TLDs)

Thermo-luminescence detectors have been proposed for the first time in 1953 by F. Daniels. Contextually, he also provided a definition of Thermoluminescence: “*Thermoluminescence is the emission of light produced by heating a solid to a temperature below that of incandescence. It is exhibited by crystals, such as alkali halides, that have been exposed to x-rays or radioactivity and then heated rapidly*”. This definition has been lately proved by F. Daniels, C. A. Boyd and D. F. Saunders: they discovered that high-energy radiations dislodge electrons in the lattice of these crystals. Once the necessary amount of kinetic energy is supplied, in case of TLDs due to the increase of temperature, the previously excited atoms return to ground state providing emission of photons. Several ways can be used to supply the required amount of energy, leading to several types of scintillation detectors that rely upon different technologies to measure luminescence. Most common technologies are Optically

Stimulated Luminescence (OSL), Radio Photo Luminescence (RPL) and TLD: in the first two, the luminescence process is triggered by a light flash, while heating is used for TLDs.

Once the energy has been supplied, the emitted light has to be quantified in order to assess the amount of absorbed dose. This process has been named luminescence dosimetry: “*Luminescence dosimetry is the process of quantifying the absorbed dose of ionizing radiation using detectors that exhibit luminescence*” [25]. Various thermoluminescent materials can be used inside the detector, most common are lithium fluoride (*LiF*) activated with magnesium and titanium and calcium fluoride (*CaF₂*). Filters can be used in case of calcium fluoride to make it tissue-equivalent [24]. In Figure 1.6 the relative thermoluminescent intensity is plotted with respect to temperature, the plot refers to lithium fluoride.

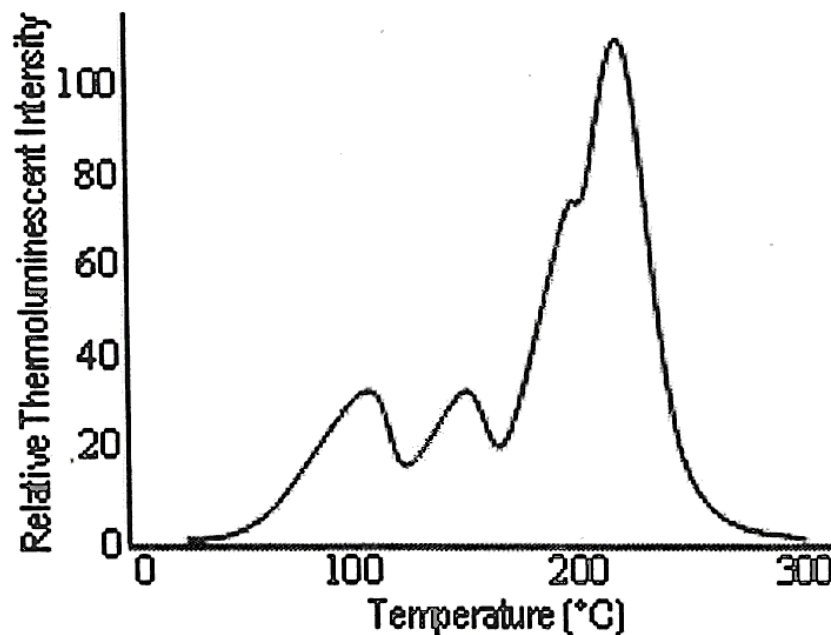


Figure 1.6: thermoluminescent glow curve for LiF: Mg, Ti [26]

Concerning the capability of retaining the information, TLD are less affected by environmental factors with respect to photographic films. Even if some fading effects are inevitable, with any kind of TLD material, these are not of primary importance for personnel monitoring due to short time intervals between two reads, usually limited to a couple of months. Problems could arise if these dosimeters are used in environmental or workplace monitoring in which larger intervals are used [22].

Moreover, this kind of dosimeter have many advantages:

- Good linear response and high sensitivity: TLDs response to dose is mainly linear. In addition they have a good sensitivity to low doses leading to a great range of doses that can be measured, compared to film-badges dosimeters [27];
- Ease of reading: dose assessment is obtained easily and can be carried out on site, without shipping the dosimeter for assessment;
- Reusability: unlike film-badge dosimeters, TLDs can be easily reused;

However, a main drawback is related to thermoluminescent dosimeters: due to the physical phenomena on which the reading process is based, each read cancels the information stored in the dosimeter with a zeroing of the TLD device. In addition, due to thermoluminescence phenomena, TLD devices are sensitive to high temperatures: accidental exposure to heat sources can affect accuracy and calibration of the device.

Direct Ion Storage (DIS) dosimeters

“*The Direct Ion Storage (DIS) dosimeter is based on the coupling of a gas-filled ion chamber with a semiconductor non-volatile memory cell*” [28]. This technology has been developed to overcome the

aging and problems of the widely spread TLDs and to substitute them with an electronic alternative. In particular, these devices are passive dosimeters able to store information regarding the accumulated dose and to be read, after a certain period of time, without losing these stored information. In addition, DIS dosimeters can be read directly on site, avoiding the periodical shipment of dosimeters to the laboratory for the estimation and validation of the dose. This also allows daily reads of the dosimeter that were not possible with previous technologies. However, the DIS dosimeters have to be recollected to the laboratory in case of too high doses, low battery, unusual behaviour or after a certain period of time.

The basic detection unit inside the DIS dosimeters is composed by an ionization chamber combined with a modern electronic DIS memory cell. These kind of memory cells, called EEPROM (Electrically Erasable Programmable Read-Only Memory) and represented in Figure 1.7 a, have spread in the 90s thanks to the possibility of analogic storage. Indeed, contrary to standard memory cells, these devices can store a variable analogic voltage for indefinite time. These memory cells could be used as detector, since highly energetic radiations manage to penetrate inside the oxide layer and change permanently the voltage. However, in order to create a detector that has a sensitivity that could be useful for radiation protection purposes, changes have been applied. As shown in Figure 1.7 b, the control gate has been removed, as well as one side of silicon oxide layer, moreover an ionization chamber has been added. These changes allow the exposition of the top surface of the floating gate to the gas of the ionization chamber. In this way, an easier transportation of ion-electron pairs from the gas to the gate is achieved. As in a normal ionization chamber, the gas is contained within a conductive outer casing and gets ionized by incoming radiations. An electric field is generated by the initial voltage applied to the floating gate, this manage to drift the ion-electron pairs that have been created. The deposition of these ionization charges on the floating gate modifies the charge on the floating gate causing a change in the source-drain conduction. In order to avoid recombination within the gas, the electric field should be high enough in all the area of the chamber. For this reason the maximum dimensions of ionization chambers in DIS dosimeters is limited to few millimetres, based on typical voltages applied on the floating gate [28].

Measuring the resistivity variation of the channel, shallow part of material that connect source and drain (shown in Figure 1.7), it is possible to read the stored information without interfering with the charge on the floating gate, as mentioned before.

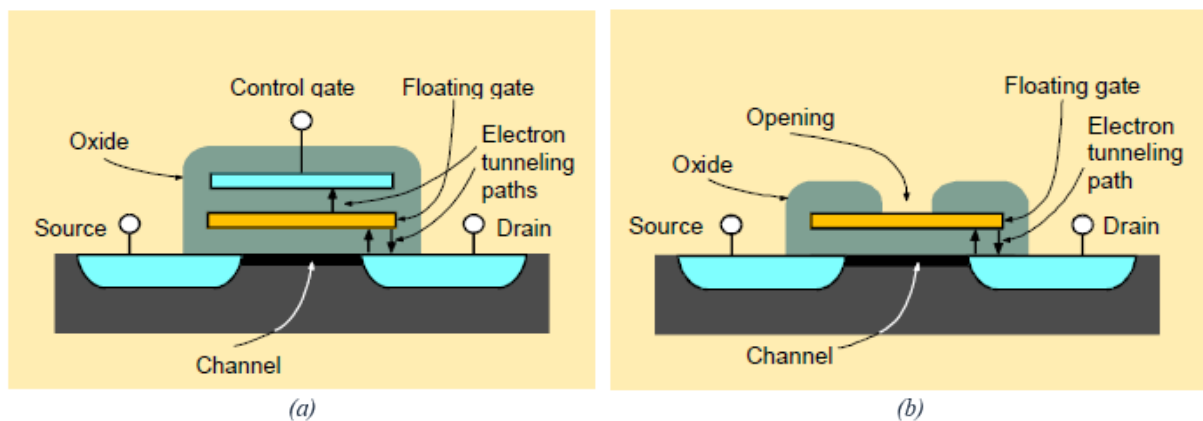


Figure 1.7: (a) EEPROM memory cell (b) DIS memory cell [29]

DIS dosimeters are usually constituted by different detection units that aim to collect deep and shallow doses, respectively $H_p(0.07)$ and $H_p(10)$, on different ranges. These ranges can vary from one dosimeter to another. Since these devices are based on ionization chambers, they are able to detect α -particles, β -particles, γ -rays and X-rays. Clearly, sensitivity of these detection units changes with the type of incoming radiation. In addition, thanks to the design of these dosimeters, they are not affected by very strong electro-magnetic or radio frequency fields [30].

Active Personal Dosimeters (APD)

Eventually, Active Personal Dosimeters (APD) are going to be briefly introduced since they represent one of the most recent dosimeter technologies. They aim to remedy limitations of passive dosimeters:

lack of direct dose display and absence of alarm indication for high dose rate. Indeed, among the many advantages presented by this technology the major one regards the possibility of an immediate read-out. Thanks to this feature, the workers may be more aware of the dose they are receiving. This dosimeters type is also generally characterized by a lower detection limit and data transfer possibilities.

However, some disadvantages are present, and they are mainly related to the dosimeter cost, mass and size. Due to the system used for storing and transferring of data, lack of security can be also foreseen [31].

1.3. Machine Learning

Machine learning is a subset of Artificial Intelligence (AI) field, which is based on the concept of developing computational algorithms to execute any kind of tasks, in order to make them learn the structures underneath data feed to the algorithms and perform different predictions and decisions without following explicit instructions. Machine learning field is huge and continuously growing, leading to development of modern algorithms and technologies. It is possible to distinguish among three main subfields in ML called: unsupervised learning, reinforcement learning and supervised learning, as summarized in Figure 1.8.

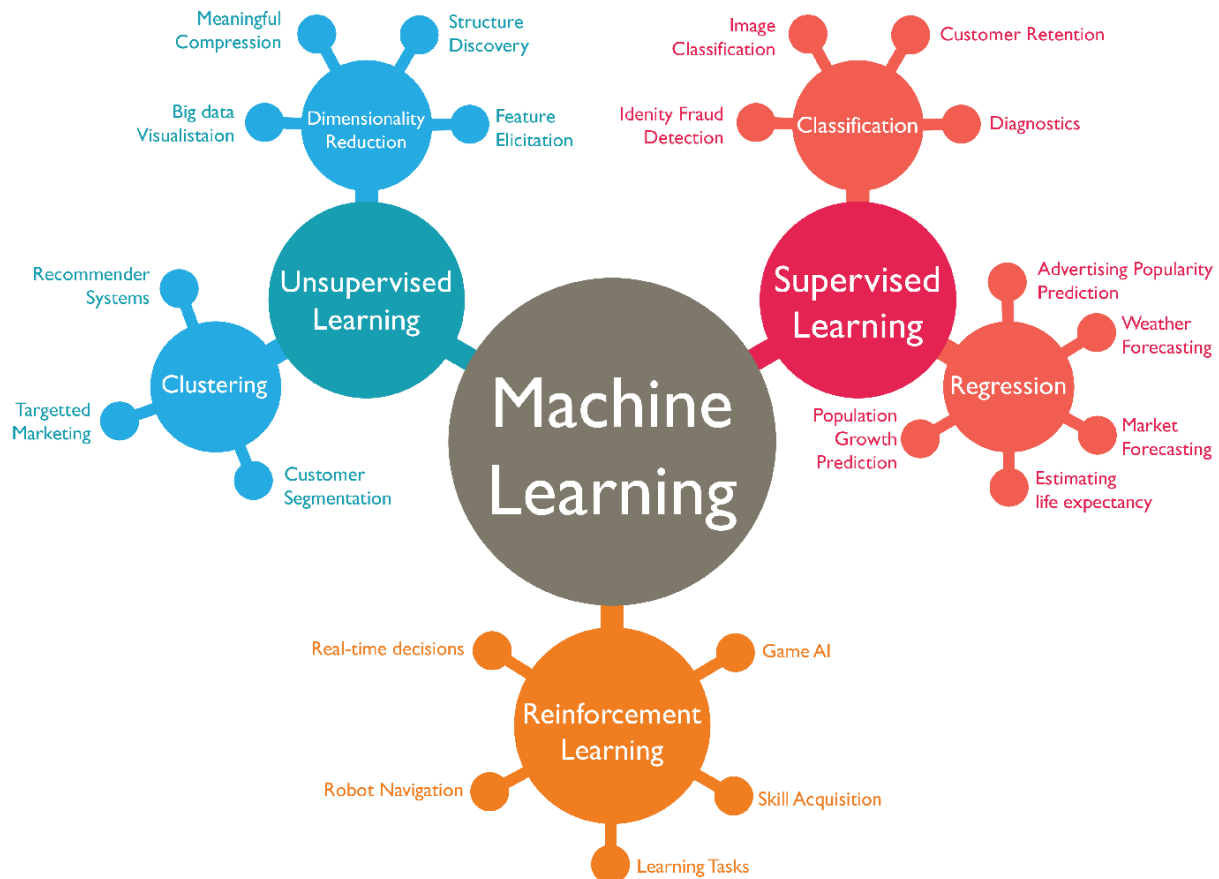


Figure 1.8: scheme for description of machine learning subgroups [32]

1.3.1. Unsupervised learning

Unsupervised learning aims to find hidden patterns or structures within provided unlabeled data. These data do not include any output: the model aims to provide them. In order to organize and understand complex data, clustering and dimensionality reduction techniques are commonly adopted. Clustering processes aim to group similar data points based on similarities or distances in a feature space, as represented in Figure 1.9, where it may be possible to appreciate also the difference between supervised and unsupervised learning. The term “feature” in machine learning relates to a variable or attribute of

the data that is used for prediction purposes. The implementation of unsupervised learning models with complex datasets makes use of dimensionality reduction techniques and Principal Component Analysis (PCA), which aims to reduce the number of features, for more efficient computational purposes, preserving the important information. Many other techniques are typical of unsupervised learning but are not objects of this study [33].

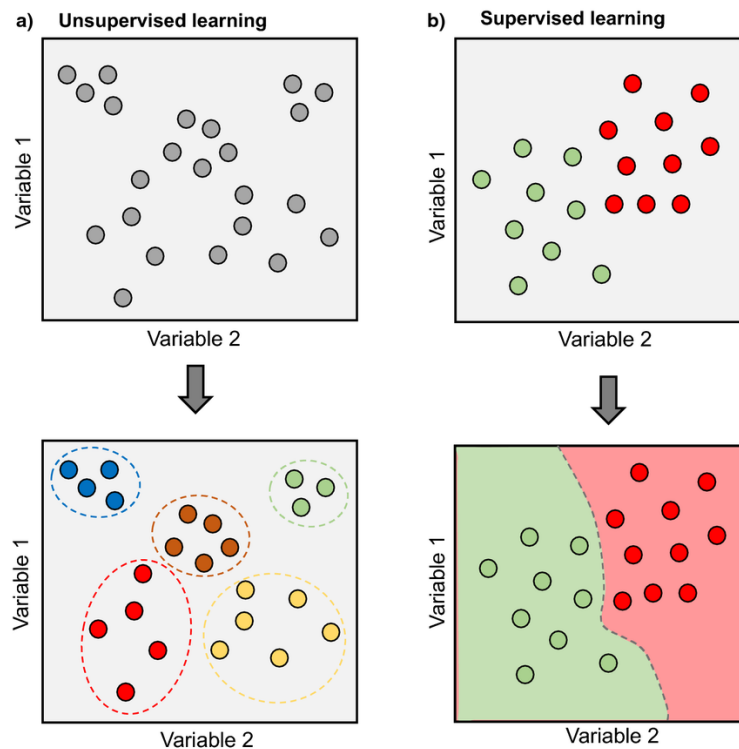


Figure 1.9: general scheme for distinction of supervised and unsupervised learning [34]

1.3.2. Reinforcement learning

Reinforcement learning algorithms aim to perform sequential decisions supported by interaction with an environment. These models try to replicate the “learning by experience” process of human and animals in which they maximize the reward while navigating through the environment. In this field the concept of environment relates to an external system with which the learner, or decision-maker, interacts and where actions are taken. Decisions about the actions to take are based on the environment and the state: the term state, in reinforcement learning, identify a specific configuration or situation perceived by the learner from its environment at a particular time during the learning process. Reinforcement learning is typically applied to robotics or game playing, like chess or video games. For example, in a chess game the environment is represented by the chess board, the state by the surrounding pieces and each action taken by the model has a certain “reward” related to the environment itself. One of the main peculiarities of these models is the ability to take into account a long-term reward [35].

1.3.3. Supervised learning

Supervised learning involves the use of labeled dataset for model training purposes. It is also defined as inductive reasoning method since it is opposed to deductive methods. In this case, labeled outputs are provided for each input data. Basically, many examples are feed to the model, in order to make it learn the intrinsic relations on which input and correspondent output rely, providing procedures or rules to follow in order to make future predictions of new unlabeled data. The algorithms used for these tasks aim to minimize the error between provided and predicted results adjusting the parameters that the model use for making the predictions. Supervised learning is widely used in many fields, such as business, medicine and public policy. Algorithms that belong to this realm are broadly categorized in classification and regression. These two types are based on the nature of the predictive task: regression aims to predict

continuous and numerical values, while classification aims to assign a certain class or label to the input data, therefore the output is discrete and categorical [33].

Many algorithms have been implemented during the years for regression and classification purposes. Linear and polynomial regression are commonly used in the regression field, such as decision trees and random forests. These last two models are also implemented for classifications alongside with k-nearest neighbors, support vector machine and neural networks. An example of decision trees implementation for classification is shown in Figure 1.10. In chapter 3 the classification algorithms implemented in this project are described in detail.

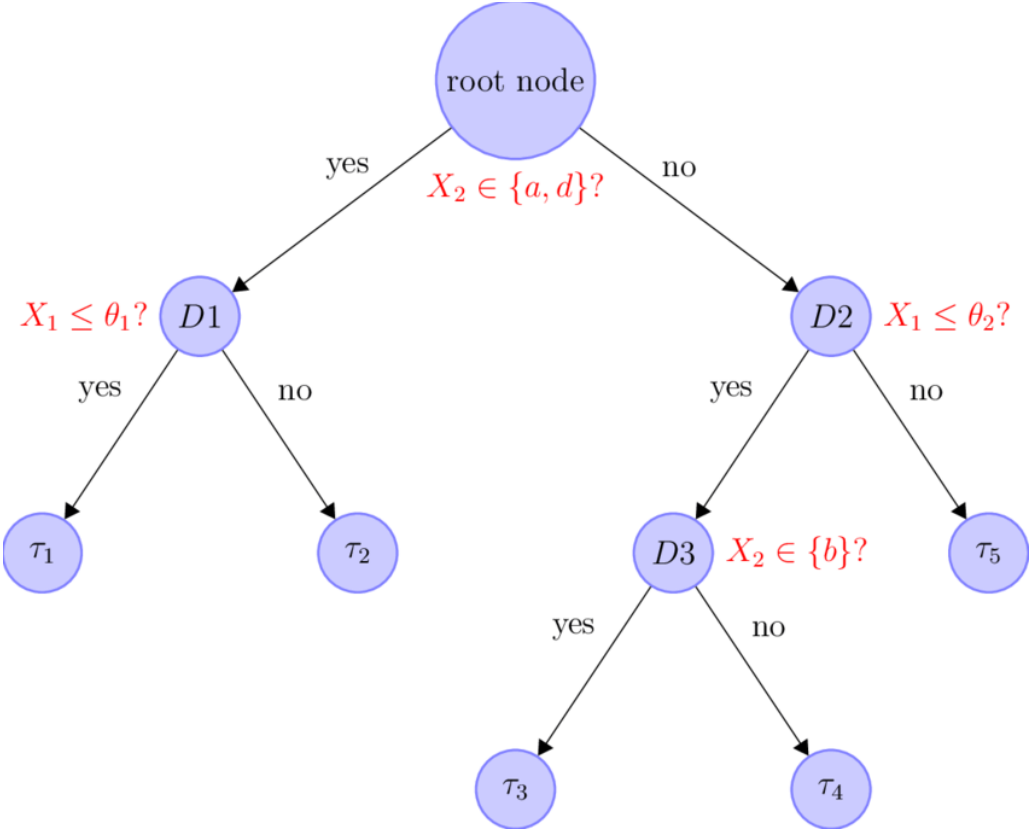


Figure 1.10: example of decision tree [36]

Many steps precede and follow the implementation of these models. Dimensionality reduction techniques are applied to the initial dataset in order to: select the main features among the provided ones, clean and pre-process the data that will be used as input. These and other preliminary data analysis steps are fundamental in order to obtain good results. Moreover, techniques for overfitting control are utilized such as cross validation and model score analysis.

Overfitting is one of the main concerns in the training of ML models: it describes a tendency of the models to follow not only the hidden patterns in training data, but also the outliers, or noise, of the data. In this case, the model essentially memorizes the training data rather than learning the underlying relations between features and target. Cross validation techniques aim to train and test the models with different portions of the initial dataset highlighting overfitting where present. Figure 1.11 shows how increasing the model complexity may lead to overfitting. Indeed, a too high complexity of the model could result in a very small training error, since the model follow deeply the intrinsic noise in the training data. On the counterpart, the model cannot generalize well on new unseen data, leading to a higher error on the testing dataset.

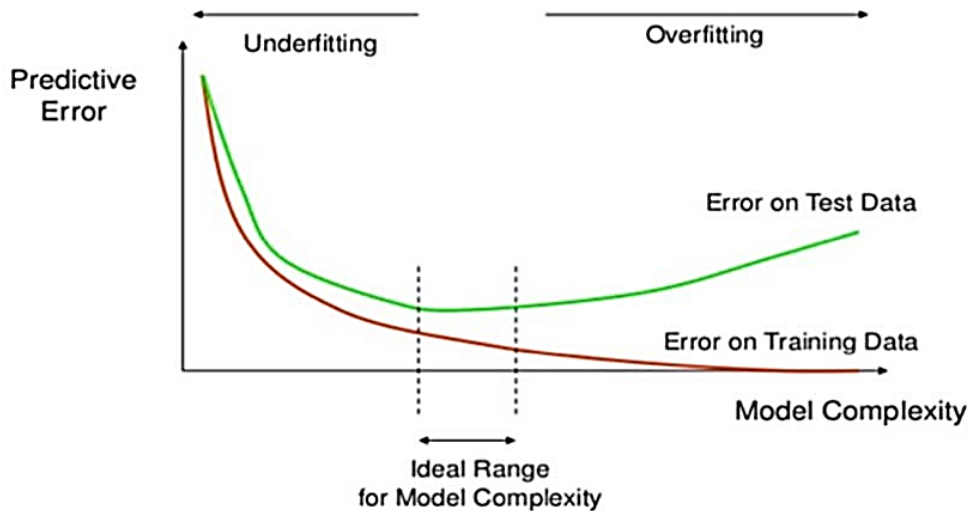


Figure 1.11: training and testing error related to model complexity [37]

1.3.4. Deep Learning

Another important subfield of ML that must be cited is Deep Learning (DL), which essentially utilizes so-called artificial neural networks formed by multiple layers of computation, with the purpose of performing different, and usually complex, tasks. Deep learning can be used in the same fields as of machine learning including classification, regression, clustering and generation. Many applications nowadays use these powerful tools, common examples are autonomous vehicles driving, computer vision and recommendation algorithms. The functioning of a neural network is based on a multi-layer network composed of nodes that interacts among them, as shown in Figure 1.12. The set up of bias parameters, activation functions and algorithms for minimization of errors and specific functions tasks is required. The complexity of the functions reproduced by the models could be elevated thanks to non-linearity of activation functions employed: these models are able to replicate highly nonlinear patterns found in the data. On the counterpart, DL models usually need a lot of data for training and, in addition, a lot of intrinsic parameters have to be defined. In these cases, the so-called “hyper-parameter tuning” may be challenging.

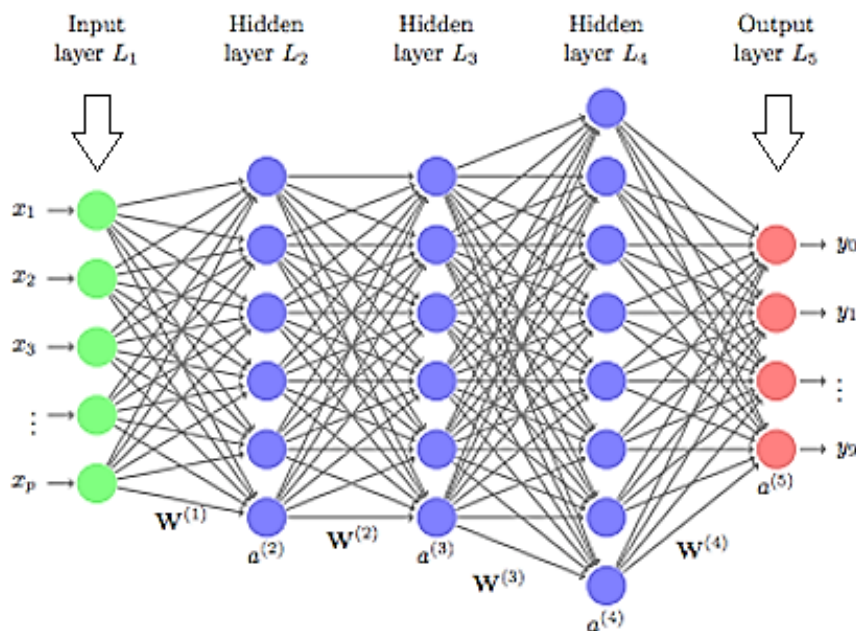


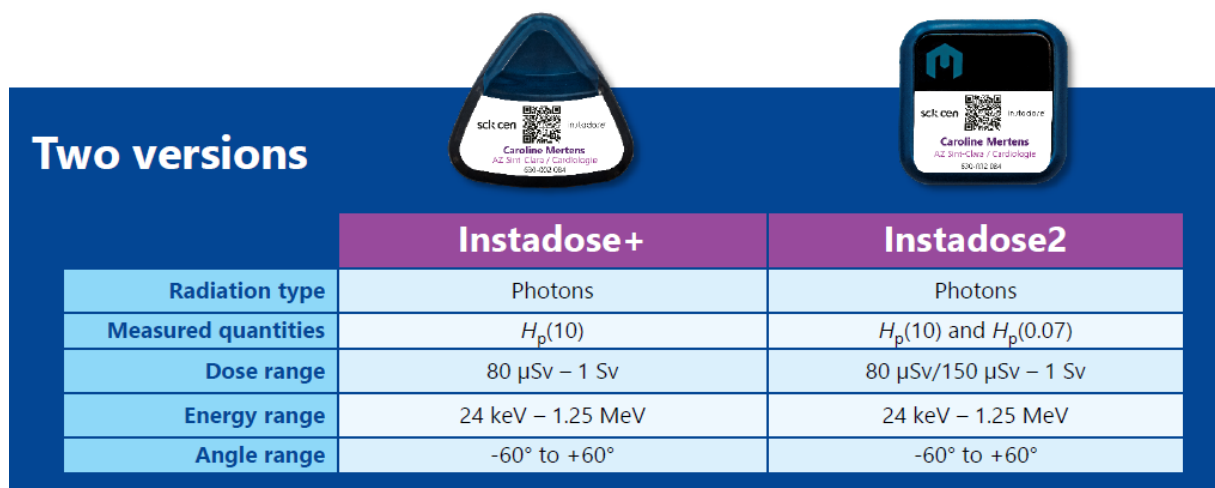
Figure 1.12: an illustration of a deep learning neural network [38]

2. Validation process and database configuration

In order to accomplish the aim of simplifying the validation process of the Instadose dosimeters performed at SCK CEN, a deep understanding of how the latter is performed is a primary requirement. This has allowed a further comprehension of the main features that are regularly taken into account during the validation by personal dosimetry experts. The following paragraphs contain a brief introduction of the data acquisition and of the manual validation process performed by personal dosimetry experts at the SCK CEN Belgian Nuclear Research Centre, along with the measures taken for the configuration of the database.

2.1. Instadose dosimeters

The validation process considered in this thesis concerns semi-electronic dosimeters. At the moment, SCK CEN provides two different kinds of Instadose dosimeters, both supplied by Mirion Technology. These are dosimeters based on the direct-ion-storage technology briefly described in paragraph 1.2.3. There are two types of dosimeters, called Instadose + and Instadose 2, they differ for some characteristics summarized in Figure 2.1. Both dosimeters have been taken into account for the improvement of the validation process considered in this thesis.



Two versions		
	Instadose+	Instadose2
Radiation type	Photons	Photons
Measured quantities	$H_p(10)$	$H_p(10)$ and $H_p(0.07)$
Dose range	80 μ Sv – 1 Sv	80 μ Sv/150 μ Sv – 1 Sv
Energy range	24 keV – 1.25 MeV	24 keV – 1.25 MeV
Angle range	-60° to +60°	-60° to +60°

Figure 2.1: differences between Instadose + and Instadose 2 dosimeters [39]

As previously mentioned, the readout of each dosimeter is carried out without recalling it to the laboratory, thanks to internal readout by the dosimeter and wireless data transmission by means of an acquisition system used by the customers. Three possibilities for the acquisition are offered: via the mobile app, via the InstaLink hotspot or via the InstaLink USB device. The data transfer is performed wirelessly by means of the Bluetooth Low Energy (BLE) technology integrated in the devices.

These dosimeters have been configured in order to perform a read once a week and at the end of each month, all labelled as “hard reads”. Hence, the monthly dose of a dosimeter can be validated. The customer might also decide to perform additional manual readings, which are also registered as “hard reads” on the Structured Query Language (SQL) database and on the Excel sheet created for the validation process, described in details in paragraph 2.2. Hard reads are the readings that are effectively validated. In addition, a reading is automatically performed once a day. These latter measures, labelled as “soft reads”, are considered less reliable with respect to the previously cited hard reads due to lack of temperature compensation, moreover they are also less accurate. However, they are useful to obtain information on temperatures and dose trends within one hard read and another. This distinction is fundamental for the creation of the database, as explained in paragraph 2.3.

As shown in Figure 2.1, two different values of personal dose equivalent are obtained by these dosimeters: Instadose + gives information regarding the before-called “deep dose” ($H_p(10)$), while

Instadose 2 provides information also regarding the so-called “shallow dose” ($H_p(0.07)$). For what concerns the validation process, since the deep dose value is considered as main reference, the shallow one has not been taken into account in the database. A unique database has been created for both the dosimeter types, containing information regarding only the deep dose, eventually evaluated by both dosimeter types.

As briefly explained in paragraph 0, Instadose dosimeters are passive dosimeters. Dose evaluation is therefore performed by means of a formula that takes into account the difference between the cumulative doses evaluated at two different times. For the evaluation of dose intake coming from the working activities, the natural background radiation, the fluctuation in the dosimeter reads and an individual calibration factor of the dosimeter have to be taken into account. Consequently, the formula used for the dose calculation is reported below:

$$Y = \frac{(\Delta U - U_{bg} - U_{intr}) \cdot f_1 \cdot f_2 \cdot f_3}{I} \quad (2.1)$$

In this formula, Y represent the personal dose equivalent ($H_p(10)$ or $H_p(0.07)$) in a specified time interval, evaluated in [mSv], ΔU represents the cumulative shallow or deep dose difference between two consecutive reads, while U_{bg} and U_{intr} are the dose amounts related to backgrounds – radiations and intrinsic background respectively – of the dosimeter. All these quantities are expressed in [Mirion mSv]. This peculiar unit of measure is used since the raw data read from the dosimeter have to be corrected with an individual calibration factor I , that is specific for each dosimeter and converts the raw data, measured in [Mirion mSv], to personal dose equivalent in [mSv]. In addition, some factors that take into account energetic dependence (f_1), angular dependence (f_2) and non-linearity (f_3) are considered. These latter factors are dimensionless.

The terms U_{bg} and U_{intr} can be reformulated as the product between the time interval within two consecutive reads (n), expressed in [days], and background doses related to natural background (B_{bg}) and intrinsic effects (B_{intr}). These last quantities are expressed in [mSv/day].

$$Y = \frac{\Delta U \cdot f_1 \cdot f_2 \cdot f_3}{I} - n \cdot (B_{bg} + B_{intr}) \quad (2.2)$$

The intrinsic background B_{intr} , addressed in this formula, refers to a variation, within dosimeter and time, in the observed background dose measured by the dosimeter. Although no studies related to this effect have been found and no information have been provided by Mirion Technology company, preliminary analyses conducted by SCK CEN link this effect to the variation in environment humidity, even if the dosimeters should be hermetically sealed. This evaluated factor is hereafter addressed as intrinsic background factor and changes for measures of $H_p(10)$ or $H_p(0.07)$. However, as reported previously, for the sake of this thesis just the deep dose one has been taken into account.

2.2. SCK CEN validation process

Dosimeters provided to customers by SCK CEN are validated monthly. The validation process is performed manually based on data collected by means of an SQL database. To ease the validation process to the experts, Microsoft Excel sheets containing relevant information and graphs are automatically generated by a Python script.

During the validation process, a class is assigned to each dosimeter based on the results of the collected measurements. Ten different classes have been identified and are reported below:

- A) The dosimeter performs perfectly and can be validated without further actions;
- B) The dosimeter does not perform well and should be kept under control during the next validations, no actions have to be taken for the moment;
- C) A value in the measurements, or several, need to be adjusted due to small problems (e.g. temperature jumps). Apart from that, the dosimeter performs well and no further actions have to be taken;

- D) One value, or several, in the measurements need to be adjusted. The dosimeter must be taken out of use;
- E) Values in the measurements can be approved but the intrinsic background factor, (B_{intr}) in formula 2.2, used for dose calculation has to be adjusted;
- F) The intrinsic background factor needs to be adjusted and values in the measurements has to be recalculated;
- G) The dosimeter can be validated but it has to be recalled to the SCK CEN laboratory;
- H) Fake readings have to be inserted before the dosimeter can be validated. This is usually related to lack of the end-month measurement: this reading is created thanks to an interpolation of previous and next readings;
- I) No validation can be performed usually due to data missing or necessity of next month's values for proper judgement;
- Z) The dosimeter performs well and a dose is detected.

Three different radiation protection experts are in charge of the validation process: the head of dosimetry lab, the head of the Radiation Protection Dosimetry and Calibration group, and head of Dosimetric and Calibration Services Unit at SCK-CEN. Usually, the customers are divided in three groups, based on companies, and each expert performs validations always on the same companies. By attending some examples of validation process performed, a subjective bias has been discovered concerning the above-mentioned classification. Indeed, since some classes do not have precise boundaries, different persons could validate the same dosimeter adopting different classes. Although this bias should be considered for the implementation of the ML models, it has not been possible to take it into account since the classification that is performed is also deeply linked with the customer group that is validated. For example, one expert is usually performing validation for hospitals: higher values of dose are expected by these kinds of customers with respect to other companies. This results in a higher number of dosimeters classified as Z with respect to dosimeters validated with the same class by other validators. For this reason, no information regarding who validated the measurements has been inserted in the training and test database.

2.2.1. SQL database

SQL is a programming language for storing and processing information in a relational database. In this kind of databases, information are stored in tabular form, with rows and columns representing different data attributes, and the various relations between the data values. Specific information can be retrieved by keywords used to manipulate the data stored inside the database [40]. Two main keywords used in the SCK CEN database are represented by "Group number" and "Dosimeter serial number". All the dosimeters that are provided to the same company are identified by the same group number, in addition, each dosimeter is identified by a unique serial number. A reading ID (IDentification) is also assigned to each collected measurement in order to uniquely labelled it: fake readings generated for validation class H have a read ID equal to -1.

This SQL database provides different views in order to simplify the search of information on a specific dosimeter or measurement. For the sake of validation, not all the information stored in the database are relevant: Table 2.1 provides a resume of all the parameters and values that could be useful for the validation process. These parameters have been taken into account for the creation of the database used for the ML models implementation.

<i>Information regarding the dosimeter type and customer</i>	Group number	Number of the group to which the dosimeter belongs
	Dosimeter serial number	Serial number of the dosimeter
	Dosimeter type ID	Dosimeter ID used to identify the type of dosimeter: Instadose + or Instadose 2

<i>Information regarding the reading</i>	Carrier information	First and last name of the person to which the dosimeter is assigned. A registration number and a spare ID are also stored
	Reporting level	Level above which a dose value is reported to the customer
	Wearing location	Wearing location of the dosimeter: it distinguishes among above, below or without lead apron
	Read ID	ID assigned to identify different reads
	Read type ID	ID that identifies the type of readings: hard or soft read
	Exposure and created date	Date on which the reading is performed and date at which the reading is stored in the database
	Deep and shallow raw dose values	Raw dose values evaluated in [Mirion mSv] for deep and shallow dose depending on dosimeter type
	Battery percentage	Battery percentage at the moment of the reading
	Temperature	Temperature measured at the moment of the reading
	Free space percentage	Free memory space in the device at the moment of the reading
	Flags for anomaly or initial read	Flags used to assess anomalies or first reads of new dosimeters
<i>Information regarding validation process</i>	Validation status	Flags used to determine the status of the validation
	Name of validator expert, validation message and remarks	Information on the validation process such as: name of the validator, comments and remarks on the validation
	Begin and end linking period	Information regarding the month to which the validation refers
	Calculated dose	Dose evaluated with formula 2.2 within two consecutive hard reads based on raw dose, correction factors and time interval
	Background and individual factors	Factor used for dose calculation
	Intrinsic factor for deep and shallow dose	Intrinsic factor used for dose evaluation
	Begin and end date	Dates at which the individual factor has been modified

Table 2.1: information retrieved by the SQL database for validation process

As before mentioned, these information have been used to generate Microsoft Excel sheets in order to assist the validation process.

2.2.2. Microsoft Excel sheets

The Microsoft Excel sheets used for the validation process are automatically created by a Python script developed within SCK CEN. One Excel file is created for each group number and it contains all the sheets that refer to that group number. One sheet is produced for each dosimeter evaluated and contains the main information resumed in Table 2.1 with different forms. An example of this Excel sheet is reported in Figure 2.2.

Wearer Number	Wearing Location	Start Date Time	End Date Time	Hp(0.07)	Hp(10)	Battery Percent	Free Space Percent	Temperature
Reserve 1	no apron	31-10-2023 23:30	03-11-2023 12:00		-0.018	100	105	22
Reserve 1	no apron	03-11-2023 12:00	03-11-2023 16:15		0.001	100	105	22
Reserve 1	no apron	03-11-2023 16:15	10-11-2023 12:00		0.015	100	104	22
Reserve 1	no apron	10-11-2023 12:00	17-11-2023 12:00		-0.011	100	104	22
Reserve 1	no apron	17-11-2023 12:00	24-11-2023 12:00		0.010	100	104	22
Reserve 1	no apron	24-11-2023 12:00	30-11-2023 23:30		-0.016	100	104	21
Reserve 1	no apron	30-11-2023 23:30	01-12-2023 12:00		-0.015	100	104	21
Reserve 1	no apron	01-12-2023 12:00	01-12-2023 13:34		-0.002	100	104	22

Wearer Number	Wearing Location	Start Date Time	End Date Time	Monthly Hp(0.07)	Monthly Hp(10)	Linking End Date	Transfer Date
Reserve 1	no apron	31-10-2023 23:30	03-11-2023 12:00		-0.019	31-12-9999 23:59	03-11-2023 16:16
Reserve 1	no apron	03-11-2023 12:00	03-11-2023 16:15		-0.019	31-12-9999 23:59	03-11-2023 16:16
Reserve 1	no apron	03-11-2023 16:15	10-11-2023 12:00		-0.019	31-12-9999 23:59	01-12-2023 13:00
Reserve 1	no apron	10-11-2023 12:00	17-11-2023 12:00		-0.019	31-12-9999 23:59	01-12-2023 13:00
Reserve 1	no apron	17-11-2023 12:00	24-11-2023 12:00		-0.019	31-12-9999 23:59	01-12-2023 13:00
Reserve 1	no apron	24-11-2023 12:00	30-11-2023 23:30		-0.019	31-12-9999 23:59	01-12-2023 13:00
Reserve 1	no apron	30-11-2023 23:30	01-12-2023 12:00		-0.017	31-12-9999 23:59	01-12-2023 13:00
Reserve 1	no apron	01-12-2023 12:00	01-12-2023 13:34		-0.017	31-12-9999 23:59	01-12-2023 13:35

Figure 2.2: table with hard reads in Microsoft Excel sheet

Figure 2.2 reports how the hard reads are shown inside the sheet, the table has been divided in two pictures, for better representation. In Figure 2.2, the cell highlighted with a bold border shows the value of the dose taken in the entire month, and it marks also the last measurement of the month, therefore the last two rows in the figure have not been taken into account in the current validation process and will be validated only once all readings from that month are available. Along with this table, a box containing main information on the dosimeter status and calibration are reported, as shown in Figure 2.3.

```

Serial number = 23043433
Linking = 111-06, Reserve 1
Natural background= 0.002 mSv/day
Individual Factor (IF) Hp(10) = 1.0099
Background factor (BG) Hp(10) = -0.00103
Last calibration= 2023-05-10 00:00:00

Status = Sent

Errors: final reading status

Last transferred status= 2023-12-01 13:34:04

Battery voltage = 100%
Lowest temperature = 20 °C
Highest temperature = 22 °C
Natural background Hp (10) soft reads= 0.0014 mSv/day
Natural background Hp (10) hard reads= 0.0008 mSv/day
Accumulated dose = 2.389 mSv
Internal clock: Ok: 2023-12-01 13:34:04 < 2023-12-01 13:35:23

```

Figure 2.3: main information regarding dosimeter in Microsoft Excel sheet

However, the most significant part of the Excel sheet is constituted by the graphs, generated from all the measurements collected in the previous months, related to the dosimeter that has been taken into account. Different graphs are generated in order to retrieve different information, where all the measurements from the previous months are used along with the newly acquired ones.

Figure 2.4 shows the total dose calculated by the dosimeter: as expected, the curve grows with time. The red dots in the graph represent the previously called hard reads, while the blue ones represent the soft reads.

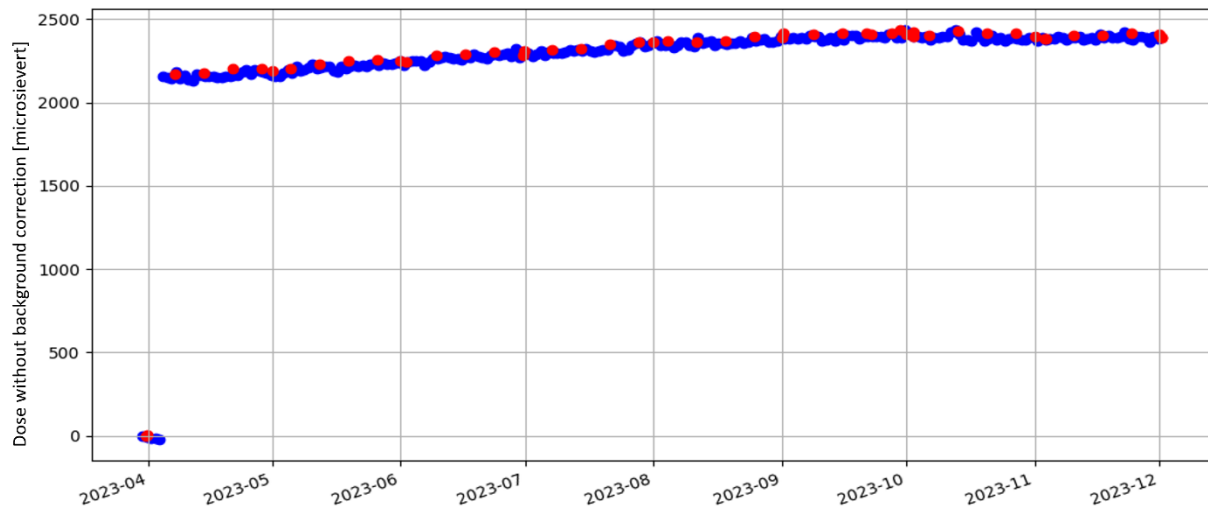


Figure 2.4: real dose graph with no filters for dosimeter validation

Figure 2.5 shows, instead, filtered values in order to have a better comprehension of the dose received by the customers. In details, this graph is evaluated subtracting the background contribution to the dose evaluation and the values of dose that exceed 1 mSv . In this way, the dose from the customers related to their working activities is clearly visible. In the graph reported in Figure 2.5, other peculiarities can be noticed: along with the dose values, the temperature is plotted (small light blue dots). Temperature values are fundamental to validate the dosimeter since very high or low temperatures influence deeply the dosimeter behaviour, leading to odd measurements. If these are encountered, it is important to recognise if they are related to temperature jumps or not, in order to carefully classify the dosimeter. Two vertical lines are also drawn: the green line takes into account changes in the intrinsic background effect of the dosimeter, while the purple one outlines changes in the natural background.

Before delivering the dosimeter to the customers, those are left for a certain period of time in a controlled environment in the SCK CEN facility, where the natural background is known and equal to $2.2\ \mu\text{Sv/day}$. During this period the intrinsic effect mentioned in chapter 2.1 is estimated, subtracting the natural dose from the measured one. This effect is taken into account in the dose calculation by means of the B_{intr} factor reported in formula 2.2. For this reason, a green line is present after the first months of measurements in Figure 2.5. For what concerns instead the purple line, the natural background has changed from $2.2\ \mu\text{Sv/day}$ to $2.0\ \mu\text{Sv/day}$. This has been done because the dosimeter was sent to a customer where the natural background dose rate was estimated to be $2.0\ \mu\text{Sv/day}$, this estimation is based on a statistical analysis of the dosimeters used by this customer.

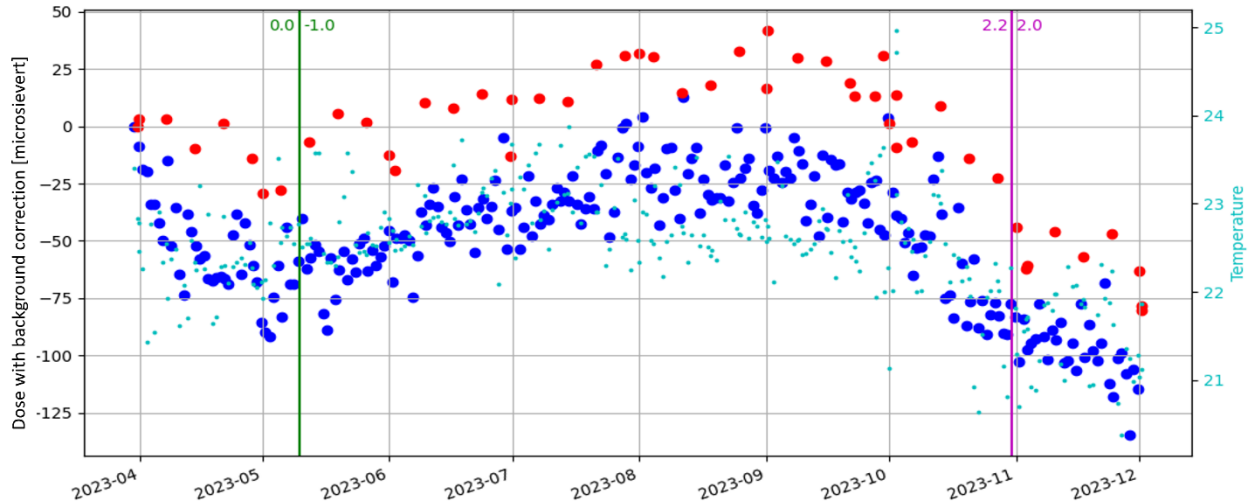


Figure 2.5: dose graph filtered from outliers and background contribution

Moreover, the graph in Figure 2.6 is used to check the necessity of intrinsic background recalibration. In order to do so, the dose readings with values higher than 150 μSv are deleted: in this graph just the dose related to background radiation should appear.

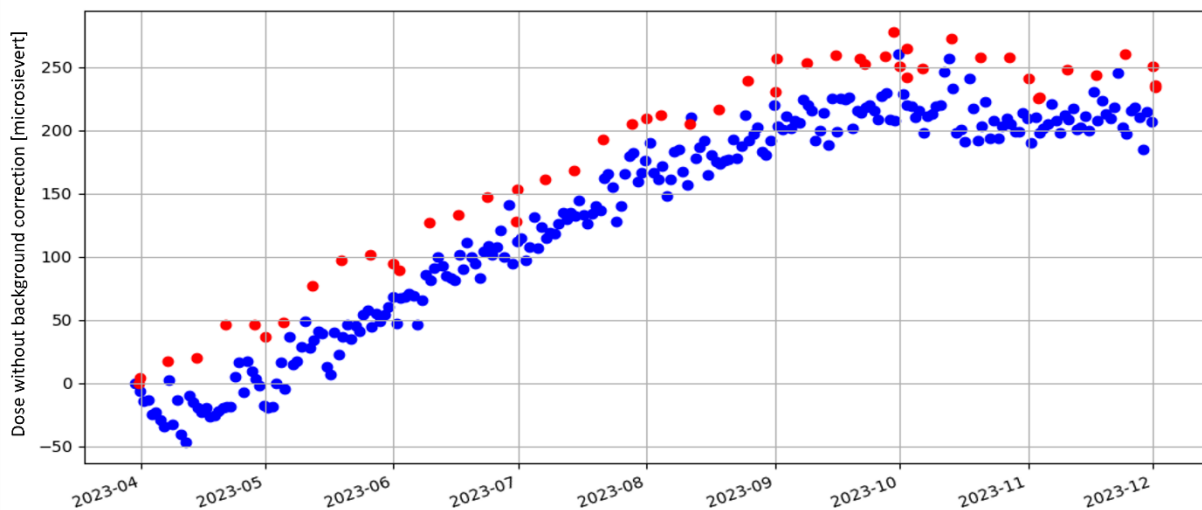


Figure 2.6: dose graph filtered with jumps higher than 150 μSv

As above-mentioned, these graphs contain the main information on which the validators base the classification process. In order to classify a dosimeter as well-performing (class A), the graphs reported above should present clear trends with small variations and just few outliers, mainly related to temperature jumps. The graph that takes into account background contribution (Figure 2.6) should present a constant behavior, high slopes must be corrected changing the intrinsic background factor with class E or F. During this process, data contained in Figure 2.2 and Figure 2.3 are used along with these graphs to choose the most suitable validation class.

2.3. Configuration of final database for ML implementation

With the purpose of model implementation, a labelled database has to be provided as input. This database has been later on divided in training and testing datasets. The first have been used to fit the parameters and hyperparameters of the model, which will be later validated thanks to the testing dataset. Since the supervised classification models that have been later implemented are non-parametric models, predictions of the classes will deeply depend on the data that are used for the training. Indeed, non-

parametric models do not make assumptions on the structure of the data: this topic have been explained in detail in chapter 3. In this way, there is a high dependency of the model parameters on the data provided for model training, which means that the training dataset must be representative of all the possibilities that can be encountered in the testing, and later on during the application of the model to new unseen data.

In addition, the necessity of structuring the dataset in order to give more importance to hard reads with respect to soft ones emerged from the validation process analysed in the previous paragraphs. In the following paragraphs the measures taken to build the final database have been summarised.

2.3.1. Data rearranging and grouping

As mentioned previously, the models that have been implemented are supervised models, meaning that the database on which they are trained and tested contains previous validated data for which the classes have been already identified. The steps of the validation process described before have been developed during the last years, and they are more or less unchanged since the beginning of 2023. For this reason, it has been chosen to consider just the validations performed from 1st January 2023 onwards.

In order to give a smoother reading, the several databases created as intermediate steps in order to configure the final one, have been named progressively (*A*, *B*, and so on). First of all, during the validation process the assigned class is reported just on the Excel sheets, for this reason, the first step that has been performed concern the retrieving of this information from all the Excel sheets in a small database, hereafter recalled database *A*.

This *A* database contains just the serial number of the dosimeters, the assigned validation class, the validation date and the name of the Excel sheet, as reference. In this first process, validations that do not have a class, or have class information located in different parts of the sheet, have been discarded. The database that has been used to obtain the results reported in chapter 4, contains measurements up to the end of November 2023. In the eleven months going from January to November 2023, the total amount of validations retrieved from these Excel sheets were $\sim 11\,980$. Afterwards, the information collected inside this first database have to be merged with the information retrieved by the SQL database, resumed in Table 2.1.

It is important to clarify that the validation process is usually performed one month after the measurements: a validation of a dosimeter performed around mid-December refers to all the measurements collected from the previous validation session up to the end of November. Usually (but not always) the dosimeters are validated monthly, so these measurements refer just to one month. Dosimeter serial numbers and validation dates have been used in order to link hard and soft reads contained in the SQL database with the related validation classes, based on what has been explained above.

Since the validation process is performed only on hard reads, no dose evaluation has been automatically included inside the SQL database for soft reads. However, since they have to be inserted in the final database used for the implementation of the ML models, it is required to evaluate the dose for these soft reads. Formula 2.2 has been used for this purpose: factors and dose values retrieved by the SQL database have been employed. The time interval that has been considered for this dose calculation has been evaluated taking into account the closest hard or soft read. Factors f_1 , f_2 and f_3 contained in formula 2.2 have been all assumed to 1 for this calculation. At the end of the implementation process, it has been found out that one of these factors was incorrect. Indeed, the energy correction factor used for dose evaluation is equal to 0.86 for $H_p(10)$ evaluation in Instadose +, and equal to 0.98 for $H_p(10)$ evaluation in Instadose 2. However, multiplying all the values by the same factor, even if not correct, do not influence the implementation of the ML model. An implementation with the corrected dataset has been performed for some of the models leading to comparable results. Because of lack of time, the overall models have not been re-trained on the corrected dataset, based on the test previously performed. In order to give an order of magnitude concerning the size of the database up to this point, the merging of readings in SQL database with the $\sim 11\,980$ validations, contained in database *A*, produce a new database, named *B*, containing $\sim 484\,250$ rows and 33 columns: each row refers to a soft or hard

measurement while each column -or feature- refers to an information on that measurement. Nevertheless, database *B* is not clean yet and other measures have to be implemented in order to generate the final one.

As previously pointed out, hard reads are more significant with respect to soft ones. In database *B*, the major part of the rows contains soft reads, since they are daily reads, therefore, once the data are feed to the ML models, a too high importance is given to soft reads. In order to overcome this problem, a data grouping has been performed: all the soft reads collected within two consecutive hard reads have been grouped in a single row. In this way, information regarding validation class, dosimeter and factors for dose evaluation remain unchanged (since these remain fixed within two hard reads). On the other side, information regarding the readings, such as temperature, dose, battery percentage and free memory have been added to the database. In particular, the latter have been added taking into account the mean, the maximum and the minimum values, as well as the standard deviation and the variance of the selected soft reads. In addition, it has been decided to add a column containing information on the number of soft reads that has been grouped during this process: examples are reported in the below list under the name of “Class size”.

After rearranging columns order, a database containing $\sim 138\ 000$ rows and 37 columns has been obtained, named database *C*. In data analysis and ML implementation, the columns of a database are usually called features, as cited in paragraph 131.3.1. An exhaustive list shows the cited features contained in the final database:

- Validation class: class assigned to the measurement after the validation;
- Class size: number of measurements taken into account in that row. This value is equal to 1 for what concern hard reads (1 row is related to just 1 measurement), and equal to the number of soft reads grouped together, for what concern soft reads;
- Dosimeter serial number;
- Validator: a number that uniquely addresses who validated the reading;
- Group number: the group number to which the dosimeter belongs;
- $H_p(10)$ mean value: value of deep dose for what concern hard reads while, for soft reads, contains the mean value of the grouped readings;
- $H_p(10)$ maximum value: value of deep dose for what concern hard reads while, for soft reads, contains the maximum value of the grouped readings;
- $H_p(10)$ minimum value: value of deep dose for what concern hard reads while, for soft reads, contains the minimum value of the grouped readings;
- $H_p(10)$ standard deviation value: value of deep dose for what concern hard reads while, for soft reads, contains the standard deviation evaluated with the grouped readings;
- $H_p(10)$ variance value: value of deep dose for what concern hard reads while, for soft reads, contains the variance evaluated within the grouped readings;
- Background factor: background factor retrieved from the SQL database view for hard read dose evaluation;
- Intrinsic factor: B_{intr} value, retrieved by the SQL database, cited in formula 2.2 for the deep dose evaluation;
- Individual factor: individual factor retrieved by the SQL database used to calculate the deep dose;
- Wearing location: number that uniquely define the location of the dosimeter (above, below or without lead apron);
- Read type ID: as described in Table 2.1, it contains information on the type of reading collected in the rows: hard or soft reads;
- Battery percentage features: as for the $H_p(10)$ dose, also in this case there are 5 different features taking into account mean, maximum, minimum, standard deviation and variance;
- Temperature features: as for the $H_p(10)$ dose and battery percentage also in this case 5 different features are present;

- Free space percentage features: as for the previously cited information also in this case there are 5 different features;
- Anomaly flag: flag taken by the SQL database that identifies anomalous readings;
- Initial reads flag: flag taken by the SQL database that identifies first readings of the dosimeters;
- Begin month: number of the month retrieved by the Begin Linking Period (BLP) field in the SQL database cited in Table 2.1. It provides information on the months to which the validation process refers;
- End month: number of the month retrieved by the End Linking Period (ELP) field in the SQL database cited in Table 2.1. It provides information on the months to which the validation process refers, it usually is equal to the “Begin month” cited above, since validation is performed monthly;
- Exposure day: number of the day retrieved from the Exposure Date (ED) field in the SQL database;
- Exposure month: number of the month retrieved from the ED field in the SQL database;
- Grouping key: number that has been previously added in order to allow the grouping of the soft reads;

All the steps cited up to now, as well as the one contained in the next paragraphs, have been performed by Python scripts by means of Pandas library [41].

Simplifications and cleaning

Outliers and anomalies can appear during the process, while working with real data. Since the created database *C* is extensive, rows presenting missing information are present and have been deleted for the sake of ML models implementation. This cleaning process is fundamental in order to create a suitable dataset for ML algorithms libraries.

The cleaned dataset does not contain measurements validated as class I: this has been done since, for the sake of automatization of the validation process, class I can be simply detected without the necessity of a prediction from a ML model. In addition, in order to be correctly classified as validated in the SQL database, two flags have been assigned to each measurement, therefore a correct validation process results in both flags equal to 0: measurements that do not belong to this group have not been taken into account.

During the grouping process some soft measurements have been deleted since there is a long gap between two subsequent hard reads: around 700 measurements have been cancelled for this reason.

Updating process

With the aim of automatization using Machine Learning implementation, an updating process of the database has been developed. Up to now it has been used to insert newly validated measurements inside the previously cited database *C*. For this reason, it is performed at the beginning of the month, when the validation process for the entire previous month is finished. It follows the steps described in the previous paragraphs, taking into account just the readings acquired in the last month.

For what concern the future implementation of these ML models on newly acquired data, the updating process can be used to retrieve also not validated measurements. Afterwards, developed models will be used to provide a prediction of the validation class easing the validation process.

2.3.2. Dimensionality reduction

A good practise for the pre-processing of datasets, for the subsequent implementation in ML algorithms, is the implementation of dimensionality reduction techniques, on the grouped database *C*, previously described. These techniques are fundamental in order to reduce the run time of the models and to lower the complexity of the data presented to the algorithms. Indeed, the aim is to decrease the number of features present in the database without deleting information that could be important for the interpretation of the data structure by ML models. The process of selecting the features that are essential

for describing the database is called Feature Selection, and in this project it has been performed by means of correlation matrices. A correlation matrix shows the correlations that are present between two features that belong to the same dataset. Indeed, each entry of the matrix shows a coefficient that expresses the strength of the correlation between two intersecting features. The term “correlation” refers to a “*mutual relationship or connection between two or more things*” [42], features in this case. In this way, it’s also possible to quantify which feature mostly influence the validation class. Two types of correlation have been studied in this analysis: Pearson’s and Spearman’s correlation.

“The Pearson correlation coefficient is a measure of the linear association between two variables. It quantifies the strength and direction of the relationship between these variables. The coefficient ranges on values between -1 and $+1$: a value of ± 1 indicates a perfect linear relationship between the two variables. Negative values simply indicate that, as one variable increases, the other decreases. A value of 0 indicates no linear relationship, while coefficients that differ from 0 but are not equal to ± 1 indicate a linear relationship, although not a perfect one” [43].

A first analysis of correlation between features has been performed by means of a Python library named Sweetviz [44]. Thanks to this library, the Pearson’s correlation matrix, reported in Figure 2.7, was retrieved. However, since some features in the database are categorical, Pearson’s correlation cannot be applied on all the features. Associations that involve categorical features, such as the validation class, are evaluated based on uncertainty coefficients and correlation ratios with a value that goes from 0 to 1. These kinds of correlation are represented by squares in the matrix, reported in Figure 2.7: bigger is the square, stronger is the correlation. Circles represent numerical features associations evaluated by means of Pearson’s correlation coefficients, which goes from -1 to 1 , as reported in the colour-bar on the right side of Figure 2.7. For what concerns the uncertainty coefficients, used to evaluate categorical associations, they are asymmetrical. This means that each values on a specific row represents how much the row taken into consideration provides information on the column (i.e. feature) that is intersecting. In particular, the first column on the left of Figure 2.7, informs about which features (on the intersecting rows) give more information on the validation class.

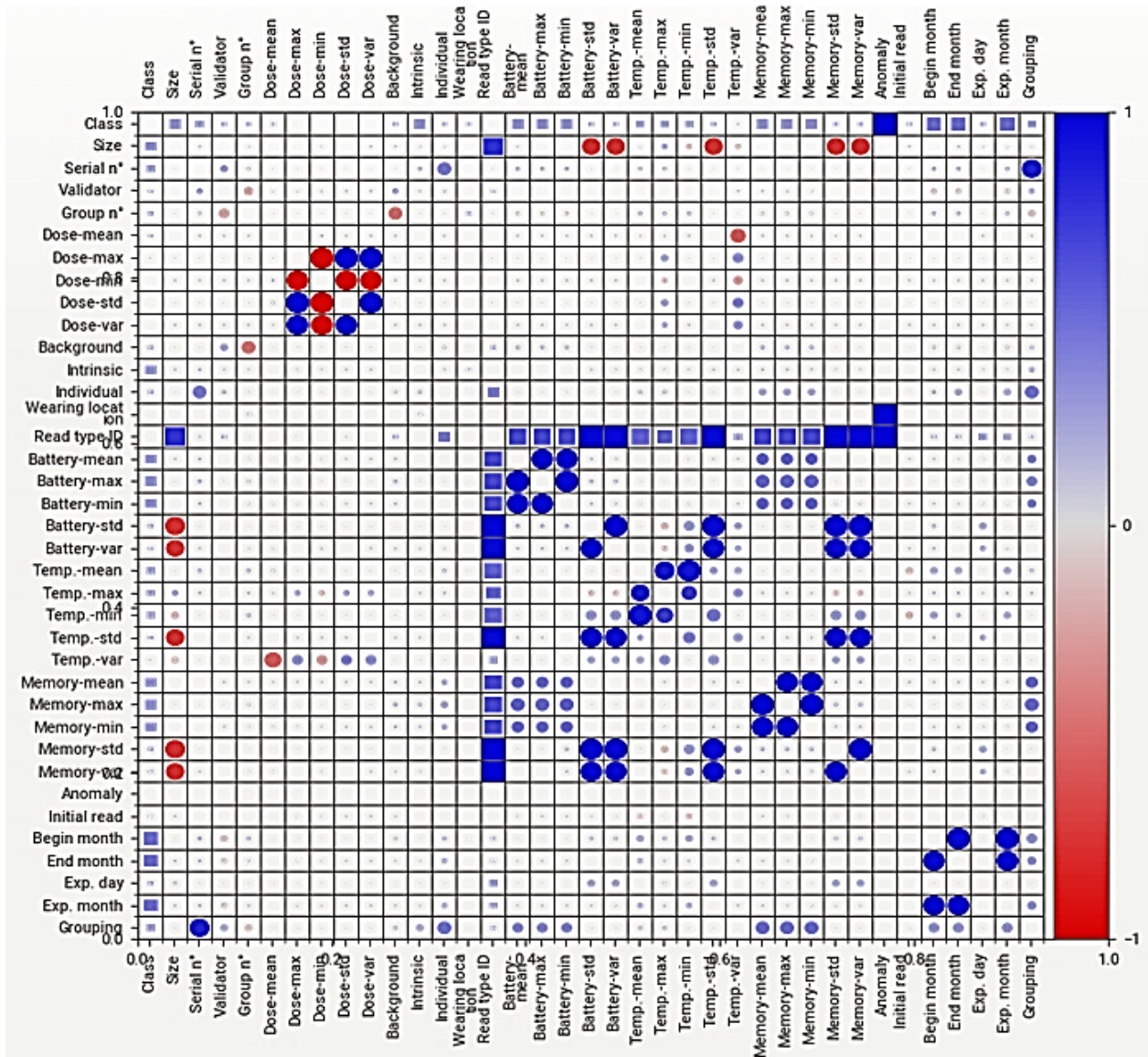


Figure 2.7: correlation matrix for grouped database evaluated with Sweetviz library

Concerning Spearman’s correlation, it has been proposed as a “measure of the strength of an association between two variables”. “It assesses how well an arbitrary monotonic function can describe a relationship between two variables, without making any assumptions about the frequency distribution of the variables” and, unlike Pearson’s, it does not imply that the relationship has to be linear [45]. The correlation matrix reported in Figure 2.8 contains Spearman’s correlation coefficients, and it must be noted that, in order to create it, the categorical features have been temporarily deleted.

Results from these correlation matrixes have been compared with the information used in the manual validation process in order to decide which features can be deleted without losing essential information. During this process, a conservative approach has been adopted in order to delete just the features that surely do not affect the validation class prediction.

Eventually, features that have low correlation coefficients reported in Figure 2.7 and Figure 2.8 have been removed. They are listed below:

- Validator: deleted because the subjective bias described in paragraph 2.2 cannot be implemented in the ML models. This information is somehow distorted for the validation process;
- Wearing location: this feature shows very low correlation coefficients and does not significantly influence the validation class;
- Anomaly and Initial read flags: looking at the data contained in these columns, just few measurements in the database present a value different than 0 (just 1 measurement for the Anomaly flag and 26 for the Initial read flag). This make it useless for a ML implementation since this information is not present for a sufficient amount of measurements;
- Exposure day: correlation coefficients for this feature are low. In addition, the day of the exposition should not influence the measurement, while the month can be related to temperature and humidity variations that influence the performances of the dosimeter;
- Grouping key: this feature has been inserted during the creation of the database and does not regard the measurements, therefore it has been deleted;

Although the features have been selected via feature selection, some of them are barely used during the manual validation process. Other features present low correlation coefficients, such as the background, intrinsic and individual factor. Even though these coefficients are small, intrinsic factor is important for the model, since it is one of the main reasons for dose recalculation. Therefore, they have been kept in the database for this first ML implementation. In future updates of this study, they might be carefully removed looking at the model performance. Correlation matrices retrieved after the Feature selection process are finally reported in Figure 2.9 and Figure 2.10.

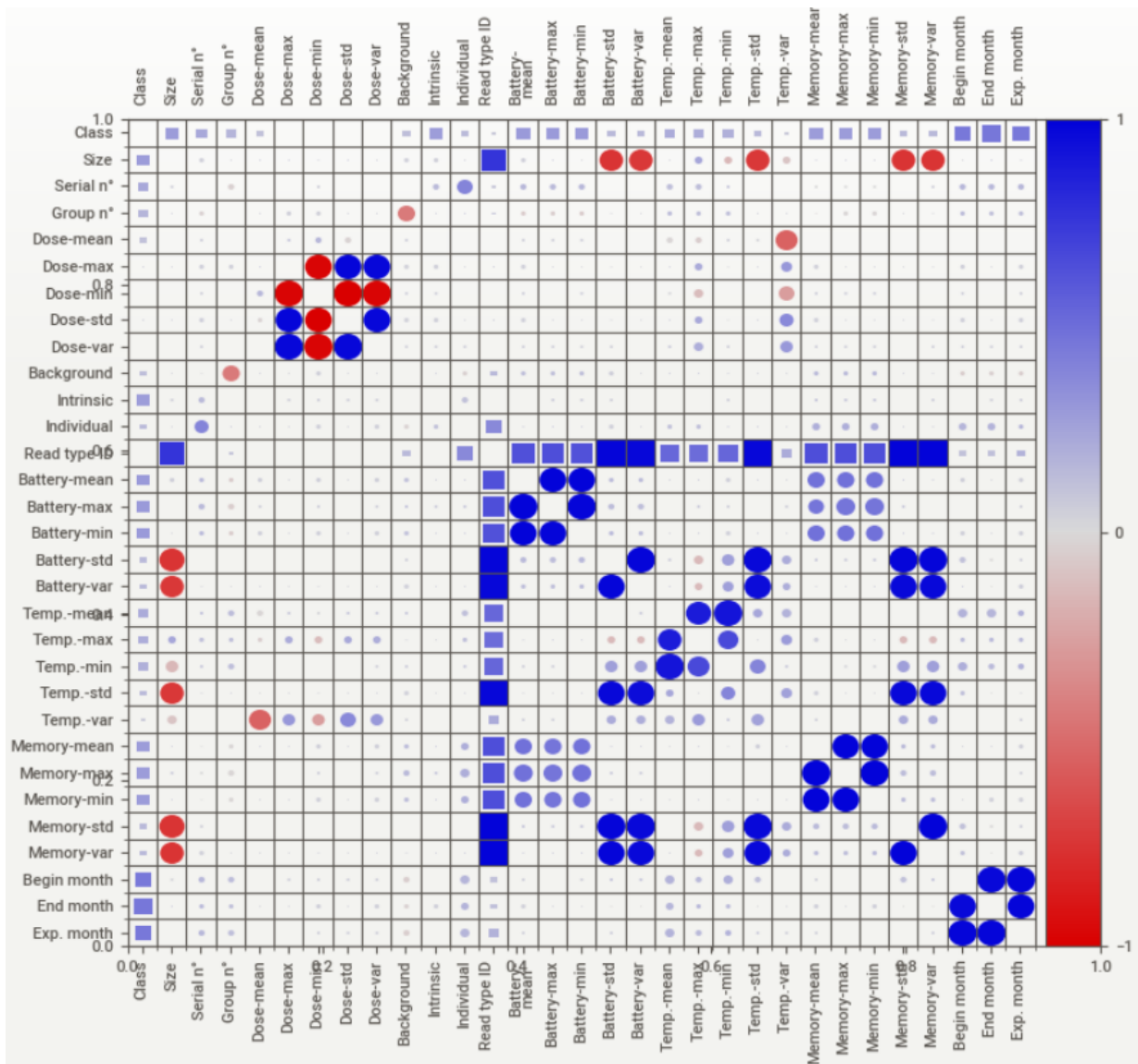


Figure 2.9: correlation matrix for grouped database evaluated by Sweetviz library after feature selection

2.3.3. Pre-processing of the database

In order to implement ML models, the training and testing datasets have to be perfectly clean, without presence of NaN (Not a Number) variables that can cause errors during the implementation. With this final cleaning process, database *C* analysed before, decreased from ~ 137 000 rows to ~ 130 000 (5.65 % of the measurements have been deleted for lack of information).

The main concept behind this ML implementation is to classify the well-performing dosimeters from all the other classes: this led to a binary classification between class *A* and the sum of all the other classes, grouped in a newly formed class *L*. This process is also important for class balancing: in order to obtain reliable results from supervised classification models, the training dataset should contain comparable classes sizes. As shown in Figure 2.11, class *A* counts the main contribution. It has to be noticed that, as discussed previously, there are no measurements validated as class *I* since they have been deleted.

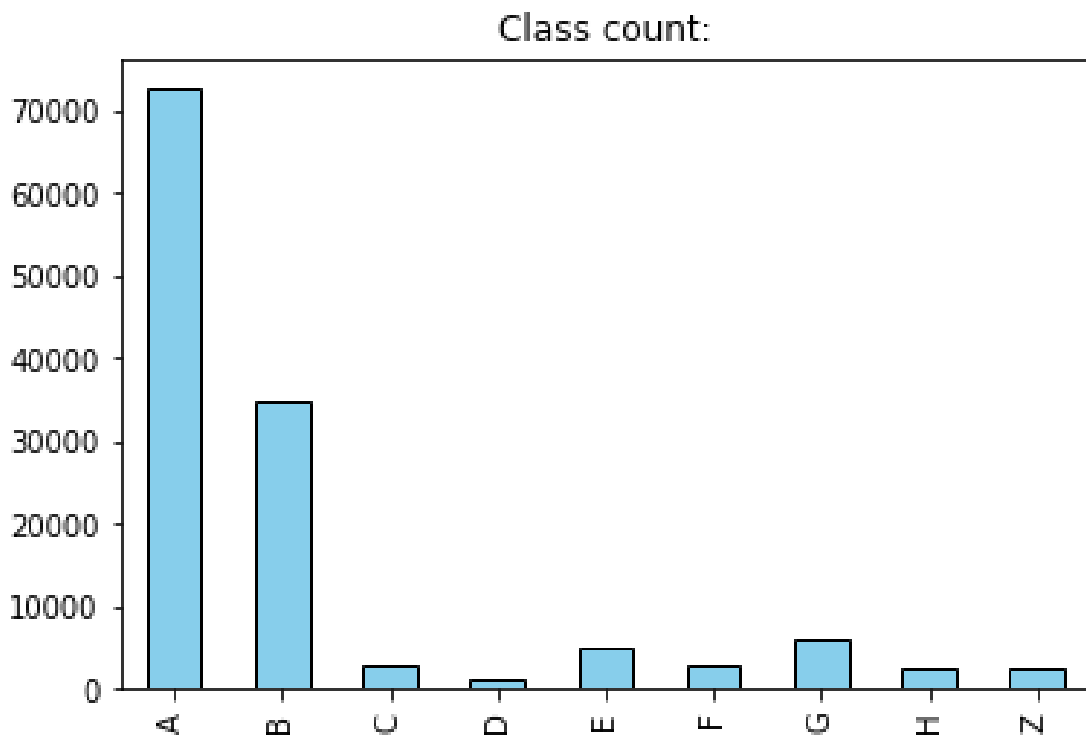


Figure 2.11: histogram with classes distribution in grouped database

Figure 2.11 reports the classes contained in the database after the cleaning process. In this database the measurements validated with class *A* represent the 55.96 %. After the grouping process, distribution of the classes in the database is reported in Figure 2.12.

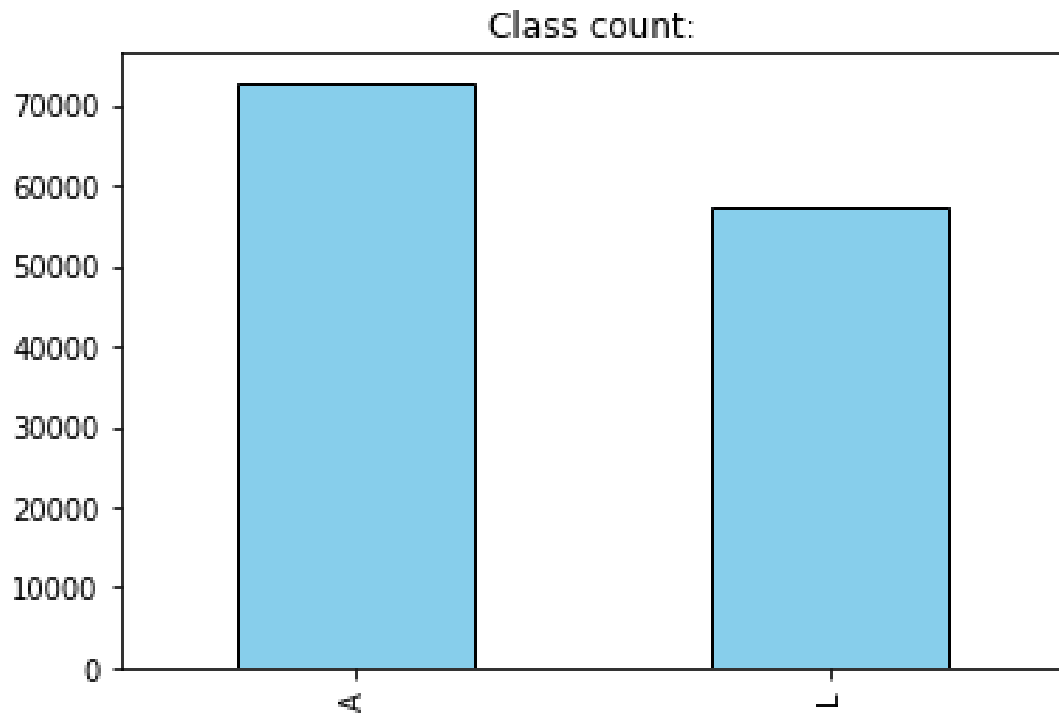


Figure 2.12: histogram with classes distribution in grouped database after class balancing process

A last step has to be performed before the ML models implementation: the database provided needs to be standardized. This process has been performed by means of the StandardScaler class contained in the sklearn.preprocessing module provided by scikit-learn [46]. This tool standardizes the values of each feature by setting the mean to zero and scaling to unit variance. This standardization process does not involve the target feature, that in this case is represented by the validation class. The terminology “target feature” refers to the feature that will be the output of the ML classification models.

3. Machine Learning algorithms

For the purpose of automatizing the validation process, supervised classification models have been implemented. As briefly explained in paragraph 1.3, supervised ML models use labelled inputs in order to predict outputs. Indeed, they are based on an inductive process for which the algorithm “learns” independently how to perform, in this case, the validation process without giving instructions on how to perform it: the model based its learning process on the relationship present among the labelled data provided in the training dataset. For this study, the models need to perform a classification process: the requested output is categorical, and it is represented by the validation classes.

Various supervised classification algorithms are available and deeply described in literature. In the following paragraphs, the models used in this study are briefly described: K-Nearest Neighbours (KNN), Decision Tree (DT), Random Forest (RF) and Neural-Networks (NN). *“Often, the methods used for classification first predict the probability that the observation belongs to each of the categories of a qualitative variable”*, in this sense these models also behave like regression methods [33]. For this reason, all the cited models can be also used in regression problems.

The models that are used in this study are all defined as nonparametric models, this means that they do not make assumptions about the form or structure of the function that maps inputs to outputs and so also about the distribution of the data. This is fundamental for the implementation of data that do not follow a normal distribution, as in this case.

These models have been implemented following a procedure developed for providing the best possible results. The main steps of this procedure are reported below:

- Creation of the dataset: assumptions and measures taken for the creation of the dataset has been extensively explained in paragraph 2.3;
- Splitting the dataset in training and testing ones: the dataset has been split in two. One dataset, containing the majority of the measurements (usually ~ 80 %) is used for the training of the models. The remaining part (~ 20 %) is used for testing. The mentioned percentages are commonly used in literature, they are related to best possible results and have been evaluated empirically and then demonstrated [47];
- Hyperparameter selection: any ML model needs a certain amount of hyperparameters. Hyperparameters of a model, are those that remain unchanged during training, therefore, being in another level of abstraction, are called “hyper”. Indeed, an hyperparameter is defined as a parameter that is assessed before the learning process of the model begins [48]. Values for these parameters have been chosen a priori and directly affect how accurate a model can be. For the sake of simplification, just the most important hyperparameters for each model are going to be described. Since it is unfeasible to know a priori which are the best values for each hyperparameter, a selection of the most appropriate or suggested values has been carried on. In this way, all the possible combinations between these values of the selected hyperparameters have been generated;
- Implementation of the models: once the combinations have been defined, a model for each combination has been implemented on the training and testing datasets defined before. This process allows to identify the best combinations based on the results of these models;
- Hyperparameter tuning: looking at the implementation results, the combinations of hyperparameters that results in the highest performances have been identified. These will serve to build a much smaller subset of combinations that can be implemented easily on new incoming data. This step is essential in order to accelerate the actual validation process: it is unfeasible for the dosimetry experts to run every month all the possible combinations of hyperparameters described above, therefore a sorting of the hyperparameters which perform better has been introduced, to provide just the latter to the final user for a faster running of the models;
- Cross-validation process: this is an heuristic method used to assess the model’s performance on new unseen data, in order to understand how well the model can deal with generalization. It is used to point out overfitting phenomena assuring that the model is not learning the noise inside

the data. In cross-validation, a fixed number of folds (or partitions) of the dataset are created, the analyses are then performed on each fold averaging the overall error estimation. This process involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set, in order to have each fold used as testing once. The cross validation process has been performed only on the subset of the hyperparameter combinations retrieved in the previous step. This has been made necessary due to lack of time during the final stages.

As a reminder, to understand properly why cross-validation is performed, it is important to give a definition of overfitting. *“Overfitting is a fundamental issue in supervised machine learning which prevents us from perfectly generalizing the models to well fit observed data on training data, as well as unseen data on testing set”* [49]. A good generalization of the model is fundamental to have ML models that provide reliable results on new and unseen data. Various techniques are performed to prevent overfitting, like early-stopping, network-reduction and others.

It has to be noticed that, since the scikit-learn Python libraries have been used for the implementation of these ML models, the terminology used in the following chapters refers to this particular library.

3.1. *K-Nearest Neighbours*

This is the first and most straightforward method that has been implemented in this study. It is called K-nearest neighbours because it classifies a new data point, which belongs to the testing dataset, based on the K nearest data points contained in the training one.

A small digression should be added regarding classifiers: the most accurate one that can be adopted is the Bayes classifier [33]. The latter is a statistical classifier that has the smallest probability of misclassification among all classifiers based on the same set of features. It is based on the Bayes theorem allowing the inversion of the conditional probabilities. However, in order to implement such algorithm, the conditional distributions of the features, given the class labels, are needed, and it cannot be applicable in this case. *“Many approaches attempt to estimate these conditional distributions and then classify a given observation to the class with the highest estimated probability. One such method is the KNN classifier”* [33]. Indeed, given the positive integer K and a test observation x_0 , this classifier first identifies the K points in the training data that are closest to x_0 , represented by N_0 . Consequently, it estimates the conditional probability for a class j as the fraction of points contained in N_0 whose response values equal j . Finally, KNN categorises the test observation x_0 as the class with the largest conditional probability [33].

However, for high-dimensional parameter spaces, this method becomes less effective due to the so-called “curse of dimensionality” [50]. High-dimensional parameter space refer to a mathematical space that is used to model datasets with many attributes. Each attribute -or feature- of the dataset represents a dimension in this space.

3.1.1. **Hyperparameter definition**

As previously mentioned, in order to implement these supervised classification models various hyperparameters have to be defined. Table 3.1 reports the main hyperparameters that have been tuned for this study, concerning KNN models.

Hyperparameter	Values
K	[2: 15]
Weight	“Uniform”, “Distance”
Algorithms	“Brute”, “K-D tree”, “Ball tree”, “Auto”,
Metrics	“Euclidean”, “Manhattan”, “Chebyshev”,

Table 3.1: main hyperparameters values for KNN classifier

K value

This hyperparameter refers to the number of nearest points that have been taken into account for the classification of new measurements. An optimal values of K should be found for each application. Generally, a small value of K leads to a model that follows the variations in the data too closely, being more prone to overfitting possibility. On the other side, a larger value of K tends to suppress the effects of statistical noise but makes the classification boundaries less distinct. Effects of these two distinct cases are shown in Figure 3.1: the picture shows the change in the decision boundary with respect to the value of K. The purple dashed line represents the decision boundary obtained by the Bayes classifier.

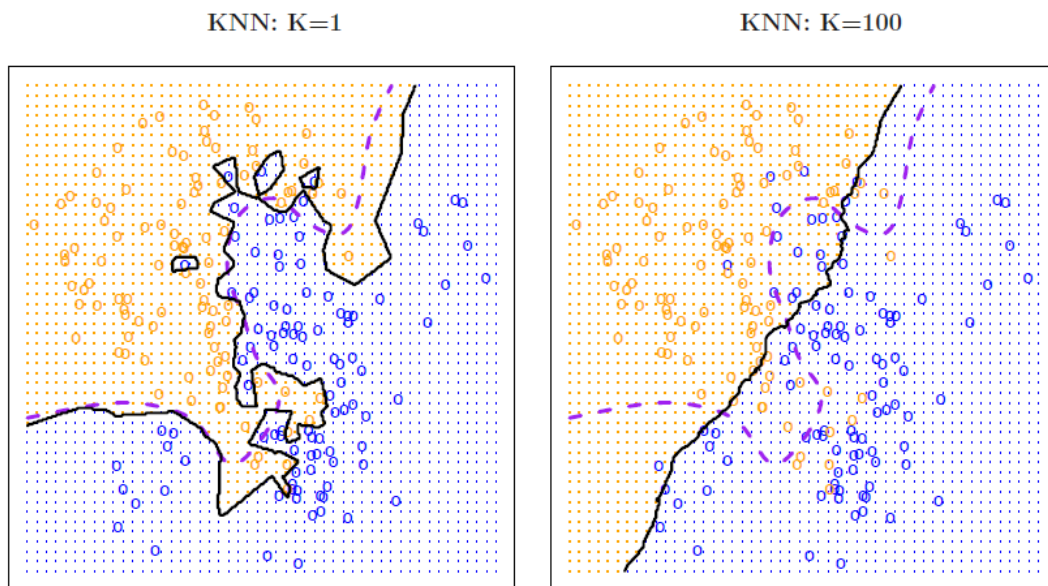


Figure 3.1: comparison of KNN decision boundaries with $K = 1$ and $K = 100$ [32]

As reported in Table 3.1, the values assumed for this hyperparameter range from 2 to 15. It is important to remember that the optimal choice of the K value is highly data dependent.

Weight function

The K points selected could have different weights in the classification process for the new data, based on the function selected in this hyperparameter. With a “Uniform” weight function all the K points in the neighbourhood have the same importance for the prediction. Instead, a “Distance” weight function assigns a weight to these points based on the inverse of the distance between these and the new point: higher is the distance, smaller is the importance assigned to that training point.

Algorithm

This hyperparameter selects which algorithm has to be used for the selection of the nearest neighbours. Different possibilities are present in the scikit-learn library and all of them have been included, they are listed below:

- “Brute”: this algorithm evaluates the distance from the point x_0 in the naivest way and it is very competitive for small data samples. However, as the number of samples grows, the brute-force approach quickly becomes infeasible, as reported in scikit-learn user guide [46];
- “K-Dimensional (KD) tree”: this algorithm is tree-based and is implemented in order to overcome the computational inefficiencies of the “Brute-force” algorithm reducing the required number of distance calculations. The KD recursively partitions the parameters space, along the data axes, into regions in which data points are filled. The construction of the tree is very fast, however, if the dimension D of the dataset grows fast ($D > 20$), the algorithm becomes inefficient due to the previously cited “curse of dimensionality”;
- “Ball tree”: it was developed to address the inefficiencies of KD trees in higher dimensions. Ball trees partition data in a series of nesting hyper-spheres, while previously mentioned KD trees use cartesian axes. In this case, data are divided into nodes defined by radius and centroid position;
- “Auto”: tries to implement the most appropriate algorithm based on the measurements used to fit the model;

Metrics

This hyperparameter selects the type of metric that has been used to evaluate the distance between points inside the model. All the various possibilities have been selected:

- “Euclidean”: refers to the “ordinary” straight-line distance between two points in a Euclidean space (L_2) norm;
- “Manhattan”: also known as the L_1 norm. It is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates;
- “Chebyshev”: it is a metric defined on a real coordinate space, where the distance between two points is the greatest of their differences along any coordinate dimension. It is also known as the maximum metric or L_∞ .

3.2. *Decision Trees*

A decision tree is a model of supervised machine learning used for both classification and regression tasks. It presents a flowchart-like structure where each internal node denotes a test on an attribute of the data, each branch represents an outcome of the test, and each leaf node (terminal node) holds an outcome (validation class in this case). It starts from the root, like an upside-down tree, and branches off to represent various realizations, as reported schematically in Figure 3.2. In this study, DT has been used fundamentally to recreate the validation process subdividing it into simple and elementary choices, based on the data provided as input.

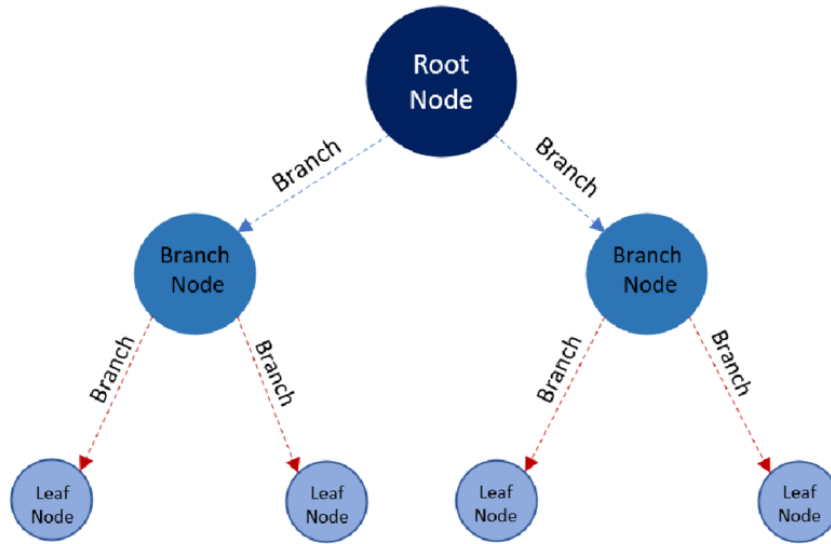


Figure 3.2: example of a basic decision tree [48]

DTs present several advantages: small data preparation is required, and thanks to their simplicity, they are requiring less time to train with respect to more complex models, such as NNs. The main disadvantage is instead represented by the risk of producing over-complex trees that do not manage to generalize the data structure, leading to overfitting. An example of this issue is shown in Figure 3.3, where the reported decision tree presents an exaggerated number of branches with respect to the validation process that has to be reproduced. Indeed, this was one of the first attempts of DT implementation. For this reason, the pruning process of DT models is fundamental for overfitting control.

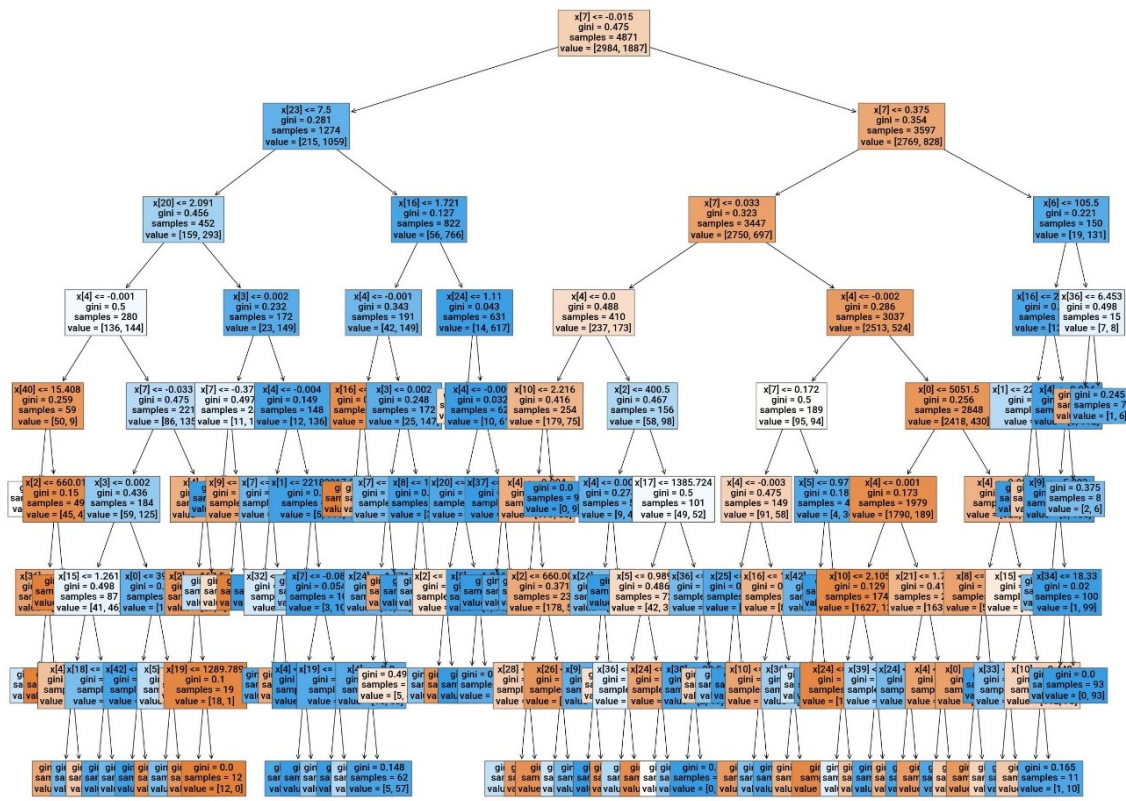


Figure 3.3: decision tree implemented in early stages during this thesis project

3.2.1. Hyperparameter definition

As done for the KNN model, the main hyperparameters taken into account for DT models are briefly described in this paragraph. The correspondent values for these hyperparameters have been summarised in Table 3.2. Explanations for the following hyperparameters are mainly retrieved by the scikit-learn guide [46].

<i>Hyperparameter</i>	<i>Values</i>
<i>Tree depth</i>	None – (15, 20, 25, 30, 40)
<i>Maximum leaf nodes</i>	None – (2, 5, 10, 30, 50)
<i>Criterion</i>	“Gini”, “Entropy”
<i>Minimum samples per leaf</i>	[1: 7]
<i>Max features</i>	“None”, “sqrt”, “log2” or 0.90

Table 3.2: main hyperparameters values for DT classifiers

Tree depth

It identifies the maximum number of layers contained in the generated tree. If this hyperparameter is set to “None”, nodes are expanded until all leaves are pure or until all leaves contain a number of samples that is lower than the minimum number of samples set for a split (other hyperparameter that has to be defined). A leaf node is considered pure if all the training samples that reach it belong to the same class.

Maximum leaf nodes

This hyperparameter addresses the maximum number of leaf nodes present in the model. Limiting the number of leaf nodes effectively controls the complexity of the model: smaller number of leaf nodes can simplify the model and prevent overfitting, while a larger number allows the model to fit the data more closely. When this hyperparameter is set to “None”, no limitation concerning the number of leaf nodes is posed.

If the maximum leaf node is different with respect to “None”, the tree is constructed in the so-called best-first fashion. In this case, at each step, the algorithm selects the split that maximally reduces the impurity among all the possible splits. The sequence by which the tree expands is determined by the quality of the splits, i.e., not all nodes at the current level of the tree need to be split before a node at the next level is split.

Criterion

It defines the function used to measure the quality of a split. Criteria that have been selected in this study are “Gini” and “Entropy”. Concerning the “Gini” criterion, it uses the Gini index, also known as Gini impurity, as a measure to determine the quality of a split. It calculates the probability that a randomly chosen instance would be incorrectly classified. While, the entropy, used in the “entropy” criterion, is a measure of disorder or uncertainty in a dataset. It helps to build an appropriate tree by selecting the best splitter and can be defined as a measure of the purity of the sub-split.

Minimum samples per leaf

This hyperparameter sets the minimum number of samples that are required to be at a leaf node. In particular, a split point, at any depth, would be considered just if it leaves at least a number of training samples in each of the left and right branches that is equal to this parameter. The tuning process of this

hyperparameter affects the smoothness of the model, and for the sake of this study, this hyperparameter has been set to vary from the minimum possible value (equal to 1) to 7.

Max features

It is used to choose the number of features that has to be taken into account for the so-called “best split”. In case of “None”, all the features are used for the split. While, the square root of the number of features are used (in case of “sqrt”) or the logarithm with base equal to 2 of it is used (in case of “log2”). If a float number is given, as done for 0.90, that fraction of the features is used for the choice.

3.3. Random forest

Random forest methods are categorized as ensemble methods, since they merge predictions from multiple base estimators constructed using a specific learning algorithm. This fusion aims to enhance generalizability and robustness, compared to the use of a single estimator. Essentially, RF comprises multiple DT models, having each tree constructed from a sample drawn with replacement from the training dataset. Consequently, RF algorithms are often referred to as perturb-and-combine techniques.

The splitting of each node, carried out during tree construction, implies an exhaustive search for the best split based on feature values. The number of features considered in this process can range from all input features to a randomly selected subset of a specified size. The feature randomness introduced in this process results in a random subset of features, which minimizes correlation among the DTs contained in the RF model. In particular, individual DTs typically demonstrate high variance and a tendency to overfit, as discussed in paragraph 3.2. The definition of variance, in this context, is the extent to which a model prediction would change if it was trained on a different set of data [33]. High variance can lead to overfitting and individual DTs are known to have high variance and a tendency to overfit. However, when they are combined into a RF model, the ensemble can help to reduce this variance. This is because the RF model averages the predictions of all the individual DTs, which can balance out their individual errors and lead to a more robust and accurate prediction. Thus, RF effectively mitigates variance by amalgamating diverse trees [51].

Concerning classification purposes, the class that obtains the highest number of votes from all the DTs in the forest is chosen as the final prediction, as reported in Figure 3.4. This model has been adopted thank to its simplicity, versatility, and robustness. Indeed, RF can handle large datasets with high dimensionality assessing which variables are important in the classification process.

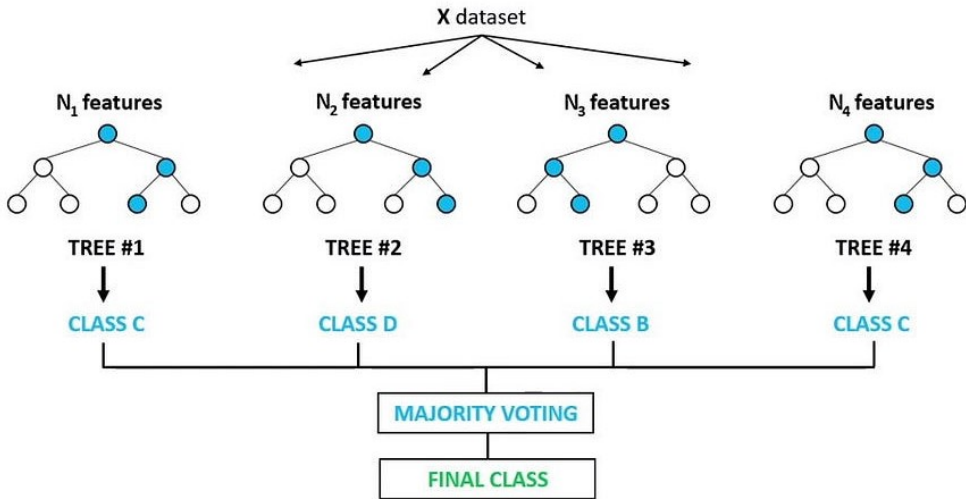


Figure 3.4: example of a basic random forest classifier [50]

3.3.1. Hyperparameters definition

RF models require the definition of many hyperparameters. Table 3.3 contains the most important ones with the related chosen values.

<i>Hyperparameter</i>	<i>Values</i>
<i>Number of estimators</i>	(10, 50, 100, 150, 200)
<i>Tree Depth</i>	None – (5, 10, 25)
<i>Bootstrap</i>	“True” or “False”
<i>Warm start</i>	“True” or “False”

Table 3.3: main hyperparameters values for RF classifiers

Other hyperparameters referred to the DTs created inside each RF model have been selected. However, not all these parameters are reported in Table 3.3, to avoid unnecessary repetitions.

Number of estimators

It can be considered the most important hyperparameter since it establishes the number of DTs that are constructed for each RF model. In order to have a representative study, this value has been set to range from models that present few estimators (number of estimators equal to 10) to models constituted by a significant number of trees (number of estimators equal to 200). Clearly, this hyperparameter deeply influences the training time of the model.

Tree depth

As for the DT models, it establishes the depth of the estimators. Values for this parameter have been decreased in number with respect to the one assumed for DT models, contained in Table 3.2. This was necessary in order to obtain a feasible number of hyperparameter combinations.

Bootstrap

This hyperparameter establishes if bootstrap sampling is used for the construction of the trees. This terminology refers to a method used in statistics that involves the sampling process of a dataset with replacement. This means that some measurements may be used multiple times in a single bootstrap sample, while others may not be used at all. If the hyperparameter is set to “False”, the whole training dataset is used to build each tree.

Warm start

When it is set to “True”, the algorithm reuses the solution of the previous call to fit and add more estimators to the ensemble. This means that the model continues the training where it stopped, allowing the addition of more estimators incrementally. If it is set to “False” a whole new forest is trained from scratch. This parameter is mainly useful in parameter search, as the one that has been performed in this thesis; for this reason, it has been added in this discussion.

3.4. Neural Networks

In the framework of ML models, neural networks represent a crucial method. With respect to the previously mentioned models, NN contains a higher level of complexity concerning the architecture of the model, as well as the underlying mathematical concepts. As cited in paragraph 1.3.4, NN constitute the so-called deep learning subfield of AI. NN try to mimic the human brain synapses, it is indeed based on an interconnection of several nodes, called neurons. The latter are schematic representations of a

weighted sum of the incoming connections from other neurons. The neurons are organized in layers in such a way that, for the simplest case, every neuron belonging to the n layer receives a connection from the neurons in the $n - 1$ layer. The output of a neuron is then supplied to the neurons of the $n + 1$ layer. To introduce non-linearity, a transformation in the output, known as activation function can be applied.

Artificial NNs are based on a supervised learning algorithm called Multi-Layer Perceptron (MLP). A brief schematization of a single layer MLP is shown in Figure 3.5. The first layer of the network, on the left, is called input layer and is followed by a certain number of so-called hidden layers (in the case of Figure 3.5, just one), connecting the input and the output ones.

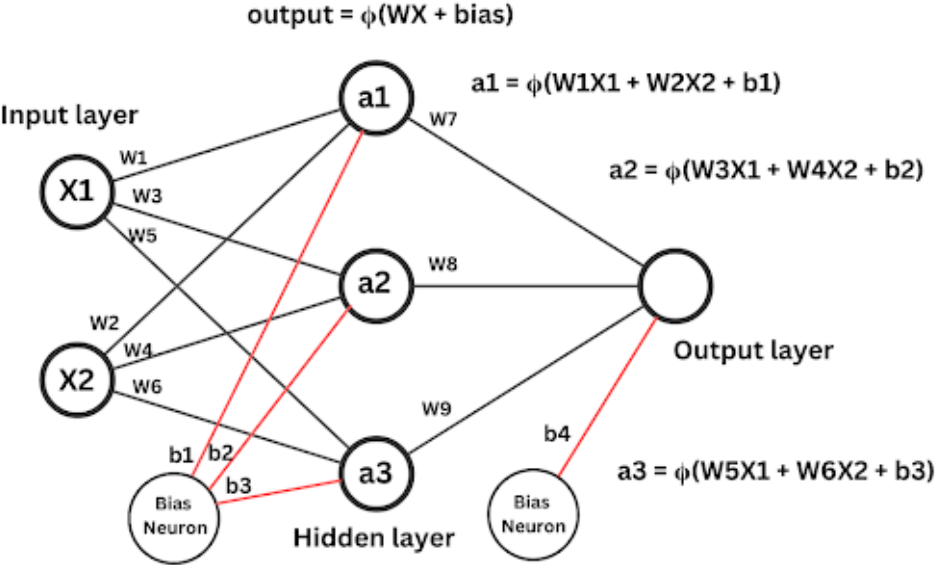


Figure 3.5: schematization of a single layer MLP [51]

Figure 3.5 shows how the information inside the networks are transported from one layer to the next one. Within a neuron, the inputs from preceding layer neurons or initial features, in the case of the input layer, are aggregated considering their respective weights (denoted as w_i in the illustration). Additionally, a bias term b_i must be included into this calculation, as depicted. An activation function (represented as ϕ in the figure) is applied to this weighted aggregation within each node, generating the output of the neuron. This output is then utilized in subsequent layers or as part of the final output. The activation function serves the crucial role of introducing non-linearity within two adjacent layers in the neural network architecture. Eventually, the only difference between a single layer and a multi-layer (MLP) structure lays on the number of hidden layers that are present. A representation of a multi-layer network is shown in Figure 3.6.

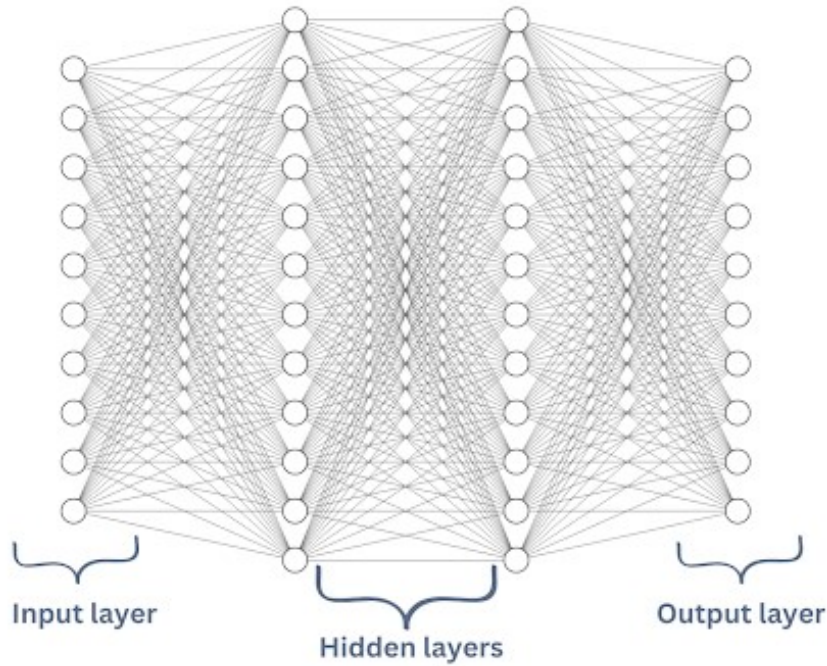


Figure 3.6: schematization of a multi-layer MLP [51]

Back Propagation (BP) algorithms, along with forward propagation ones are used to adjust the weights inside the NN during the training process. *“Idea behind BP algorithm is quite simple, output of NN is evaluated against desired output. If results are not satisfactory, connection (weights) between layers are modified and process is repeated again and again until error is small enough.”* [52]. On the contrary, during forward propagation, outputs are calculated and compared with desired outputs.

NNs use backpropagation as a learning algorithm to compute a gradient descent, which is an optimization algorithm that guides the user to the maximum or minimum of a function. In a ML context, the gradient descent helps the system minimize the gap between desired outputs and achieved system outputs adjusting the weight values for various inputs. More specifically, a gradient descent algorithm uses a gradual process to provide information on how a network’s parameters need to be adjusted to reduce the disparity between the desired and achieved outputs. An evaluation metric called loss function guides this process [53].

Several hyperparameters are requested for the implementation of a NN, which represents the main drawback concerning this model: the complexity of the structure. On the counterpart, all these parameters permit a very detailed particularization of the model allowing an efficient adaptation to highly non-linear problems. Indeed, different kind of activation function can be used, as described in paragraph 3.4.1, allowing the reproduction of complex sub-structures within the data.

3.4.1. Hyperparameters definition

As previously mentioned, many hyperparameters have to be defined for a NN implementation. Since it is not feasible to describe all the parameters in this discussion, only the main ones are reported in Table 3.4 and therefore described.

Hyperparameter	Values
Number of layers and neurons	(200 × 1), (100 × 2), (100 × 1), (75 × 2), (50 × 4), (50 × 3), (50 × 2), (50 × 1), (33 × 3), (25 × 8), (25 × 4), (25 × 2), (17 × 3)
Activation functions	“ReLU”, “ <i>tanh</i> ”, “Logistic”
Solver for weight optimization	“Adam”, “Sgd”
Scheduling strategy	“Constant”, “Adaptive”, “Invscaling”
Batch size	(32, 64, 128, 256)

Table 3.4: main hyperparameters values for NN classifiers

Number of layers and neurons

These are the first and most important hyperparameters: together, they establish the NN architecture defining the number of layers and the number of neurons contained in each layer. As it can be inferred from the terminology, the number of layers establishes the number of hidden layers in the NN: while each layer can be constituted by a different number of neurons. Concerning this study, NNs with the same number of neurons in all the hidden layers have been implemented, for this reason a compact notation is adopted in this document. In Table 3.4, the notation used denotes:

(*number of nodes* × *number of hidden layers*)

For example, (50 × 4) means that the NN is constituted by four layers and each one is formed by 50 neurons. Concerning scikit-learn notation [46], this type of NN is described by (50, 50, 50, 50).

Activation functions

Activation functions are included in order to take into account non-linearity between one layer and the subsequent one. Different kind of activation functions are usually implemented in NN models, the most common ones, concerning classification problems, are used in this study. They are explained below and the correspondent graphs are shown in Figure 3.7:

- “Logistic” (or sigmoid) function: it outputs values that lays inside the range [0:1]. The sigmoid function is especially useful in models that require a probability as output, since it is the same function used in logistic regression to convert a linear function into probabilities [33];
- Rectified Linear Unit (“ReLU”): it is the state of art for activation function. It is defined as $\max(0, x)$: if the input x is positive, it will output the input directly, otherwise, it will output zero. The main advantage of using the ReLU function, over other activation functions, lays on the capability to not activate all the neurons at the same time: converting the output of a neuron to zero, in case of negative input, does not activate the neuron [54]. This activation function does not present the vanishing gradient problem peculiar of models that use sigmoid activation function;
- Hyperbolic tangent (“*tanh*”): it outputs values that lays inside the range [-1:1]. It became preferred over the logistic function as it gave better performance for multi-layer NNs. Indeed, the tanh function is often recommended because it is a zero-centred function, meaning its output values range from -1 to 1. This can make learning for the next layers easier because the gradients are not restricted to move in a certain direction [55]. In fact, gradients are computed for back propagation algorithms. The highest gradient for the tanh function is found at 0, which means that: small changes in the input around 0 will result in larger changes in the output. This

peculiarity avoid one of the main problems that concern logistic function: 0 is the lowest point of the gradient, meaning changes in the input around 0 will result in very small changes in the output [56];

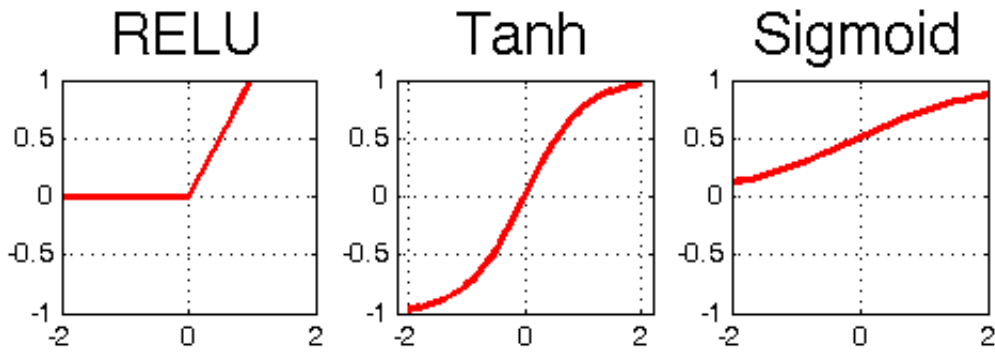


Figure 3.7: activation functions shape [57]

Solver

As previously mentioned, weights (depicted as w_i in Figure 3.5) have to be defined for the implementation of a NN model. The goal of the solver described by this hyperparameter is to optimize the choice of these weights minimizing the loss function. The term “loss function” refers to a function that quantifies the difference between the expected outcome and the outcome produced by the ML model.

For the purpose of this study two of the most used optimizers have been adopted: “Adam” optimizer and “Stochastic Gradient Descent (SGD)”. Adam typically provides good results, in terms of training time and validation score, on relatively large datasets (as the one used in this case, described in paragraph 2.3). Indeed, “Adam” solver has been created updating the “SGD” solver. The difference lays on the adjustment of the learning rate that is performed individually on each weight, in order to not generate unresponsive neurons during this process [57].

Scheduling strategy

This hyperparameter is called “learning rate” in the scikit-learn implementation and it links to the scheduler used for weights updating. This scheduler assesses how the learning rate, used for weights update, varies during the NN training process. Indeed, weights are updated at each iteration of the learning process, aiming to reduce the loss function. The formula used for the weight update is reported below:

$$w_i^* = w_i - \alpha * \left(\frac{\partial Error}{\partial w_i} \right) \quad (3.1)$$

In formula 3.1, weight w_i is updated thanks to a learning rate (α) and to the gradient of the loss function. The rate at which the NN weights are updated can change during the learning process varying the learning rate, affecting how quickly a NN model learns a problem.

Concerning scikit-learn implementation, three different schedulers can be selected. The learning rate α can be kept constant (“constant” scheduler), it can gradually decrease following an exponential law (“invscaling” one) or it can adapt to the problem based on the learning process of the NN (“adaptive” one).

Batch size

It defines how many values are seen by the NN prior to weights update process, constituted by one cycle of forward and back propagation. “While smaller batch sizes generally converge in fewer training epochs, larger batch sizes offer more parallelism and hence better computational efficiency” [58]. Indeed, big values for this batch size results in more data points provided to the model previous to the

update process, this make the training process faster, but the weights are updated on an average of the gradients computed on this large batch. On the contrary, a small batch size results in a slower training process. However, because there is more variability in the small batches, the path towards the minimum of the loss function can be noisier. For these reasons, different values for this batch size affect indicators such as overall training time, training time per epoch, and quality of the model.

4. Results

This chapter is devoted to the analysis of the results obtained implementing the ML methods described in chapter 3, with the dataset specified in paragraph 2.3. As largely disclosed, the aim of this thesis is to implement ML models for binary classification, in order to reproduce the validation process, predicting the class of pertinence of the dosimetry under evaluation. In order to assess the performance of a model, few metrics have been used and described in paragraph 4.1. The following paragraphs analyse the results obtained, aiming to retrieve a proper subset of hyperparameter combinations for the specific problem examined, as explained in the introduction of chapter 3.

4.1. Metric description

Regarding classification problems, the performance of a specific ML model can be measured in different ways: the most used ones are accuracy and precision. For each measurement in the testing dataset, the model provides a predicted class which may differ from the real one. Therefore, after the implementation of the model on the whole testing dataset, a so-called confusion matrix can be constructed. This matrix quantifies how many measurements have been predicted correctly and how many have been instead misclassified: graphically, correct predictions lay on the diagonal elements of the matrix, while incorrect ones fill the remaining positions. An example of confusion matrix for binary classification is reported in Figure 4.1, the dimension of the confusion matrix depends on the number of classes for the classification. Since this discussion focuses on binary classification, a 2×2 schematization of a confusion matrix is reported in Figure 4.1.

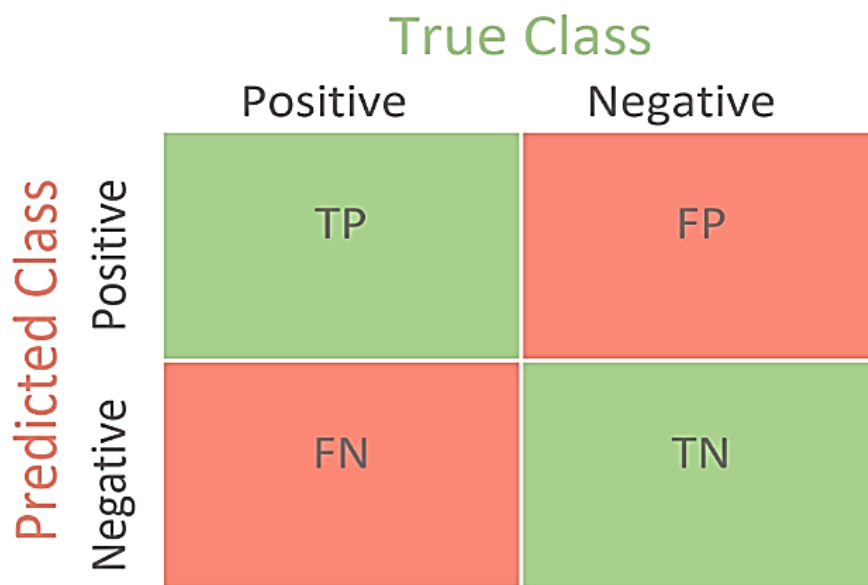


Figure 4.1: schematic confusion matrix for binary classification [60]

The confusion matrix is fundamental to understand the concept of accuracy and precision for classifiers. In Figure 4.1 the correctly predicted measures are labelled as “True”, while the misclassified ones are labelled as “False” leading to “True Positive” (TP), “True Negative” (TN), “False Positive” (FP) and “False Negative” (FN). Measurements labelled as TP belong to the positive class and are correctly classified as positive, analogously to the negative class in case of TN. This means that the quality “True” define a measurement that has been classified in the correct way. Concerning FP, they are measurements that have been classified as negative, even though they belong to the positive class. On the contrary, FN are measurements that belong to the negative class but have been classified in the positive one. The quality “False” describe misclassified measurements.

In order to link this confusion matrix to the current case study: the positive class is represented by dosimeters class *A*, while the negative one refers to class *L*. Hence, in the entry previously called TP,

hereafter called “True A ” (TA), the measurements that have been validated as class A and have been correctly predicted by the model with class A are collected. On the contrary, measurements validated and predicted as class L belongs to the cell “True L ” (TL), previously called TN. Entries named FN and FP have become respectively “False A ” (FA) and “False L ” (FL).

Accuracy

For a ML classifier, accuracy is expressed as the ratio between the sum of measures correctly classified, divided by the whole number of classified measurements. According to the notation used in Figure 4.1 and in the previous paragraph, the accuracy (A) is evaluated as follows:

$$A = \frac{(TP + TN)}{(TP + FP + FN + TN)} = \frac{(TA + TL)}{(TA + FA + FL + TL)} \quad (4.1)$$

In fact, accuracy gives information on the performance of the classifier regarding all the classes. If the classification process concerns more than two classes, the accuracy will be calculated considering all the entries of the confusion matrix.

The accuracy A is defined as a dimensionless quantity, since it is evaluated as a ratio between numbers, and it is generally expressed as a percentage.

Precision

The precision of the classifier is a measure of how precisely the model classifies measurements inside the so-called positive class (considering then both TP and FP). Indeed, it is evaluated considering just the column of the confusion matrix related to the positive class (i.e., the left one in the example in Figure 4.1). Below, the formula for the evaluation of the precision (P) is reported:

$$P = \frac{TP}{(TP + FP)} = \frac{TA}{(TA + FA)} \quad (4.2)$$

For the definition of precision, it is important to establish which class is defined as positive, which was not mandatory for the concept of accuracy, as retrieved by formula 4.1. Similarly, in this instance the precision is a dimensionless quantity usually expressed as a percentage.

4.2. Cross validation

As briefly explained at the beginning of chapter 3, Cross-Validation (CV) is performed in order to assess the presence of overfitting. Typically, CV is performed dividing the dataset in k folds: $k - 1$ folds are used for training, while the remaining one is used for testing. Then the same model is trained on the training set and validated against the testing set, and this process is iterated k times in order to use all k folds once for model testing. At the end of this process, accuracy and precision retrieved from the k implementations are compared with the results pointed out previously. This CV technique is usually called K-Fold cross validation, although this kind of CV is the only one performed in this study, it is not the only choice available.

A first CV process has been performed for the hyperparameter subset combinations set for KNN models, dividing the dataset in 5 folds ($k = 5$). Five folds are created leading to a testing dataset that contains 20% of the whole values of the dataset. Following this first approach, results from the cross validation were remarkably high in precision and accuracy, as described in paragraph 4.4.1. The reason could be related to the fact that, if the whole database is shuffled and split casually in 5, there is a high probability that measurements that has been validated together are divided within training and testing dataset. This could cause some bias for classifiers such as KNN: these measurements have same dosimeter serial number as well as validation month and other features. For this reason, the accuracy and precision in these cases have risen due to “proximity” of these measurements.

In order to overcome this problem, CV for all the implemented models has been performed dividing the dataset per month in 11 folds ($k = 11$), since the measurements in the dataset range from January to November 2023. In this way, the percentage of 80 - 20% training-testing is not respected, leading to the

percentages reported on Table 4.1. In the following table, the two columns on the right report the percentage of values contained in testing and training datasets with respect to the whole values in the original dataset.

<i>Month used for testing</i>	<i>Testing dataset [% of total values]</i>	<i>Training dataset [% of total values]</i>
<i>January</i>	4.27	95.73
<i>February</i>	4.28	95.72
<i>March</i>	15.87	84.13
<i>April</i>	9.34	90.66
<i>May</i>	8.98	91.02
<i>June</i>	9.61	90.39
<i>July</i>	8.77	91.2
<i>August</i>	8.52	91.48
<i>September</i>	9.16	90.84
<i>October</i>	9.69	90.31
<i>November</i>	11.51	88.49

Table 4.1: folds percentage of the whole database for cross validation

To take into account the difference in percentage present within the testing datasets that refers to different months, the accuracy and precision that result from this cross validation process have been weighted on the percentage of values contained in each testing dataset presented in Table 4.1.

4.3. Binary classification

As mentioned in paragraph 0, in order to perform a binary classification analysis, the classes in the dataset have been grouped in class *A* and class *L*. This last one contains all the classes but *A* and *I*, as extensively described previously.

Prior to the implementation, the dataset has been split in training and testing datasets. In order to mimic the final aim of this study, i.e., validating the measurements of the last incoming month, the testing dataset is constituted by measurements that refers just to the last validated month, i.e., November in this study. The resulting training dataset contains 87.87% of the whole measurements, while the testing one contains the remaining 12.13%.

It is also important to check that the classes contained in these datasets are balanced, i.e. the two classes are present in a percentage that is close to 50%. Percentages of the *A* and *L* classes in these datasets are reported in Table 4.1. Imbalanced classification problems (i.e. problems in which the classes are inherently not balanced in the datasets) pose a challenge for predictive modelling, such as the ML algorithms used in this thesis. Indeed, these have been designed around the assumption of an equal number of examples for each class, resulting in models that have poor predictive performance in case of class imbalance, specifically for the minority class. Regarding binary classification, the minority class is defined as the class that present lower number of examples. Samples in this class are usually harder to predict because there are few examples, by definition, meaning it is more challenging for a model to

learn the characteristics of examples from this class, and to differentiate examples from this class with respect to the other [59]. As shown in Table 4.2, both the datasets can be considered balanced.

<i>Dataset</i>	<i>A classes [%]</i>	<i>L classes [%]</i>
<i>Training</i>	58.41	41.59
<i>Testing</i>	41.02	58.98

Table 4.2: percentages of A and L classes in the datasets

Once the datasets are ready, the implementation of all the hyperparameter combinations is performed. In Table 4.3, the number of models implemented, as well as highest accuracy and highest precision obtained from all the combinations, are summarised. Detailed analysis of the results is conducted in the following paragraphs.

<i>Model</i>	<i>Number of combinations</i>	<i>Highest accuracy [%]</i>	<i>Highest precision [%]</i>
<i>KNN</i>	2 016	64.29	59.58
<i>DT</i>	112 896	69.64	88.74
<i>RF</i>	262 440	69.63	70.32
<i>NN</i>	42 120	68.43	72.32

Table 4.3: results summary concerning number of combination, highest accuracy and precision

4.4. K-Nearest Neighbour implementation

As mentioned in Table 4.3, 2 016 different KNN models have been implemented on the previously described training and test datasets. In order to detect which hyperparameters lead to better results, plots have been created and reported in the following paragraphs. In Table 4.4, the five best models obtained from the 2 016 combinations are reported, with the relative accuracy and precision. These have been selected as best models since they present the highest precision, but it has to be noticed that these results are obtained disregarding the cross-validation. In addition, in the column that concerns the algorithm hyperparameter “//” is reported: models with the same hyperparameters but different algorithms values give same results in this case, implying that, concerning KNN models, the algorithm used does not change the results.

<i>K</i>	<i>Weights</i>	<i>Algorithm</i>	<i>Metric</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
12	“Distance”	//	“manhattan”	63.82	59.58
13	“Distance”	//	“manhattan”	63.53	59.33
10	“Distance”	//	“manhattan”	63.92	59.29
8	“Distance”	//	“manhattan”	64.16	59.20
11	“Distance”	//	“manhattan”	63.60	59.11

Table 4.4: best KNN models from all hyperparameter combinations

Number of neighbours K

Concerning the best value for the K hyperparameter, extensively described in paragraph 0, it is difficult to identify an overall best value. As shown in Figure 4.2, higher values of K provide slightly higher values of precision. In this scatter plots, each dot represents one model implementation, reporting its accuracy and precision on the x and y axis respectively. The highest value of precision, reported in Table 4.4, is obtained with a K value equal to 12, as also depicted in Figure 4.3. In this last figure, it is also clear that KNN models with highest accuracies provide also highest precisions.

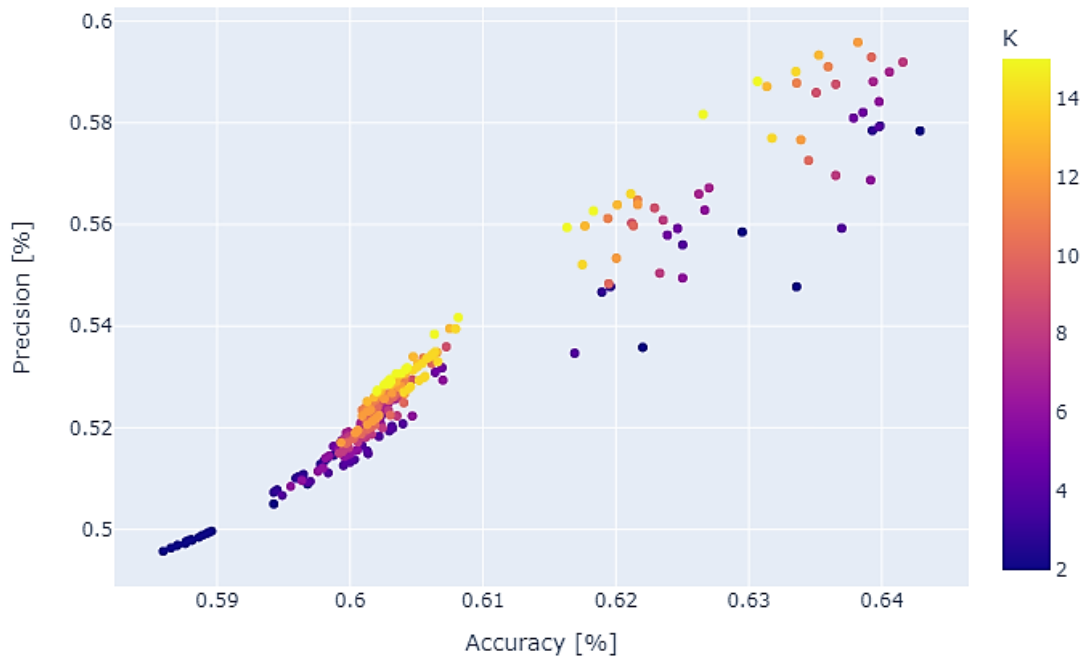


Figure 4.2: scatter plot of KNN models with accuracy, precision and K hyperparameter

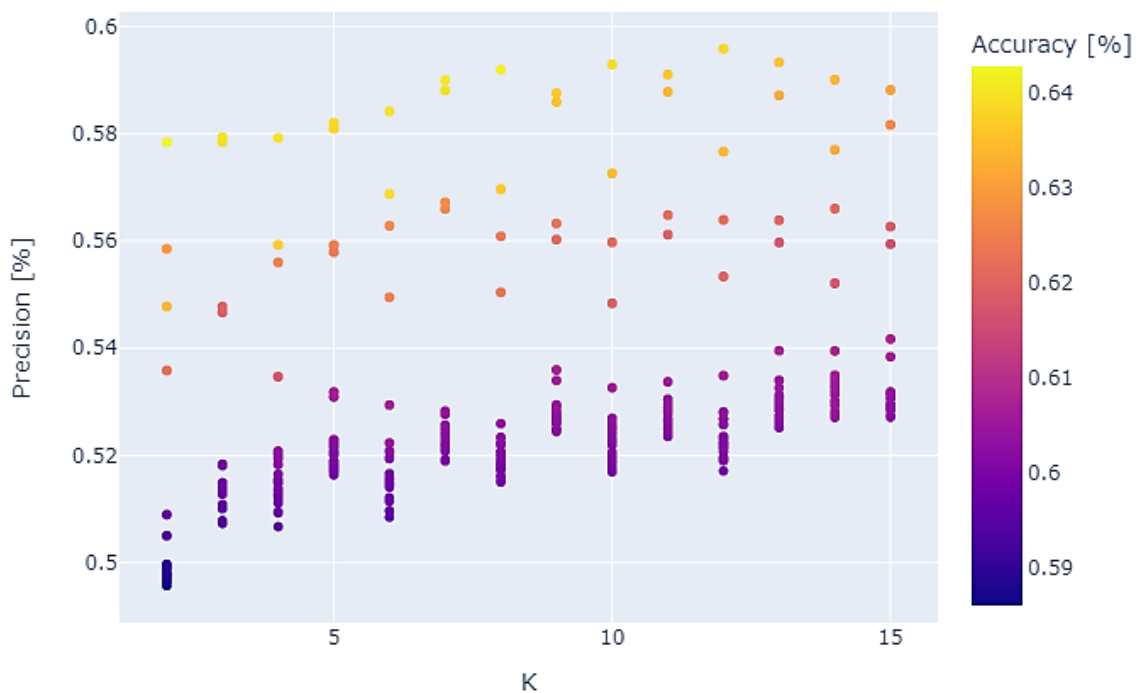


Figure 4.3: scatter plot of KNN models with accuracy, precision and K hyperparameter

Weight

Concerning the weight hyperparameter, no significant differences can be noticed between the two possibilities: “uniform” and “distance”, although the “distance” option is normally preferred by the algorithm. In Figure 4.4, the differences in the results related to this hyperparameter are reported, based on the precision of the models.

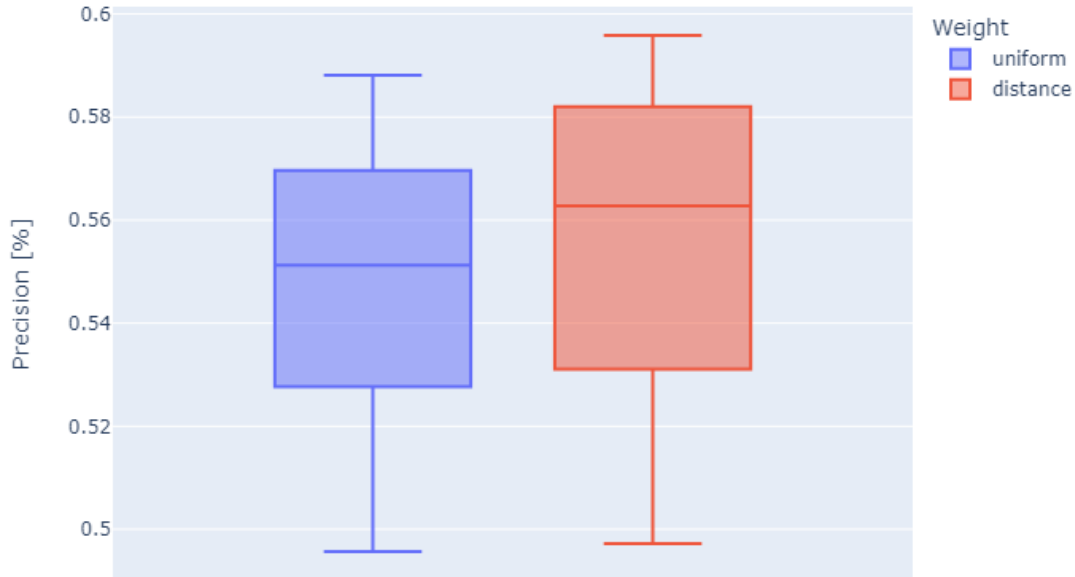


Figure 4.4: box plot of KNN models with precision and weight hyperparameter

Box plots, as the one illustrated in Figure 4.4, show many information within the same framework. Highest and lowest horizontal lines denote respectively the maximum and minimum values, concerning precision in this case. The box part of the plot lays between the first and third quartile, with the horizontal middle line equal to the median of the dataset. Quartiles are values that divide a dataset into four equal parts, each containing 25% of the data, without taking into account outliers. For this reason, the second quartile corresponds to the median.

Algorithms

As redeemed by Figure 4.5, there is not a best algorithm for KNN implementation. All the possibilities perform approximately in the same way and for this reason, all the possible algorithms have been inserted in the subset of hyperparameter combinations, described in paragraph 4.4.1. Indeed, in Table 4.4 all the models with same hyperparameter but the algorithm one result in the same accuracy and precision. Probably, all the implemented algorithms converge to the same solution since there is no more possibility of improvement.

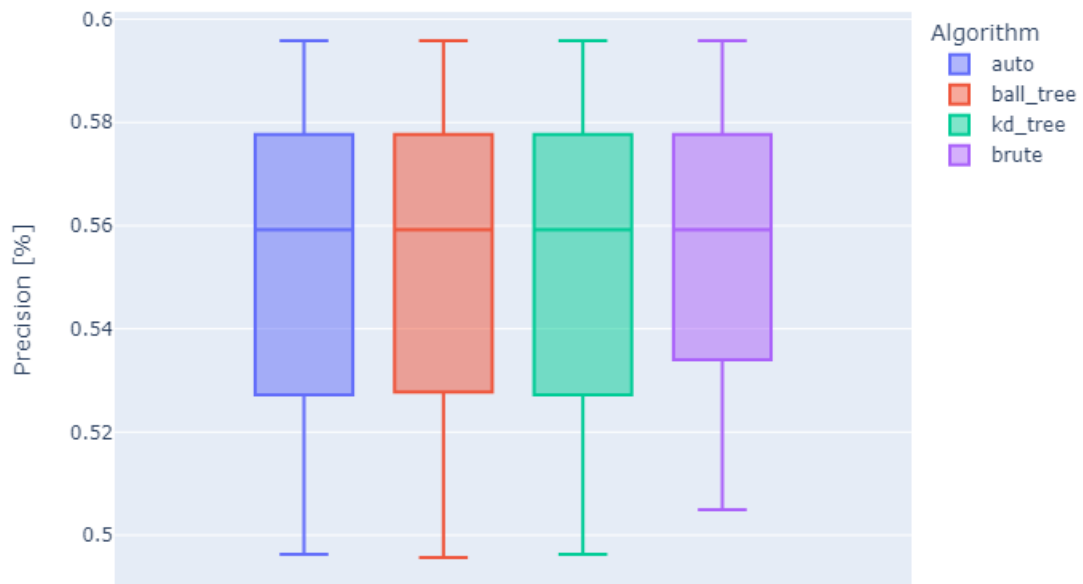


Figure 4.5: box plot of KNN models with precision and algorithm hyperparameter

Metric

From Figure 4.6, it is straightforward to identify the “Manhattan” metric as the one that provide better results. For this reason, it has been chosen as the only metric implemented inside the subset of hyperparameters.

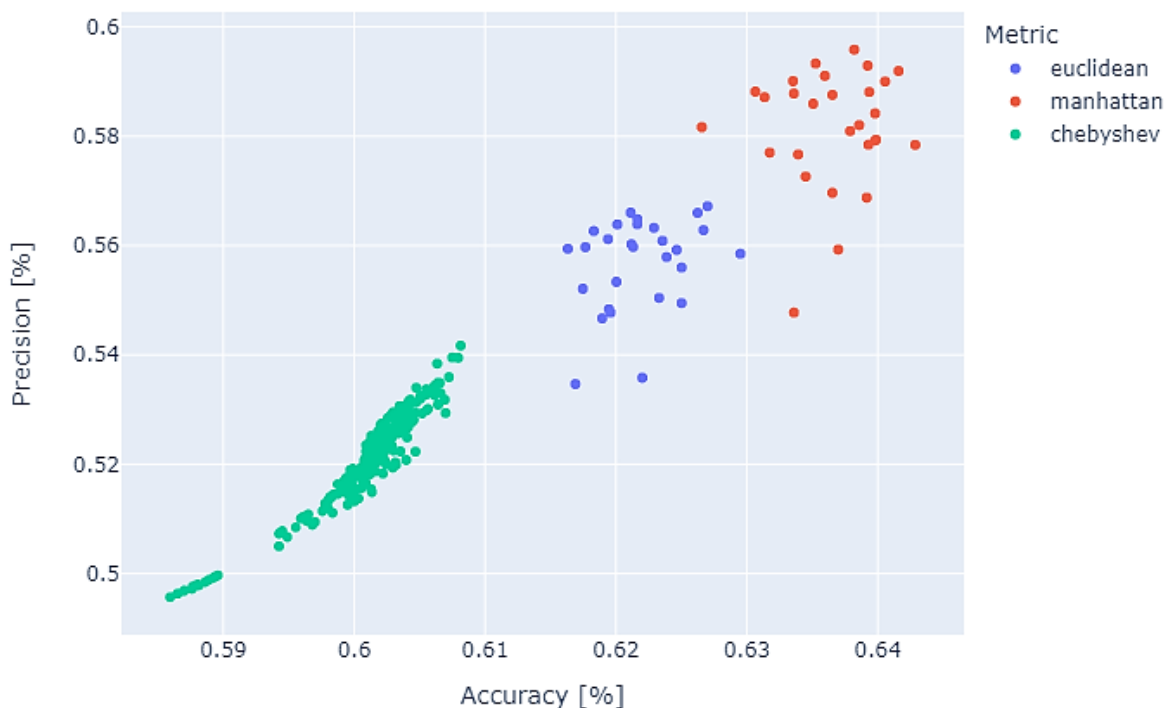


Figure 4.6: scatter plot of KNN models with accuracy, precision and metric hyperparameter

4.4.1. Results of K-Nearest Neighbours models

Following the analysis performed on the results on the whole set of hyperparameters, summarised in the previous paragraphs, a further subset of the latter has been identified in order to facilitate and accelerate

the implementation on future unseen data. The hyperparameters that have been selected are described in Table 4.5.

<i>Hyperparameter</i>	<i>Values</i>
<i>K</i>	(2, 8, 10, 12)
<i>Weight</i>	“Distance”
<i>Algorithms</i>	“Brute”, “K-D tree”, “Ball tree”, “Auto”,
<i>Metrics</i>	“Manhattan”

Table 4.5: hyperparameter subset for KNN classifiers

Considering these selected values, the number of KNN implementations decreased from 2 016 to 16. Therefore, the best results explored so far, for what concern precision of the models, retrieved by this subset, are shown in Table 4.6. These are representative of the 2016 combinations previously implemented. Indeed, the results shown are in line with the highest accuracy and precision shown in Table 4.3 and Table 4.4.

<i>K</i>	<i>Weights</i>	<i>Algorithm</i>	<i>Metric</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
12	“Distance”	“auto”	“manhattan”	63.82	59.58
12	“Distance”	“Ball tree”	“manhattan”	63.82	59.58
12	“Distance”	“K-D tree”	“manhattan”	63.82	59.58
12	“Distance”	“brute”	“manhattan”	63.82	59.58
10	“Distance”	“auto”	“manhattan”	63.92	59.29

Table 4.6: 5 best KNN models from selected subset of hyperparameter combinations

Cross validation

As cited in paragraph 4.2, a first cross validation process has been performed for the previously mentioned 16 KNN models, dividing the dataset in five folds. Results from this first cross validation approach were surprisingly high in precision and accuracy, reaching an average value around ~ 88%. Reasons have been extensively explained in paragraph 4.2, and the newly described cross validation approach has been applied to the 16 KNN models.

The accuracies and precisions that result from this new cross validation process are now closer in value to the results shown in the previous paragraph for what concern KNN models. Indeed, from the 16 models, a mean accuracy of ~ 67% has been retrieved, as well as a precision of ~ 64%.

4.5. Decision Tree implementation

112 896 different DTs has been implemented for this study. Following the same logic introduced for the KNN models, the best five DT models are reported in Table 4.7 citing the hyperparameters analysed in chapter 3.2.1, these are the models that present the highest precisions. As described in the KNN implementation, results in this table are obtained without applying cross validation.

Tree depth	Maximum leaf nodes	Criterion	Minimum samples per leaf	Max features	Accuracy [%]	Precision [%]
40	50	“entropy”	2	“sqrt”	60.13	88.74
None	50	“entropy”	6	“sqrt”	59.77	88.68
25	30	“gini”	1	“log2”	60.34	80.29
40	50	“gini”	7	None	60.08	79.18
40	50	“entropy”	4	“sqrt”	59.21	78.69

Table 4.7: best DT models from all hyperparameter combinations

Tree depth

Concerning the hyperparameter that regulate how many layers are present in the DT, it is challenging to find an overall best value, as shown in Figure 4.7.

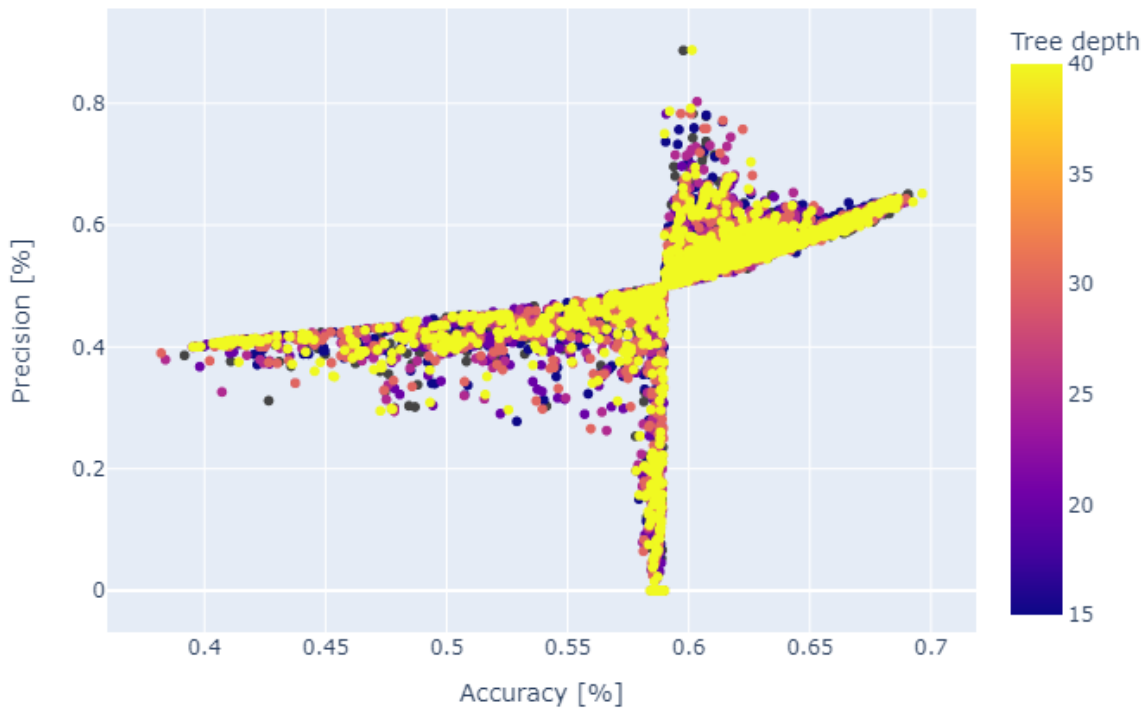


Figure 4.7: scatter plot of DT models with accuracy, precision and tree depth hyperparameter

However, it is possible to plot the maximum precision obtained with different values of tree depth, in Figure 4.8. It is fundamental to underline that DT models with tree depth hyperparameter equal to “None” are not reported in this graph. However, the latter option presents good results, as shown in Table 4.7, in addition from the same table it may be noticed that a tree depth value equal to 40 provide good results as well, as reported in Figure 4.8.

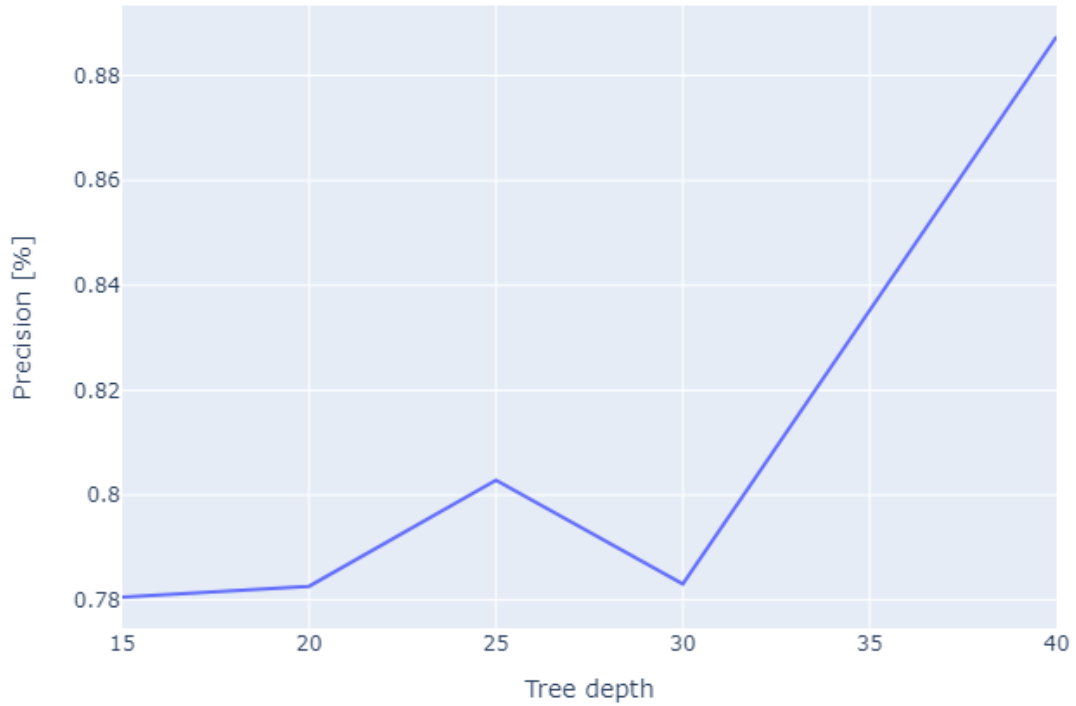


Figure 4.8: line plot with highest precision for DT models related to tree depth hyperparameter

Maximum leaf nodes

Figure 4.9 shows how accuracy and precision of the DT models are related to the maximum leaf node hyperparameter. A minor trend has been identified: higher values of this parameter are related to higher values of precision. On the counterpart, if it is set to “None”, reported as grey dots in the graph, the related accuracy increases. Considering that the objective of this study is to focus mainly on precision, higher values of the maximum leaf node hyperparameter are selected for the hyperparameter subset.

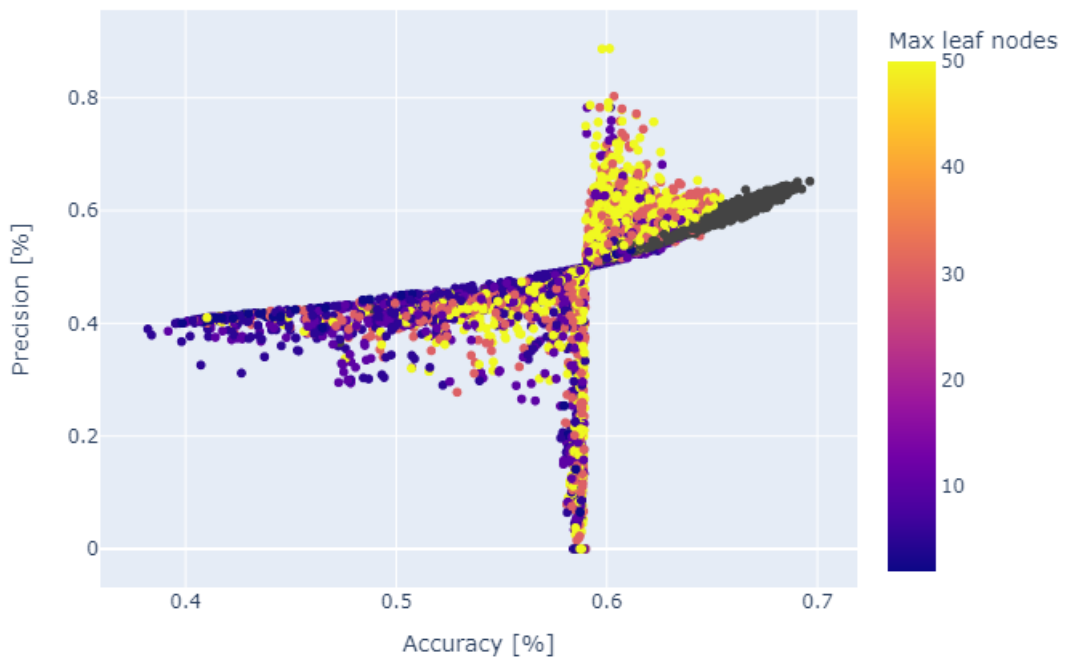


Figure 4.9: scatter plot of DT models with accuracy, precision and maximum leaf nodes hyperparameter

Criterion

Concerning the criterion hyperparameter, no fundamental differences could be appreciated from results analysis. From Figure 4.10 and Figure 4.11 it can be inferred that the results concerning this hyperparameter are substantially similar, thus, not a single criterion has been chosen for the final subset of hyperparameters that concern the DT models. It has to be noticed that the box plot shown in Figure 4.11 does not present the typical box shapes due to high dispersion of the results values.

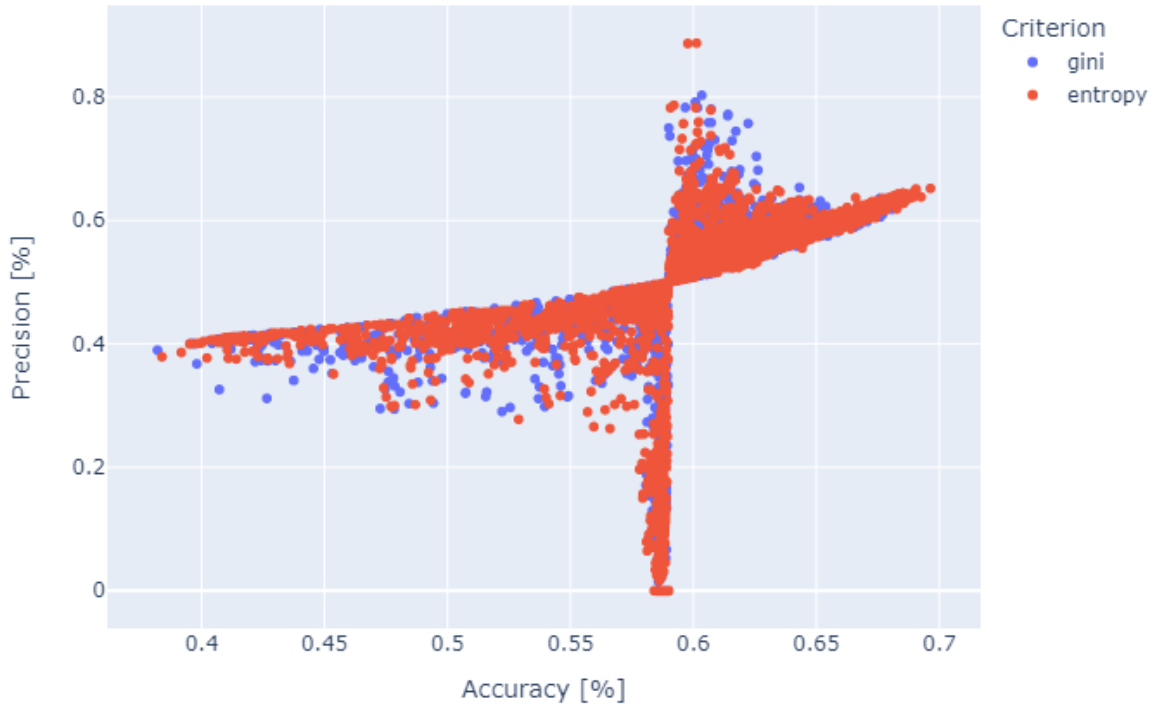


Figure 4.10: scatter plot of DT models with accuracy, precision and criterion hyperparameter

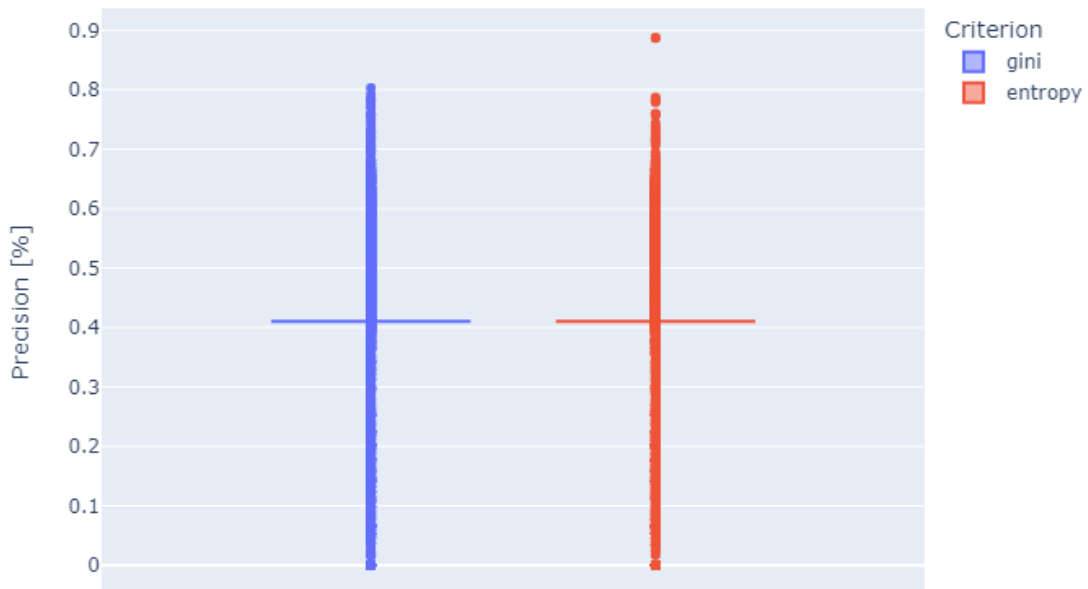


Figure 4.11: box plot of DT models with precision and criterion hyperparameter

Minimum samples per leaf

The same considerations can be deduced for the minimum samples per leaf hyperparameter, hence no specific options can be selected looking at the scatter plot reported in Figure 4.12. However, looking at the line plot in Figure 4.13, containing the maximum precision values for DT implementations, slightly better results have been achieved with minimum samples per leaf values of 2 and 6. Nevertheless, the range of precisions reported in this graph is small, meaning that the change in the results is not largely notable.

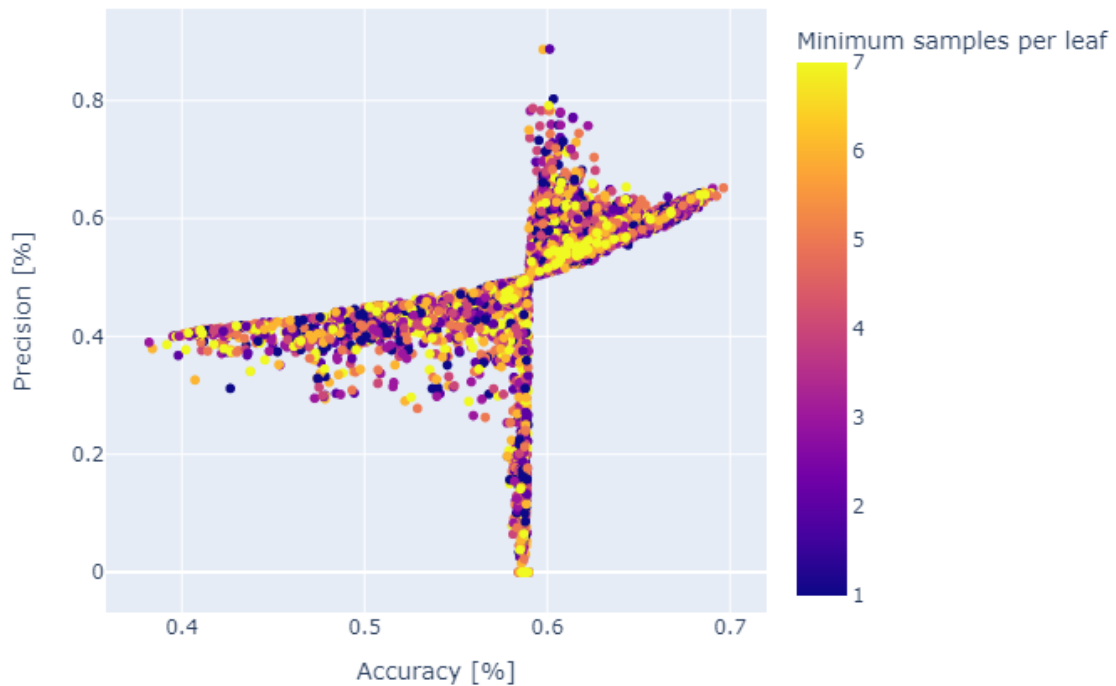


Figure 4.12: scatter plot of DT models with accuracy, precision and minimum samples per leaf hyperparameter

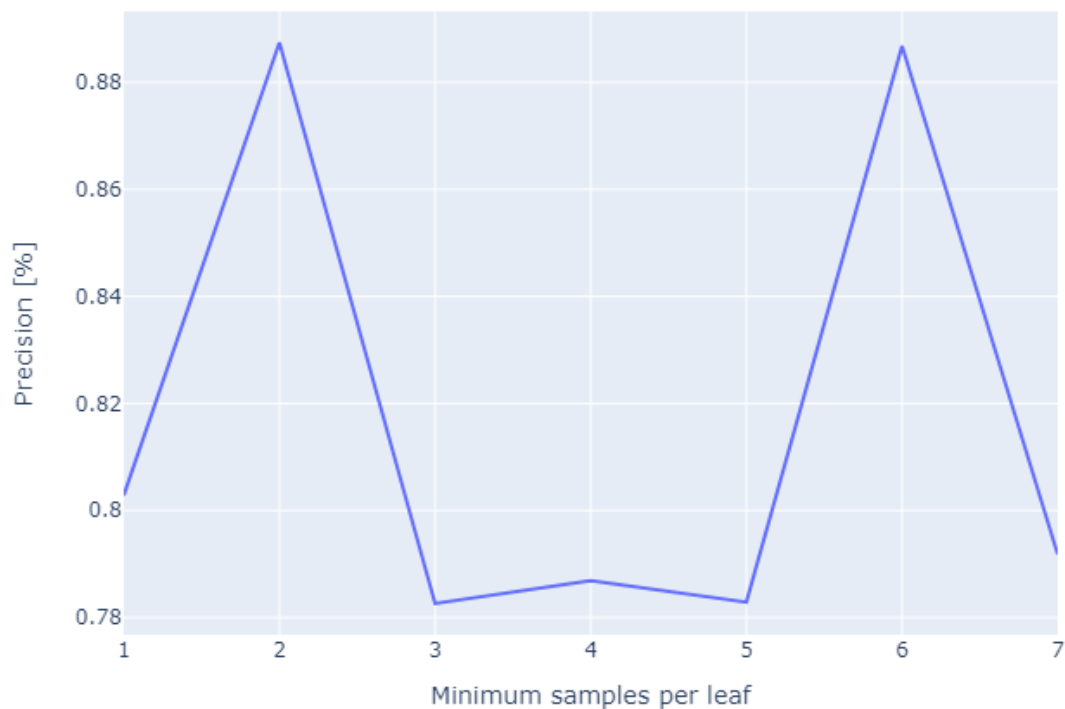


Figure 4.13: line plot with highest precisions for DT models related to minimum samples per leaf hyperparameter

Max features

Regarding scatter plot in Figure 4.14, the only information that can be retrieved concerns the value of maximum features hyperparameter equal to 0.9. Indeed, this choice gives relatively high values of accuracy, as shown in the picture, while precision values remain concentrated around 60%. However, the box plot reported in Figure 4.15 does not show substantial differences among the three values chosen for this hyperparameter.



Figure 4.14: scatter plot of DT models with accuracy, precision and max features hyperparameter

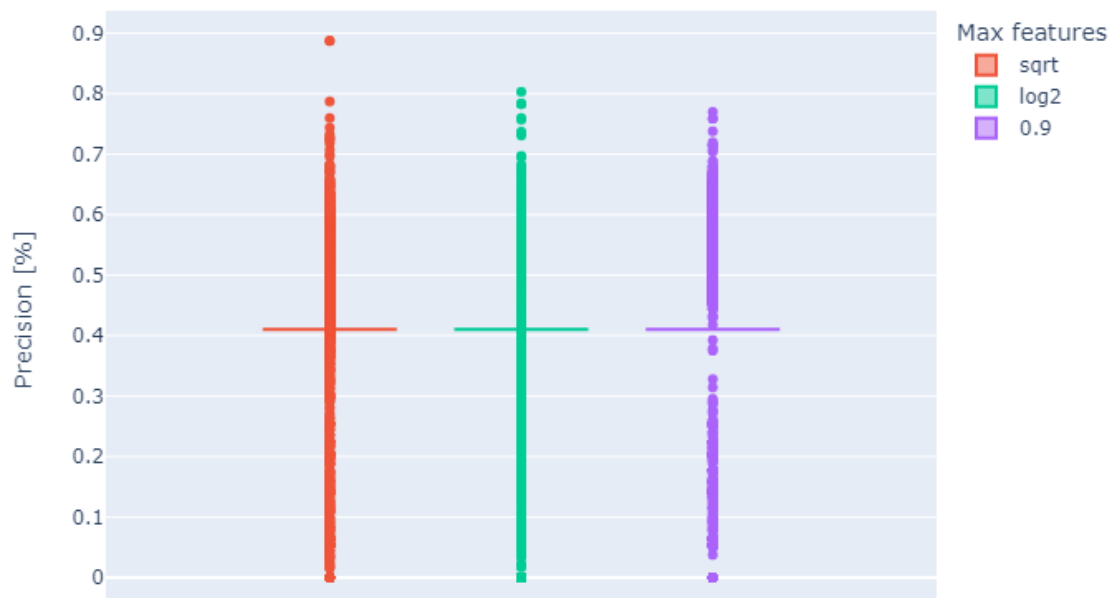


Figure 4.15: box plot of DT models with precision and max features hyperparameter

4.5.1. Results of Decision Tree models

As shown in the previous paragraphs, for the majority of the hyperparameters selected the possibilities analyzed cannot be reduced. However, one of the advantages of the DT models concerns the limited amount of time needed for the training. Thanks to this property, it is possible to not decrease drastically the amount of hyperparameter combinations. Table 4.8 reports the hyperparameter values chosen for the subset. As explained in paragraph 3.2.1, these are not all the hyperparameters selected, since some others are not addressed in this discussion.

<i>Hyperparameter</i>	<i>Values</i>
<i>Tree depth</i>	None - (25, 30, 40)
<i>Maximum leaf nodes</i>	(30, 50)
<i>Criterion</i>	“Gini”, ”Entropy”
<i>Minimum samples per leaf</i>	(2, 5, 6)
<i>Max features</i>	“sqrt”, “log2”

Table 4.8: hyperparameter subset for DT classifiers

Thanks to these measures, hyperparameter combinations are reduced from 112 896 to 576. These are mostly representative of the best models obtained by the implementation of the whole set of combinations. The five best models based on precision results, are reported in Table 4.9.

<i>Tree depth</i>	<i>Maximum leaf nodes</i>	<i>Criterion</i>	<i>Minimum samples per leaf</i>	<i>Max features</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
40	30	“entropy”	2	“sqrt”	60.28	78.06
30	50	“entropy”	6	“sqrt”	60.17	77.13
30	30	“gini”	6	“sqrt”	60.75	66.39
30	30	“gini”	6	“sqrt”	60.76	66.16
30	30	“entropy”	2	“sqrt”	59.70	64.52

Table 4.9: 5 best DT models from selected subset of hyperparameter combinations

Cross validation

These 576 DT models have been subjected to cross validation, following the specifications described for the KNN in previous paragraphs. Highest accuracy and precision obtained from these models are respectively equal to 72.30% and 70.42%.

4.6. Random Forest implementation

Concerning RFs, a massive number of models have been implemented: 262 440. Even though a lot of combinations have been tried, results for accuracy and precision are not promising. Indeed, the five best models, for what concern precision, are reported in Table 4.10, from where it can be inferred that the precisions of these models are lower than the DT ones.

<i>Number of estimators</i>	<i>Tree Depth</i>	<i>Bootstrap</i>	<i>Warm start</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
10	None	“True”	“False”	59.55	70.32
10	10	“False”	“True”	60.46	69.95
50	5	“True”	“True”	61.19	69.54
10	5	“True”	“True”	59.62	69.11
10	None	“True”	“True”	61.25	68.79

Table 4.10: best RF models from all hyperparameter combinations

Number of estimators

The number of estimators hyperparameter is fundamental for the RFs model implementation, however no clear trends can be highlighted from the scatter plot reported in Figure 4.16. In order to carry on a more insightful analysis, a violin plot has been created and shown in Figure 4.17. The width of the violin in these kind of plots gives information about the number of models that give as result that particular value of precision: larger is the violin, higher is the number of models with that result. Even from this plot, no substantial differences are observed, like for the previously mentioned scatter plot.

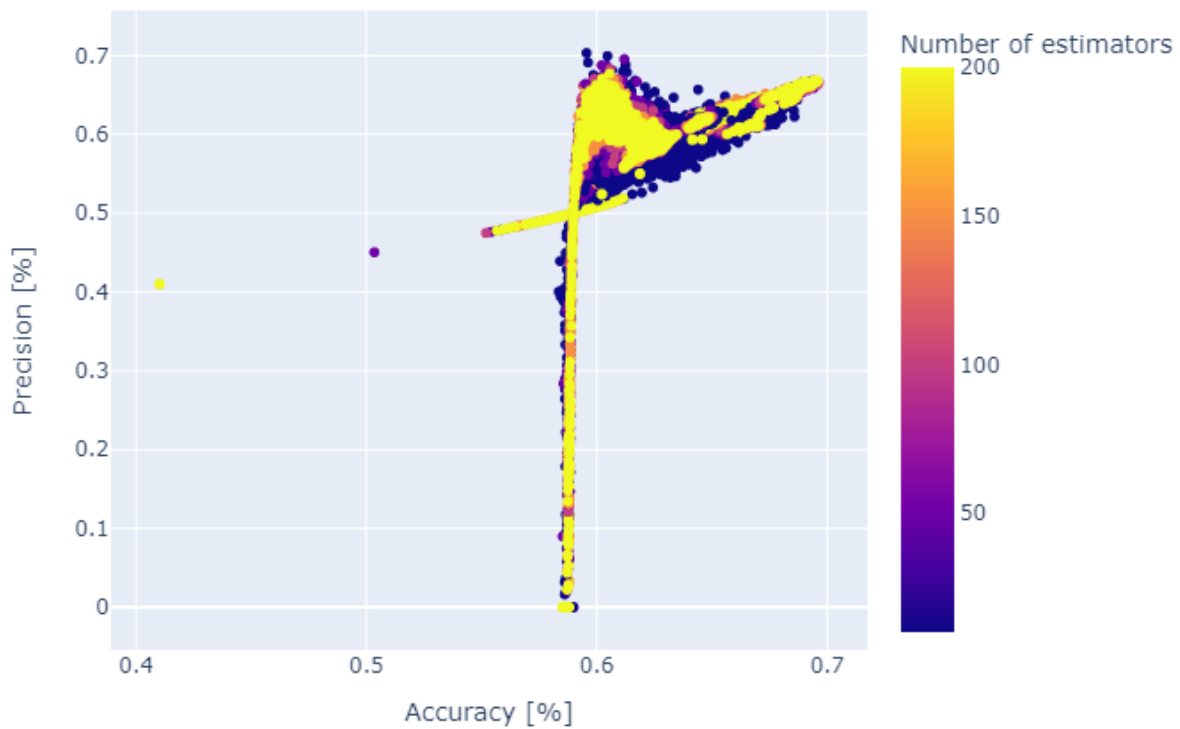


Figure 4.16: scatter plot of RF models with accuracy, precision and number of estimators hyperparameter

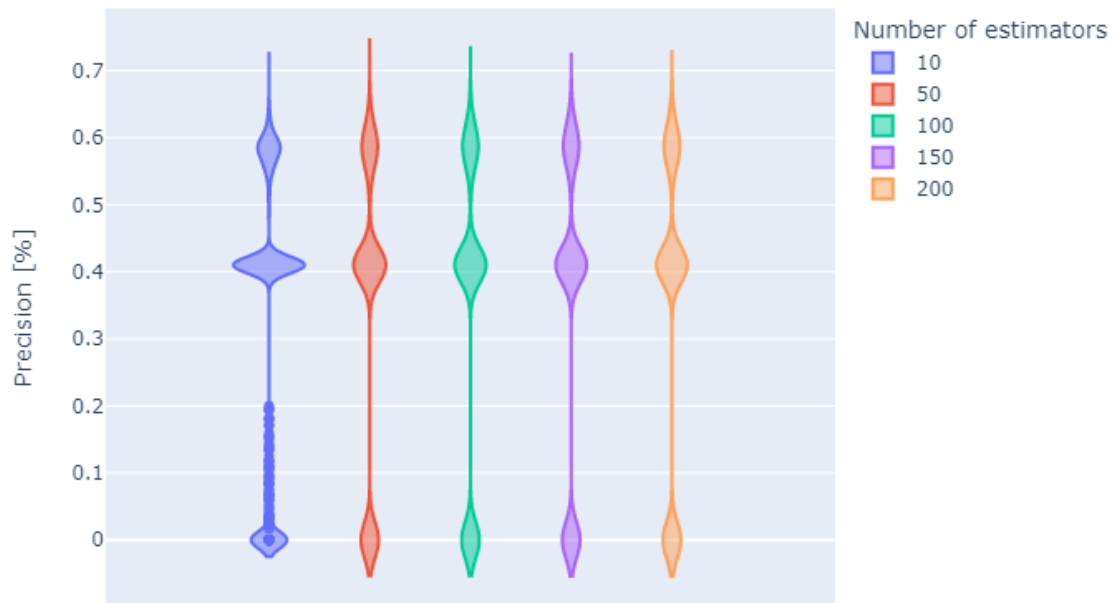


Figure 4.17: violin plot of RF models with precision and number of estimators hyperparameter

Based on these results, just three values for this hyperparameter have been taken into account for the further subset. They are equal to 10, 50, and 200 since, from Table 4.10, a number of estimators equal to 10 in the RF models provided the best values of precision. In addition, the highest number of estimators value (200) has been taken into account for the subset in order to have models with the widest possible range. Concerning this hyperparameter also the value equal 50 has been inserted, since it provides the third highest precision from Table 4.10.

Furthermore, the line plot reported in Figure 4.18 shows the mean values of precision for different number of estimators. When this hyperparameter is set to 10 it provides a slightly highest mean precision along all the values taken into account.

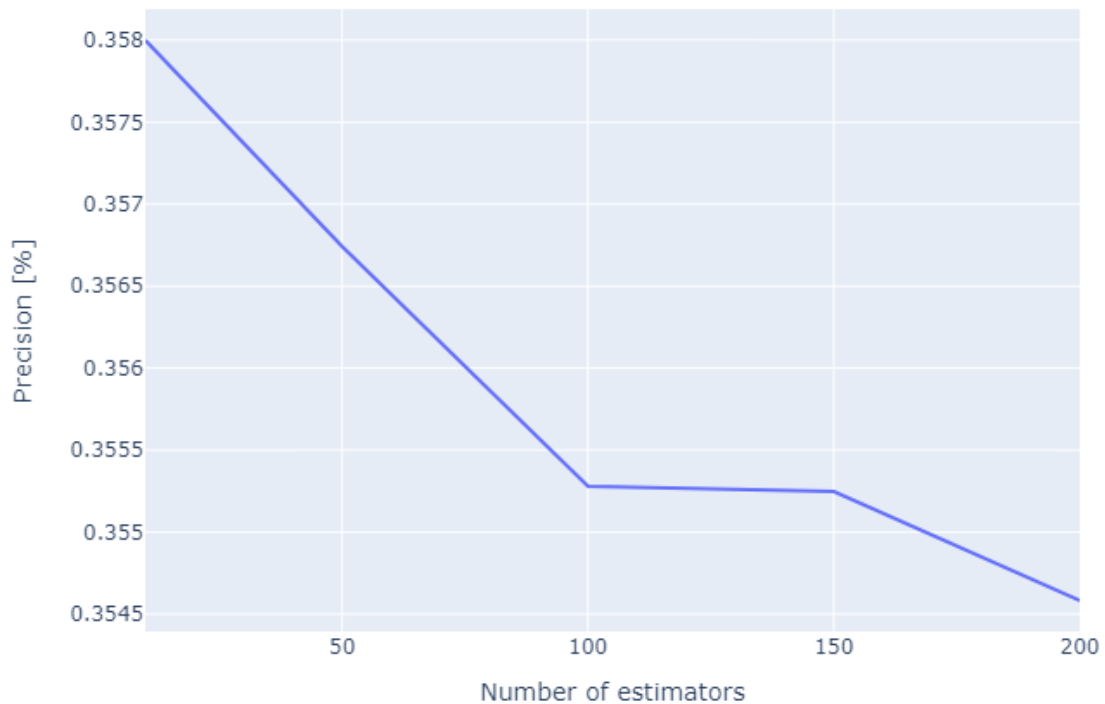


Figure 4.18: line plot with mean precisions for RF models related to number of estimators hyperparameter

Tree depth

Concerning the max depth of the tree contained in the RF models, the following plots (Figure 4.19 and Figure 4.20) have been created to reduce the number of possibilities for this hyperparameter. As done previously, a scatter plot is reported in Figure 4.19, although no useful information can be retrieved from it. In addition, a box plot is shown in Figure 4.20. Unfortunately, this box plot does not contain the models that have a tree depth parameter equal to “None”. However, no trend or better values can be identified by this box plot. For this reason, values for this hyperparameter selected for the subset of combination are: “None”, 10 and 5, since they give the best results as shown in Table 4.10.

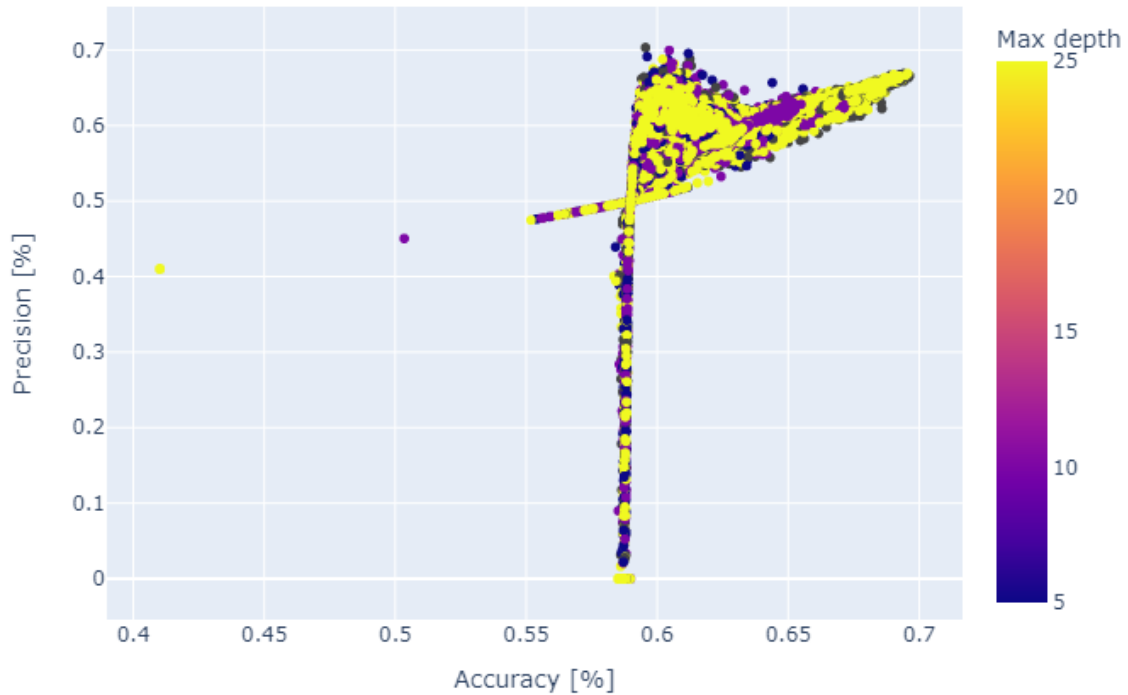


Figure 4.19: scatter plot of RF models with accuracy, precision and tree depth hyperparameter

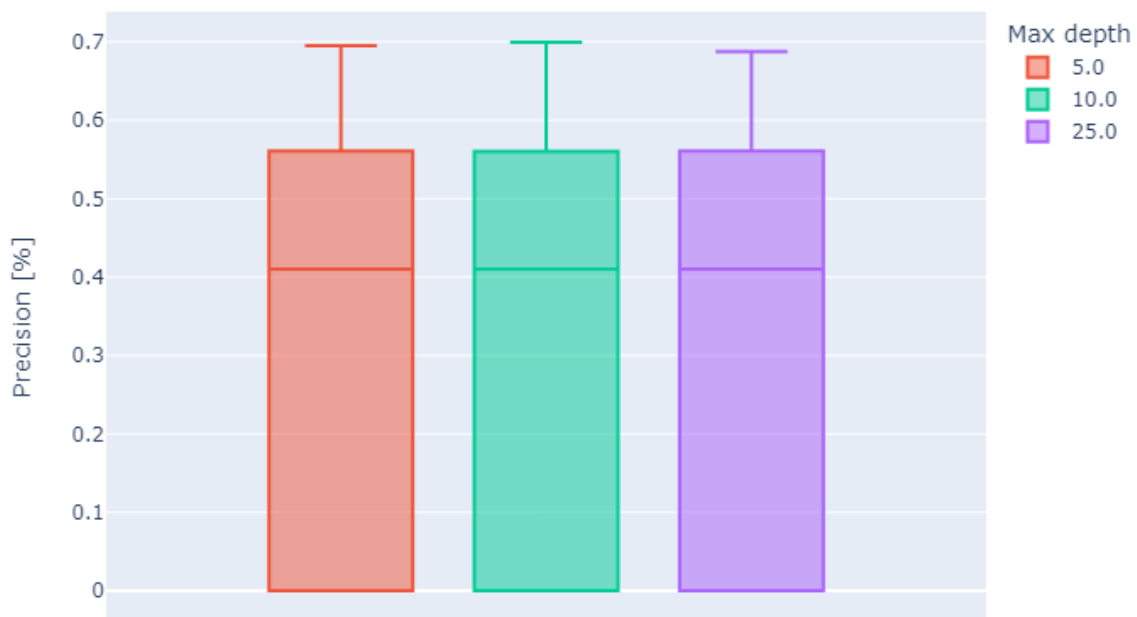


Figure 4.20: box plot of RF models with precision and tree depth hyperparameter

Bootstrap

For what concern the bootstrap hyperparameter, the box plot, depicted in Figure 4.21, shows better results for what concern “True” value. No further information have been retrieved by the scatter plot reported in Figure 4.22.

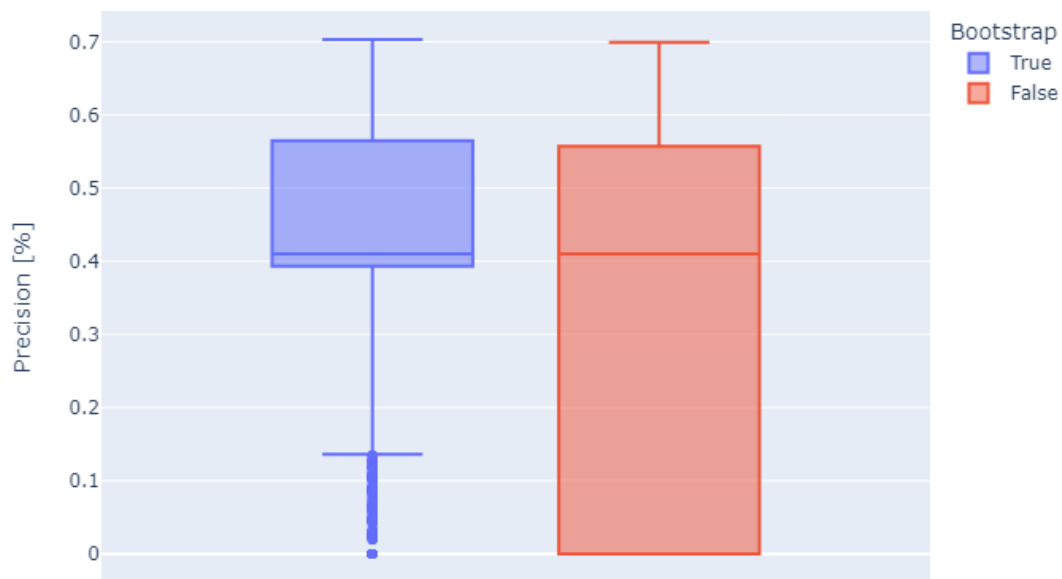


Figure 4.21: box plot of RF models with precision and bootstrap hyperparameter

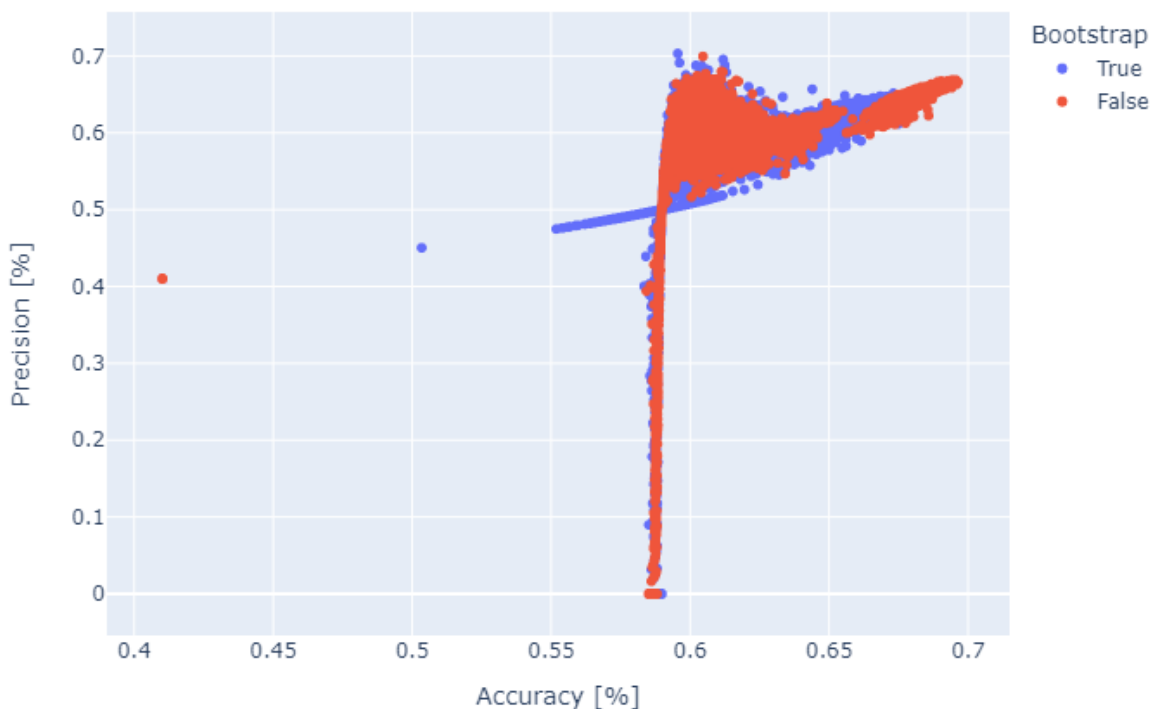


Figure 4.22: scatter plot of RF models with accuracy, precision and bootstrap hyperparameter

Warm start

The warm start hyperparameter can vary within two values: “True” and “False”, as described in paragraph 3.3.1. Scatter plot reported in Figure 4.23, as well as the violin plot reported in Figure 4.24, shows no change in the results with the variation of this parameter. In order to select a value for this

hyperparameter, the most common value in the five best models reported in Table 4.10 has been chosen, i.e. equal to “True”.

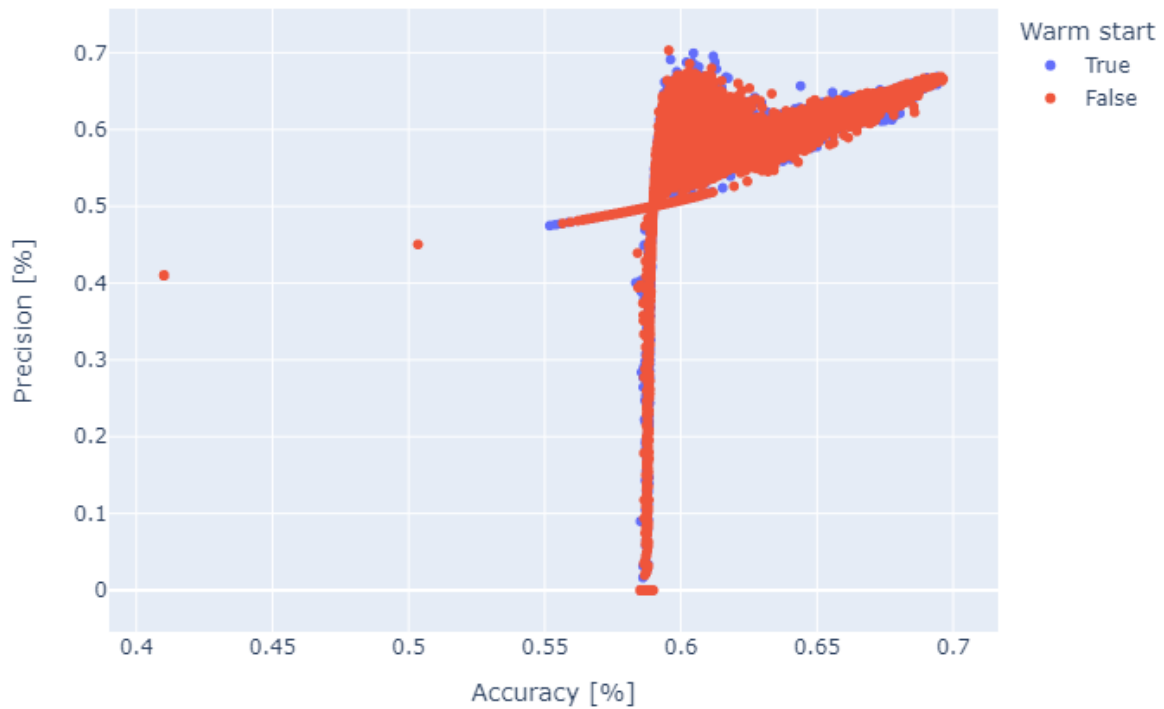


Figure 4.23: scatter plot of RF models with accuracy, precision and warm start hyperparameter

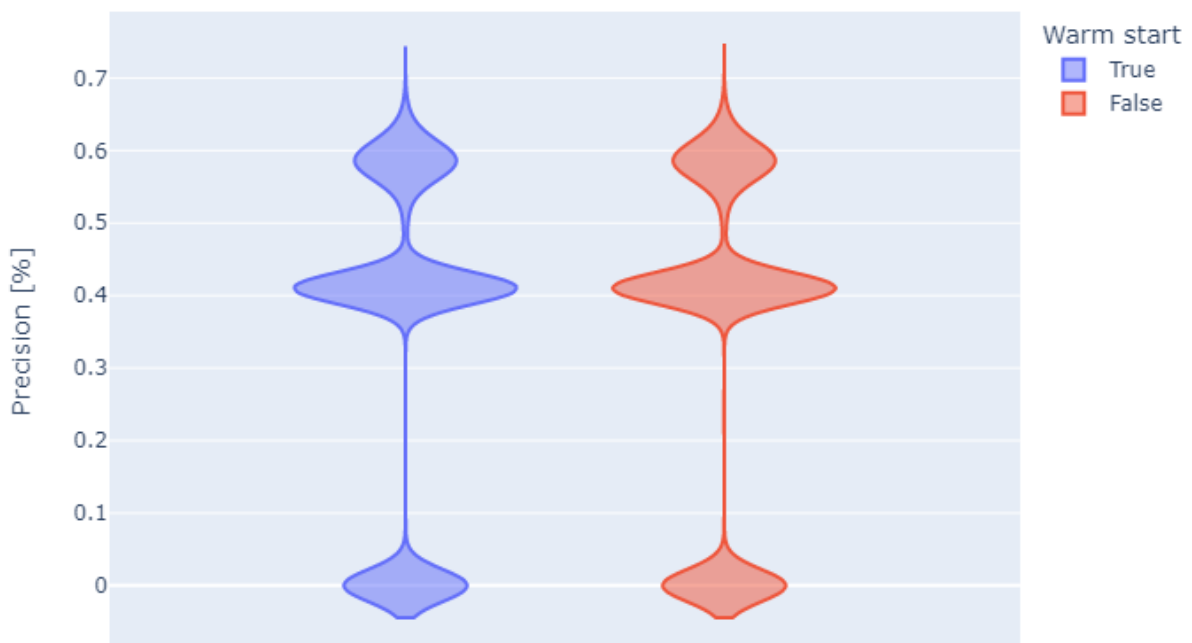


Figure 4.24: violin plot of RF models with precision and warm start hyperparameter

4.6.1. Results of Random Forest models

Concerning RF, the combinations need to be reduced significantly in order to have a feasible subset of models. As for DTs, also RF models are rapid for what concern the training time, for this reason the final amount of hyperparameter combinations in the subset is equal to 576. Table 4.11 reports the

hyperparameter values chosen for the combination subset. As explained also in paragraph 4.5.1 for DT results, these are not all the hyperparameters selected, since many are not addressed in this discussion.

<i>Hyperparameter</i>	<i>Values</i>
<i>Number of estimators</i>	(10, 50, 200)
<i>Tree Depth</i>	None (5, 10)
<i>Bootstrap</i>	“True”
<i>Warm start</i>	“True”

Table 4.11: hyperparameter subset for RF classifiers

The models contained in this subset are quite representative of the best models obtained by the implementation of the whole set of combinations, resumed in Table 4.10. The highest precision changed from 70.32% to 65.18% and the five best models from the subset implementation, based on precision results, are reported in Table 4.12.

<i>Number of estimators</i>	<i>Tree Depth</i>	<i>Bootstrap</i>	<i>Warm start</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
50	10	“True”	“True”	59.99	65.18
200	10	“True”	“True”	59.83	64.88
200	5	“True”	“True”	59.96	64.80
50	5	“True”	“True”	60.18	64.68
200	None	“True”	“True”	59.75	64.13

Table 4.12: 5 best RF models from selected subset of hyperparameter combinations

Cross validation

CV has been performed on these 576 RF models following the specifications described in detail for KNN models. Highest accuracy and precision obtained from these models are respectively equal to 73.81% and 73.58%.

4.7. Neural Networks implementation

Due to the intrinsic complexity of the architecture, as well as the time needed for the training, the number of NN models that have been implemented is smaller with respect to DT and RF ones: 42 120. The five best models, concerning precision, retrieved by these implementations are reported in Table 4.13. As previously done in Table 4.4, in the third row, the “//” symbol is inserted with the same aim explained previously. The notation used for what concern the number of layer and neurons hyperparameter has been already explained in paragraph 3.4.1.

Number of layers and neurons	Activation functions	Solver for weight optimization	Scheduling strategy	Batch size	Accuracy [%]	Precision [%]
(50, 50)	ReLU	“sgd”	“constant”	32	60.40	72.32
(25, 25)	tanh	“sgd”	“adaptive”	32	61.47	70.63
(17, 17, 17)	ReLU	“adam”	//	256	59.93	70.52
(25, 25)	tanh	“sgd”	“constant”	64	62.23	70.01
(50, 50)	ReLU	“sgd”	“invscaling”	64	60.15	69.76

Table 4.13: best NN models from all hyperparameter combinations

Number of layers and neurons

Concerning the number of layers, several values have been taken into account during the implementation. Scatter plot shown in Figure 4.25 is not informative for founding a best value for this parameter, no definite trends or patterns have been identified. On the other hand, the box plot reported in Figure 4.26 gives a better idea of the performance of NNs with different number of layers and neurons. From the latter graph it is possible to assess that (50, 50), (17, 17, 17) and (25, 25) are the best values for what concern precision results, followed by (50) and (50, 50, 50). For the sake of completeness, it is important to mention that the dimension of the output layer is equal to 1.

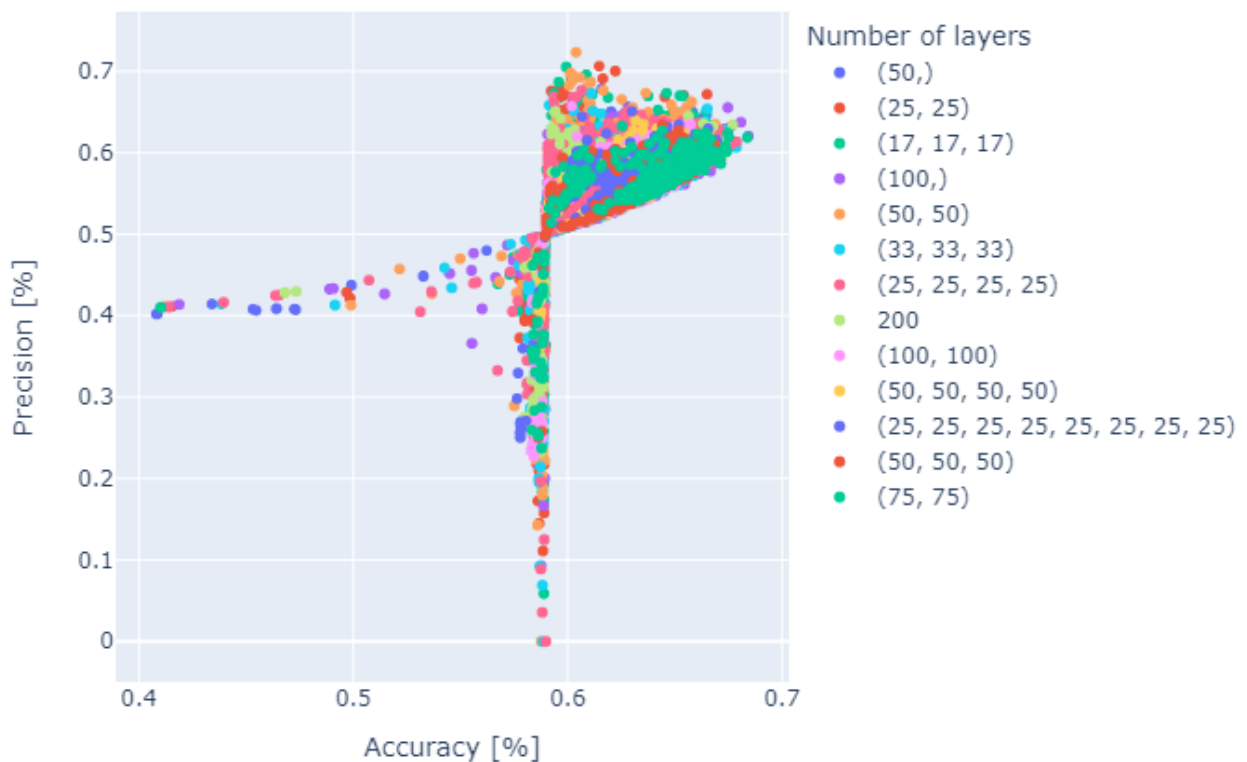


Figure 4.25: scatter plot of NN models with accuracy, precision and number of layers and nodes hyperparameter

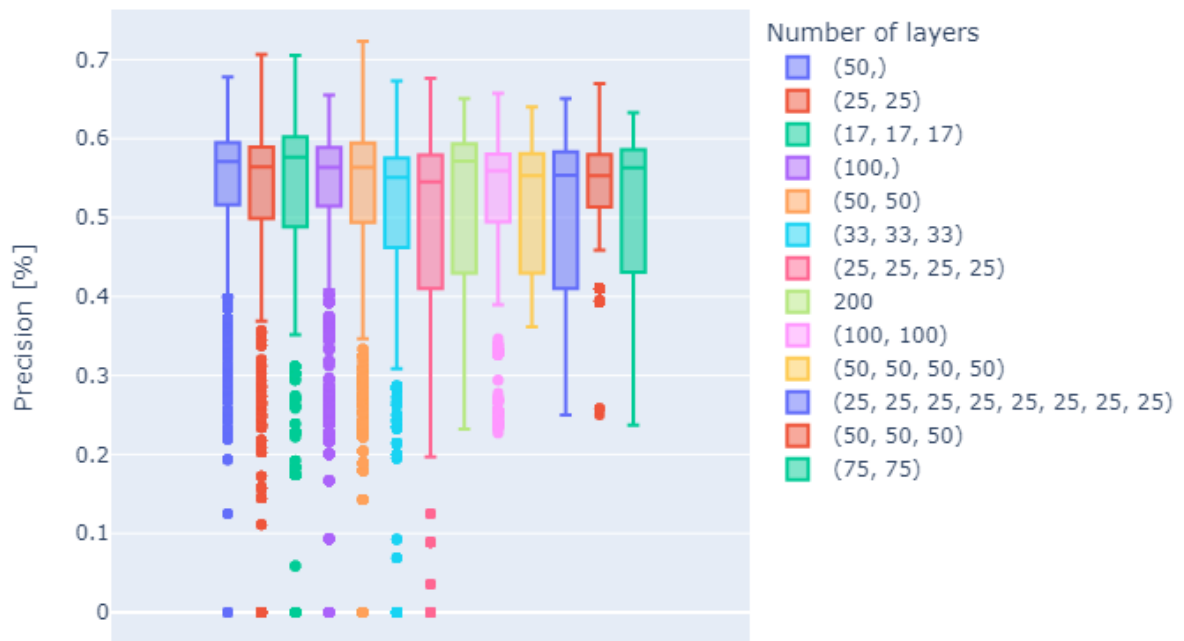


Figure 4.26: box plot of NN models with precision and number of layers and nodes hyperparameter

Activation functions

As done for the previous hyperparameters analysis, also in this situation a scatter plot is reported in Figure 4.27. However, as well as the previous one, it is difficult to retrieve a trend from this kind of plot. On the contrary, box plots are slightly more informative as shown by Figure 4.28. Indeed, this plot shows how the “ReLU” activation function provided the best precision values, while activation function which provided a higher median is “tanh”.

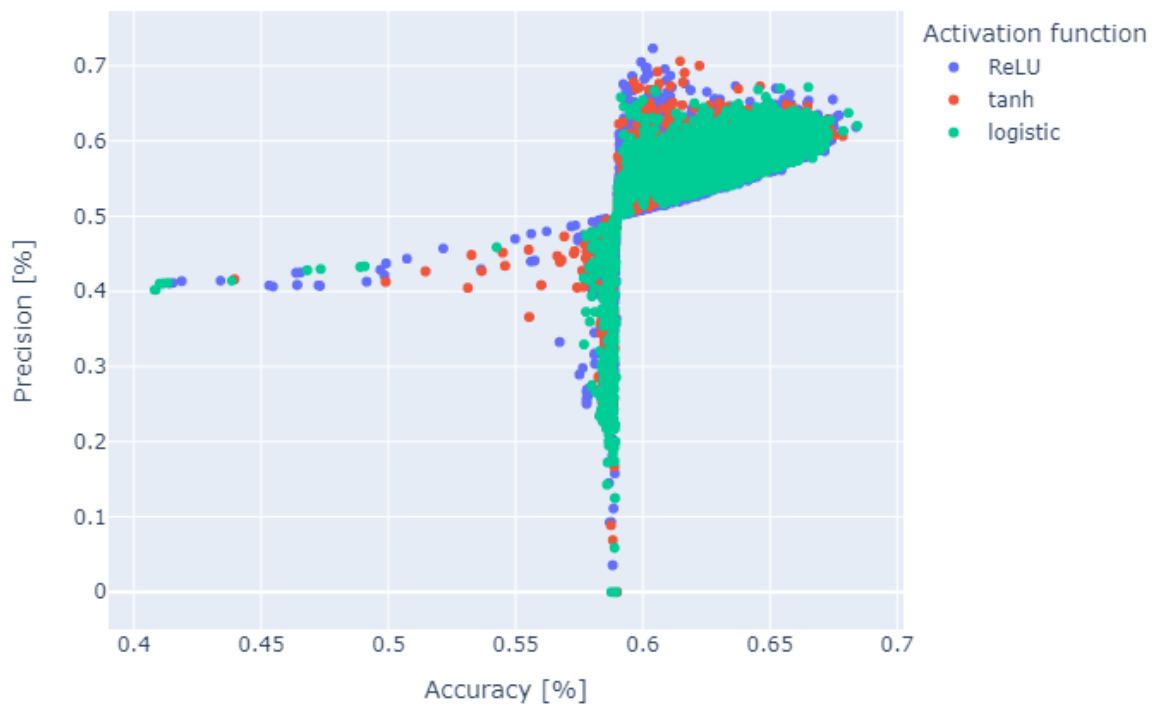


Figure 4.27: scatter plot of NN models with accuracy, precision and activation function hyperparameter

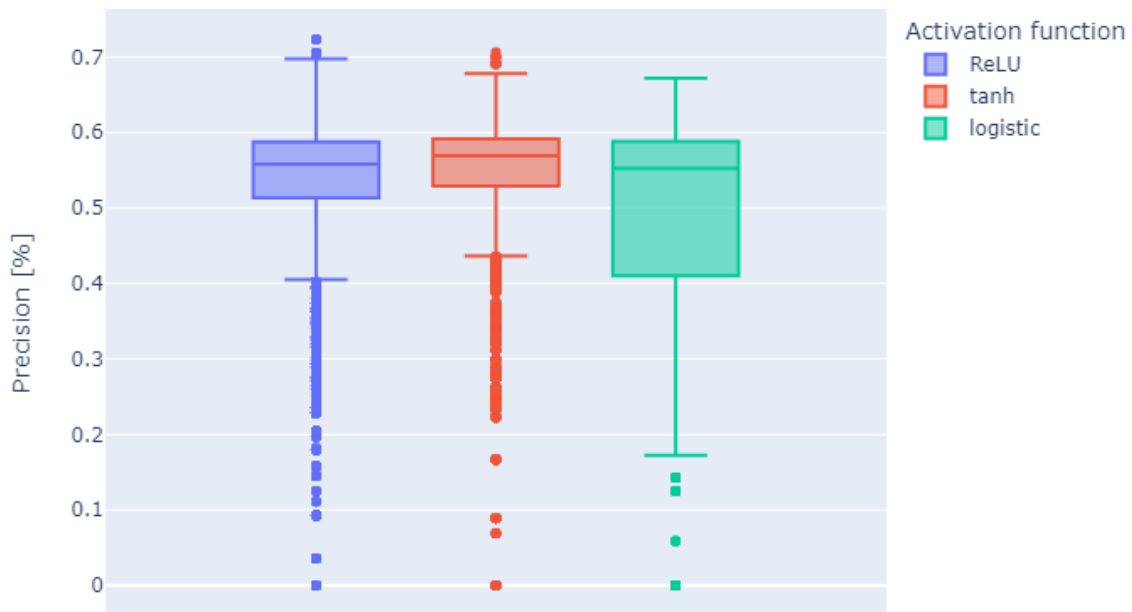


Figure 4.28: box plot of NN models with precision and activation function hyperparameter

Solver for weight optimization

Considering the following plots an analysis regarding the best solver for weight optimization is carried on. The tested possibilities are represented by “adam” and “sgd” solvers. A scatter and a box plot are reported, respectively in Figure 4.29 and Figure 4.30. From the box plot it is clear that more precise models are related to “sgd” solvers, on the counterpart the “adam” solver presents a higher median value for precision results.

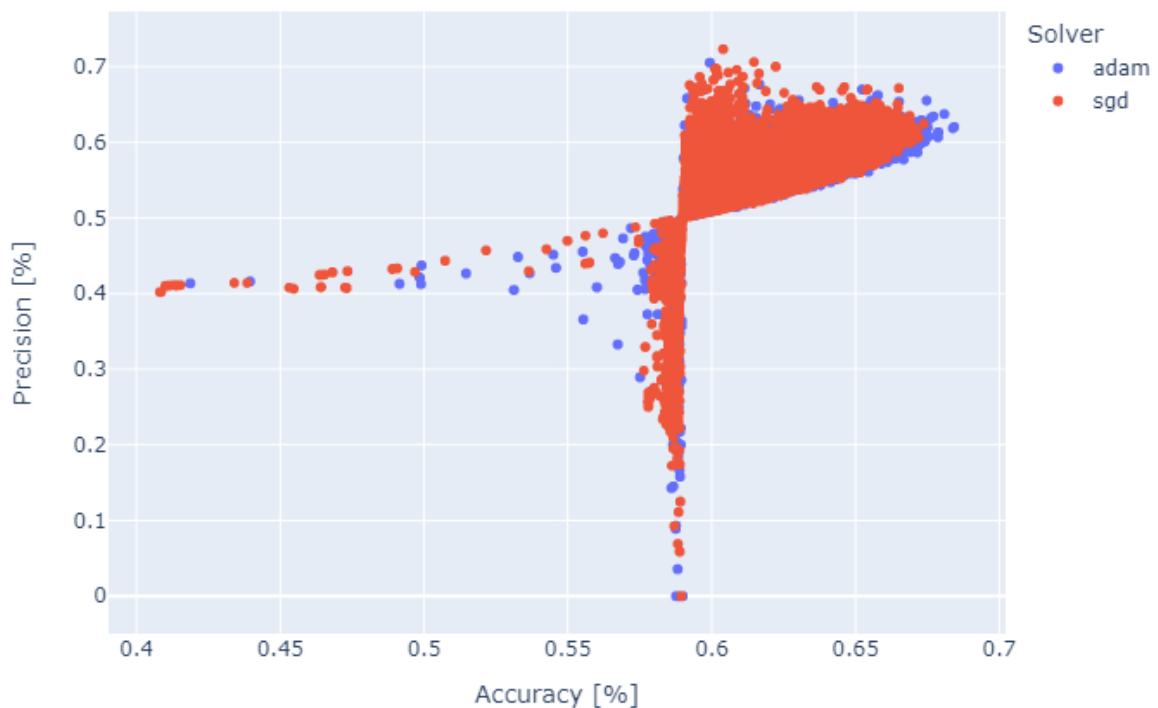


Figure 4.29: scatter plot of NN models with accuracy, precision and solver hyperparameter

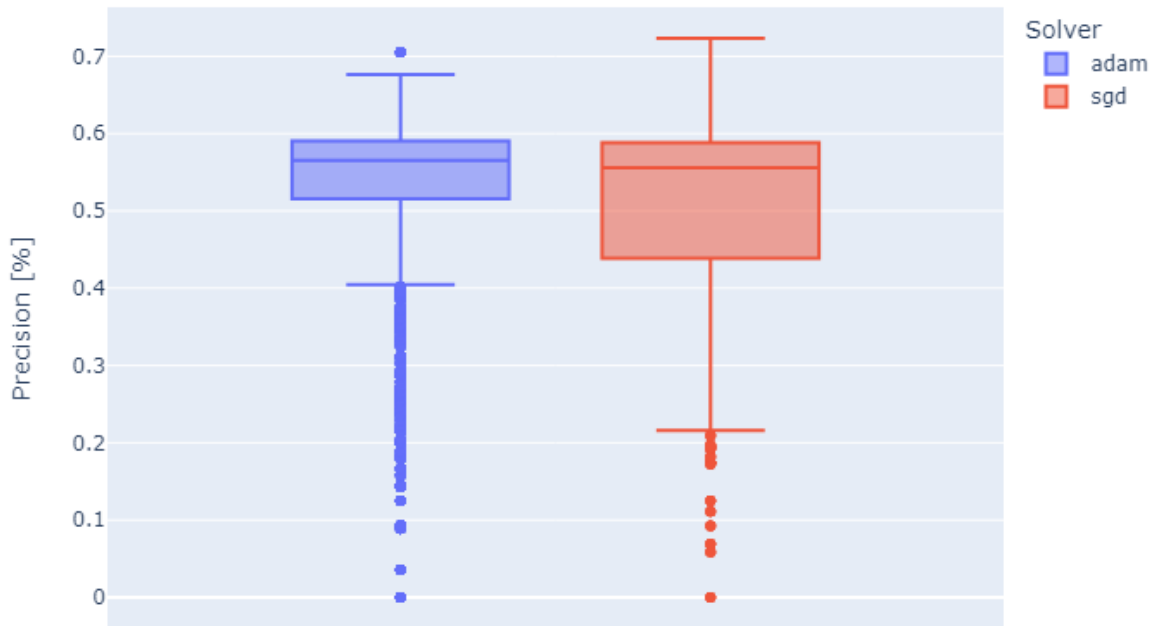


Figure 4.30: box plot of NN models with precision and solver hyperparameter

Scheduling strategy

Concerning the kind of scheduling strategy adopted, defined as “learning rate” in scikit-learn library, it is not well defined which one performs better. This is shown in the scatter and box plot reported below, respectively in Figure 4.31 and Figure 4.32.

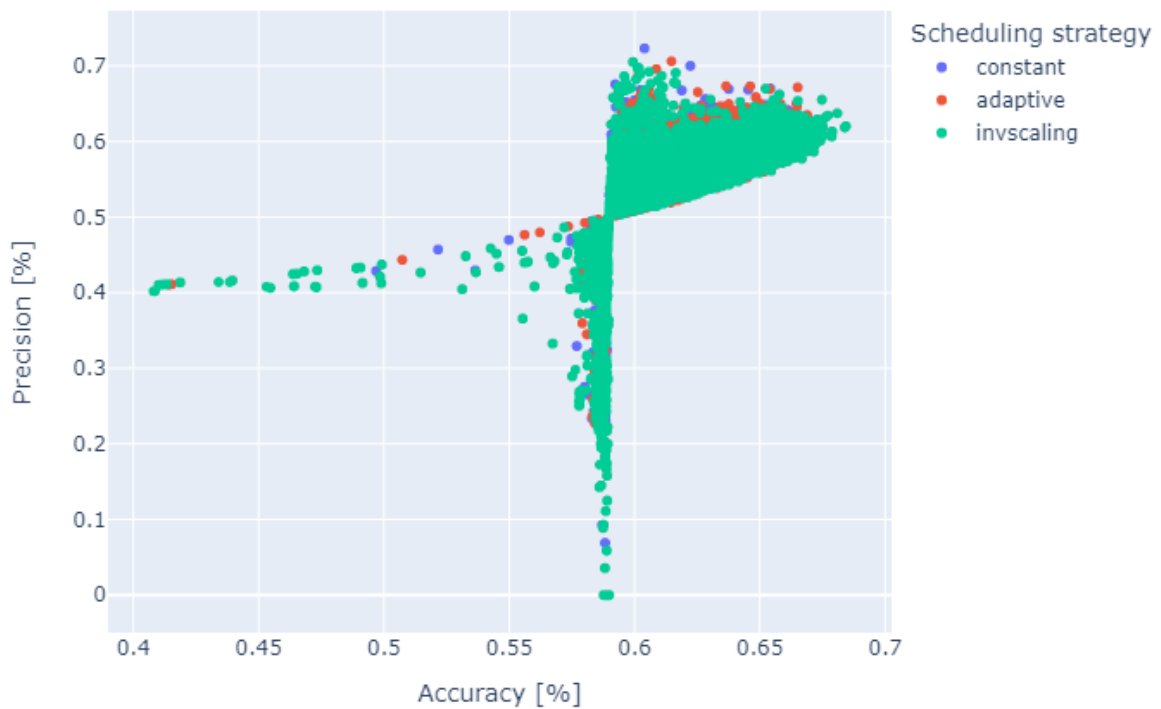


Figure 4.31: scatter plot of NN models with accuracy, precision and learning rate hyperparameter

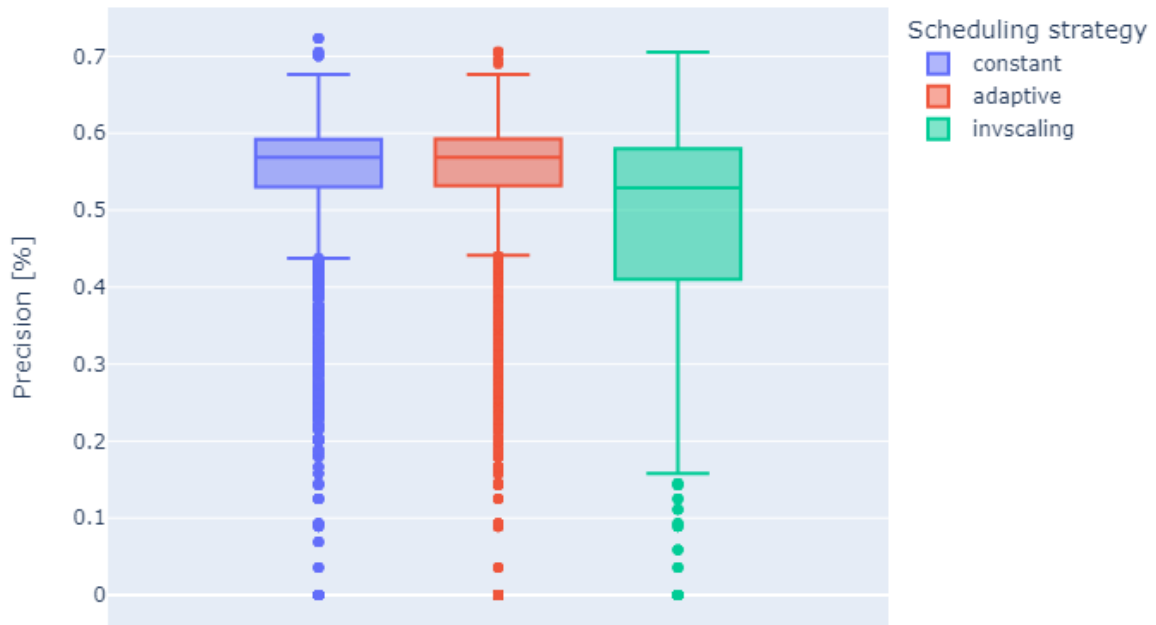


Figure 4.32: box plot of NN models with precision and learning rate hyperparameter

Batch size

The analysis of the batch size hyperparameter has been performed considering a scatter and a box plot, as the other hyperparameters shown for NN models. These are reported respectively in Figure 4.33 and Figure 4.34. No clear patterns have been identified from the scatter plot, but concerning the box plot, it is evident that a batch size of 32 produced the highest precision, while models with value of this parameter equal to 256 results in the higher median value concerning precision.



Figure 4.33: scatter plot of NN models with accuracy, precision and batch size hyperparameter

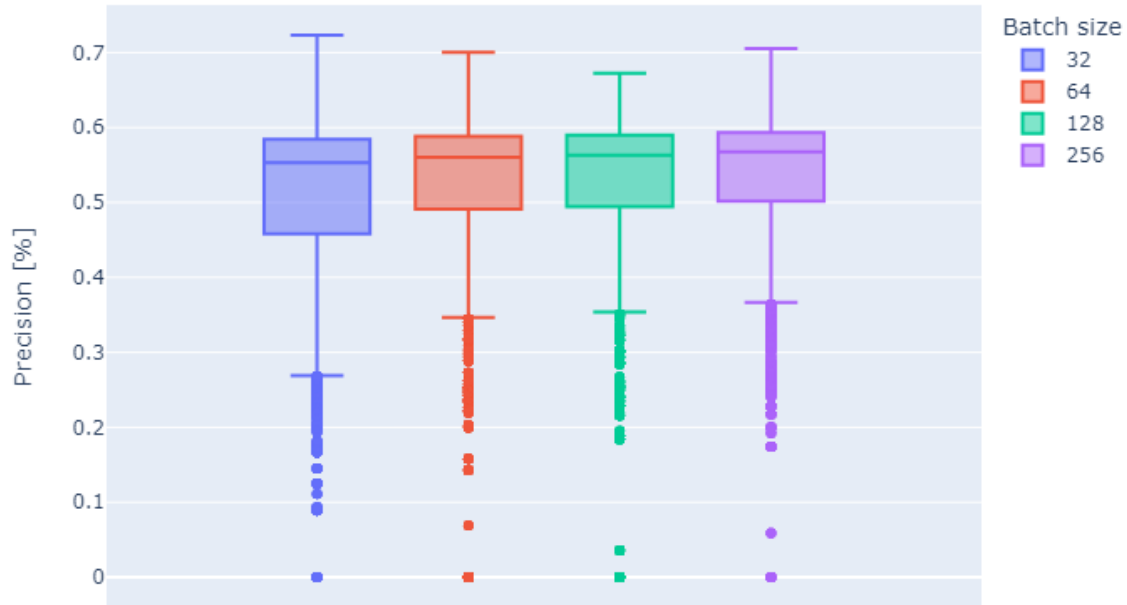


Figure 4.34: box plot of NN models with precision and batch size hyperparameter

4.7.1. Results of Neural Networks models

The results retrieved by the analysis performed on the NN hyperparameters, reported in the previous paragraph, lead to a decrease in the number of hyperparameter combinations. A subset of 240 combinations has been obtained and implemented. It is also important to underline how big these NNs are. In order to do so, the dimensions of the dataset are summarised as:

$$[0, n_{samples}]^{dimensions} \quad (4.3)$$

Dimensions is another term to indicate the number of features used to train the NN. In this case study, the dataset has a dimension equal to:

$$[0, 128\ 875]^{30}$$

The values of the hyperparameters chosen in order to form this subset of combinations are reported in Table 4.14.

Hyperparameter	Values
Number of layers and nodes	(50 × 3), (50 × 2), (50 × 1) (25 × 2), (17 × 3)
Activation functions	“ReLU”, “tanh”
Solver for weight optimization	“Adam”, “Sgd”
Scheduling strategy	“Constant”, “Adaptive”, “Invscaling”
Batch size	(32, 256)

Table 4.14: hyperparameter subset for NN classifiers

This subset takes a significant time to run, however further refinement cannot be carried on. The five best results from this implementation, concerning precision, are reported in Table 4.15.

<i>Number of layers and nodes</i>	<i>Activation functions</i>	<i>Solver for weight optimization</i>	<i>Scheduling strategy</i>	<i>Batch size</i>	<i>Accuracy [%]</i>	<i>Precision [%]</i>
(50, 50, 50)	“tanh”	“adam”	//	32	59.96	68.80
(17, 17, 17)	“tanh”	“adam”	//	32	66.11	65.85
(17, 17, 17)	“tanh”	“adam”	//	256	65.65	65.09
(50)	“ReLU”	“sgd”	“constant”	256	59.72	64.64
(50)	“ReLU”	“sgd”	“adaptive”	256	59.98	64.17

Table 4.15: 5 best NN models from selected subset of hyperparameter combinations

Cross validation

CV has been performed on the selected 240 NNs following the specifications described for previous models. Highest accuracy and precision obtained from the CV of these models are respectively equal to 76.85% and 74.56%, slightly higher than what expected.

5. Discussion and conclusions

Throughout this thesis, the implementation process of ML models for Instadose dosimeters validation has been described. In order to carry on this project, a deep understanding of radiation protection, dosimetry, and machine learning is fundamental. For this reason, chapter 1 provides to the reader a brief introduction of these concepts. Dosimetry represents a key aspect in radiation protection, since it is centered on measurement, assessment, and comprehension of radiation doses associated with exposure to ionizing or non-ionizing radiations. This aspect is crucial in ensuring the safety of individuals, populations, and environment.

Within the dosimetry domain, the role of ML algorithms is increasing, facilitating the development of new techniques along with emerging technologies. These permit the exploration of new paths and the enhancement of existing methodologies, thus contributing to the evolution of the field.

Inside this developing framework, ML algorithms have been implemented to mimic the validation process performed by radiation protection experts in SCK CEN. This thesis project aimed to validate the incoming dosimeter measurements in just two classes (well-performing dosimeters and dosimeters to be revised), hence performing a binary classification. To accomplish this task, it has been necessary to study in depth the latter process. Different manual validation sessions, carried on by different experts, have been supervised: this allowed the detection of subjective biases and slight differences within different expert validations. The supervision of this process also allowed to understand which were the main features on which the validation is based in order to consider them during the creation of the dataset.

Dataset creation process was mainly devoted to the merging of information retrieved by the Microsoft excel sheets, currently used for the manual validation, and from the SQL database, where the measurements are stored. Many challenges have been encountered during the creation of the previously mentioned dataset, mainly linked to information retrieving, merging and suitable preparation for the further adoption of ML methods. Python language programming has been extensively used for the creation of the final dataset, along with the Pandas library.

Once the dataset has been configured, dimensionality reduction and pre-processing techniques have been implemented. Dimensionality reduction aims to reduce the training time of ML models, deleting features in the database which do not add significant information. Dimensionality reduction techniques, for this project were mainly based on the use of Pearson's and Spearman's correlation matrices, as well as uncertainty coefficients and correlation ratios. Concerning the pre-processing measures, they are fundamental to retrieve reliable results from the model's implementation. They are constituted by class-balancing, cleaning and standardization measures of the database.

Following the dataset configuration, a deep study of the models to be implemented has been carried on in order to select proper values for the required hyperparameters concerning each model analysed. The ML methods chosen for this thesis project are: k-nearest neighbours, decision trees, random forests and neural networks. For each of them many hyperparameter combinations have been identified and implemented.

Results retrieved by these implementations have been reported and analysed in chapter 4, based on the concepts of accuracy and precision metrics. Thanks to the analysis of these preliminary results, a subset of hyperparameters have been identified for each model and the ML methods have been re-implemented. This is crucial for a future usage of these models, since it is not viable to implement all the analysed models (considering all hyperparameter combinations) in a practical scenario by the radioprotection experts, for each validation process. Therefore, for each method, the hyperparameter subsets which led to the best results, have been later selected in order to be the baseline on which the future dosimeters validation would be predicted.

Cross validation has been later on performed, only on the previously mentioned subsets, in order to assess the absence of overfitting phenomena and identify final precision and accuracy values for these models.

Following figures report a comparison between accuracies and precisions retrieved by the implementation of the selected subset of hyperparameter combinations with and without CV process. More in details, Figure 5.1 shows a comparison between the precisions obtained from these subsets of hyperparameters. Figure 5.2 shows the comparison, regarding accuracy results. On the other hand, Figure 5.3 and Figure 5.4 show, respectively, precision and accuracy results obtained after the cross validation.

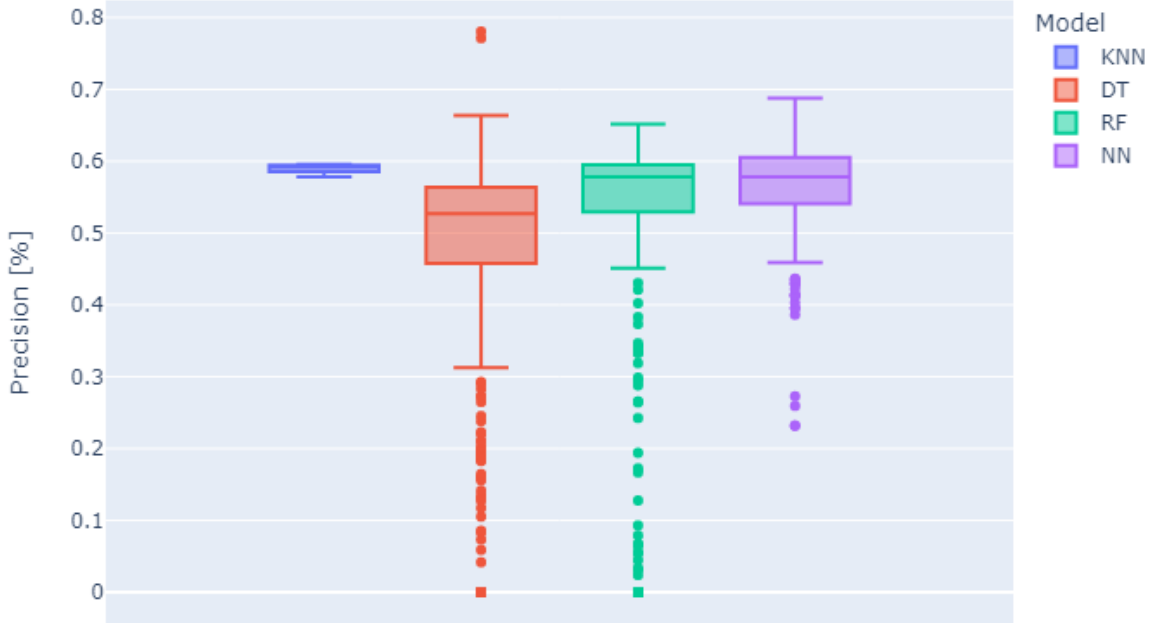


Figure 5.1: precision comparison between all the 4 ML models based on selected subset of hyperparameter combinations

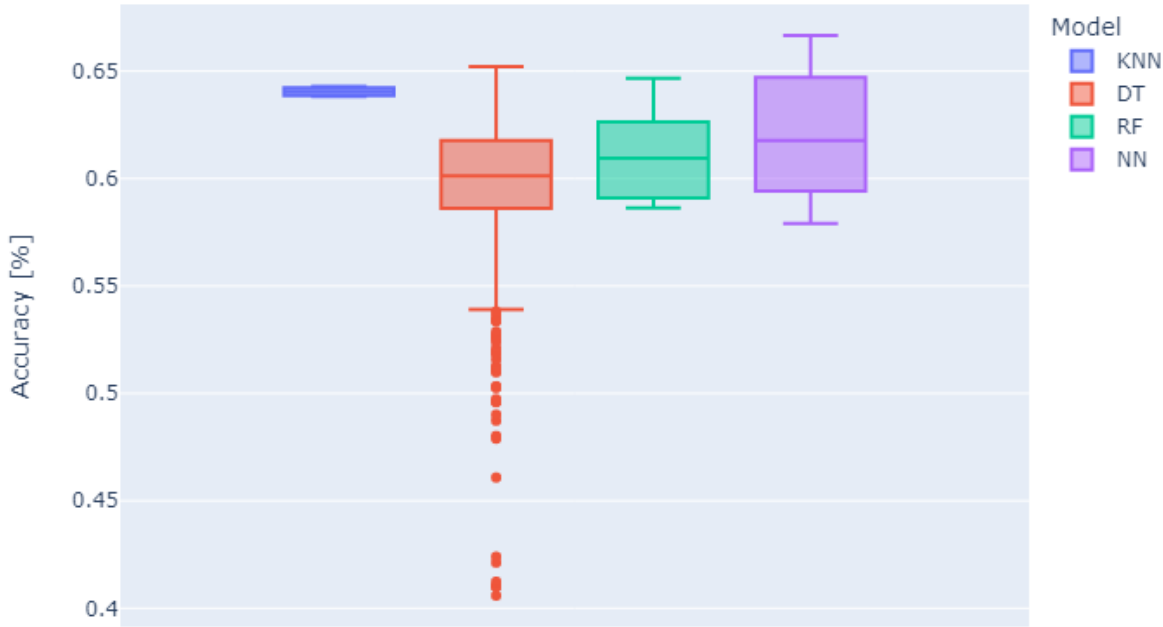


Figure 5.2: accuracy comparison between all the 4 ML models based on selected subset of hyperparameter combinations

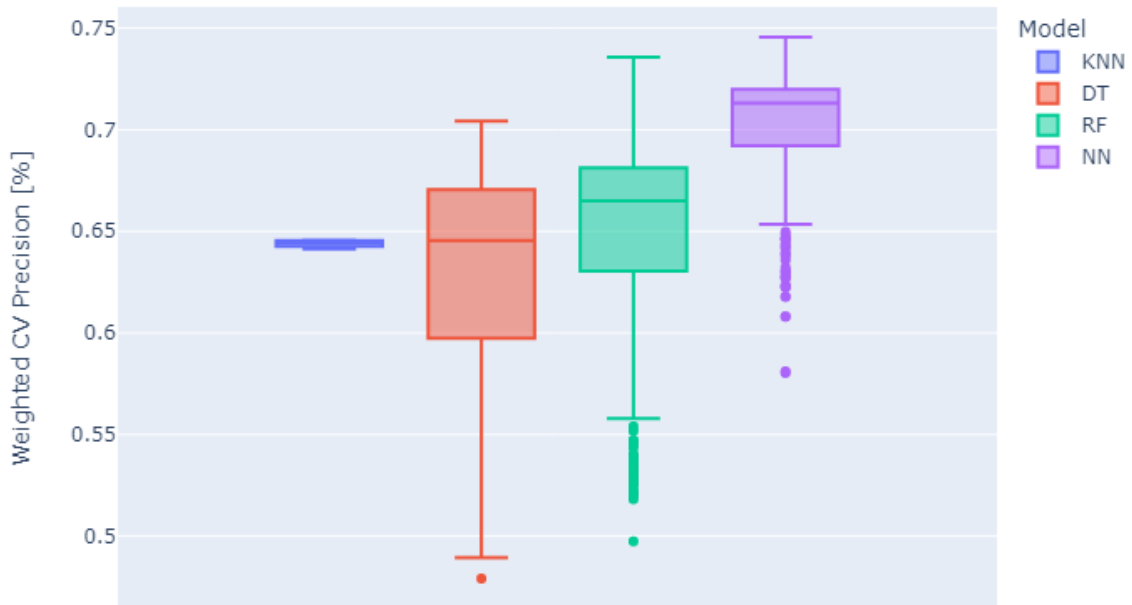


Figure 5.3: weighted CV precision comparison between all the 4 ML models based on selected subset of hyperparameter combinations

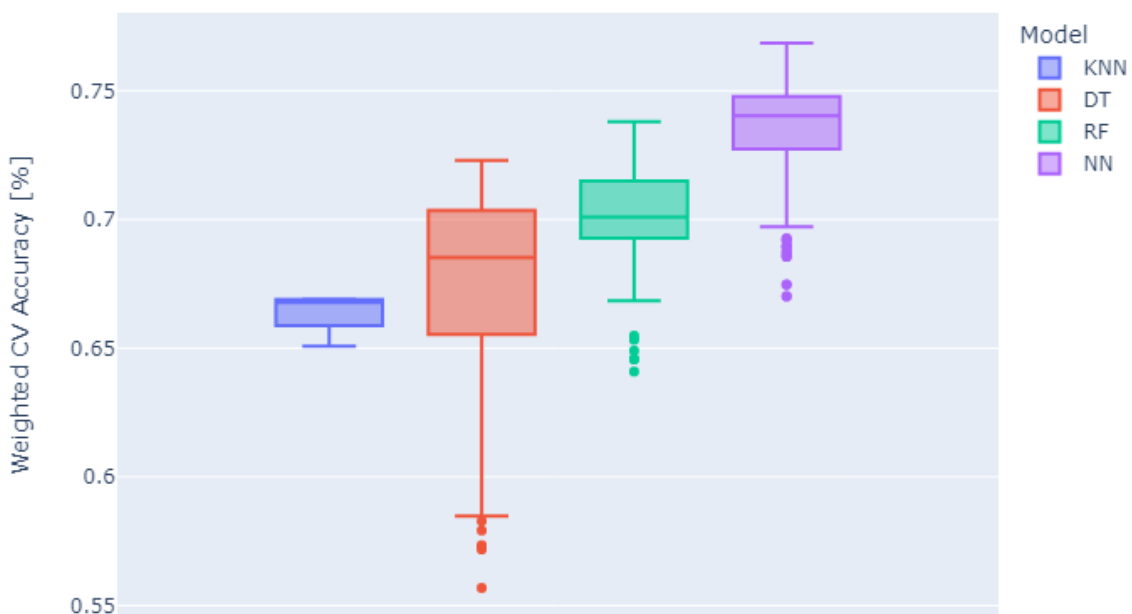


Figure 5.4: weighted CV accuracy comparison between all the 4 ML models based on selected subset of hyperparameter combinations

Concerning the results reported in the previous figures, it is straightforward to observe how the highest precision, without CV, is achieved by a DT model, as reported in Figure 5.1. However, applying CV, the method that reach better performances, in terms of highest precision and median value, is the NN, as depicted from Figure 5.3. Regarding accuracy, NNs show more accurate results, as reported in Figure 5.2 and Figure 5.4.

The overall implementation shows some differences among the models. Nevertheless, the majority of the selected models showed a precision, weighted overall the CV folds, that range from ~ 65% to ~ 75%. No models managed to reach higher precisions meaning that, with the current models and setting of database, no further information from these data can be retrieved to achieve better performances. Table 5.1 reports the classifiers with highest precisions, and relative accuracies, for each model type after CV implementation.

<i>Model</i>	<i>Weighted CV accuracy [%]</i>	<i>Weighted CV precision [%]</i>
<i>KNN</i>	66.68	64.57
<i>DT</i>	71.90	70.42
<i>RF</i>	69.45	73.58
<i>NN</i>	76.85	74.56

Table 5.1: highest precisions and relative accuracies for all 4 models after CV process

Several possibilities can be introduced in order to improve this thesis project. Starting from the dataset configuration, several choices that have been made could surely influence the models results. A further study implying different grouping methodology for soft reads can be carried on. Indeed, in this project soft reads have been grouped within two subsequent hard reads, and this approach could be changed, grouping the soft reads monthly. Another possible path that can be followed for the handling of soft reads concern the application of a different weight on these measurements. It is important to outline that this possibility regards the model implementation and not the dataset configuration.

In addition, principal component analysis can be implemented for further dimensionality reduction. Indeed, PCA identifies a set of principal components (orthogonal axis obtained by linear combinations of original data) that captures the maximum variance contained in the dataset. The number of features is reduced selecting just the principal components that contains the highest variance. Despite reducing the dimensions, PCA aims to maintain as much of the original information as possible. This is another approach that could be introduced as a future step of this project [33].

Other simplifications can also be applied to the dataset, getting rid of classes *H* and *Z*. Indeed, they can be identified without a ML model implementation, as done with class *I*.

Bibliography

- [1] P. Radvanyi and J. Villain, ‘The discovery of radioactivity’, *Comptes Rendus Phys.*, vol. 18, no. 9, pp. 544–550, 2017, doi: <https://doi.org/10.1016/j.crhy.2017.10.008>.
- [2] International Commission on Radiological Protection, ‘The 2007 Recommendations of the International Commission on Radiological Protection. ICRP Publication 103.’, *Ann. ICRP*, vol. 37, no. 2–4, 2007, doi: [10.1016/j.icrp.2007.10.001](https://doi.org/10.1016/j.icrp.2007.10.001).
- [3] International Atomic Energy Agency, *Occupational Radiation Protection*. in General Safety Guides, no. GSG-7. Vienna: IAEA, 2018. Available online: <https://www.iaea.org/publications/11113/occupational-radiation-protection>
- [4] ISR, ‘ISR History’. Available online: <https://isradiology.org/about-us/isr-history/> (Accessed: Dec. 12, 2023)
- [5] International Commission on Radiological Protection and Br. J. Radiol, ‘Recommendations of the International Commission on Radiological Protection’, *Ann. ICRP*, no. 1, p. 105, 1955, doi: [10.1016/S0074-27402880014-6](https://doi.org/10.1016/S0074-27402880014-6).
- [6] International Commission on Radiological Protection, ‘Recommendations of the International Commission on Radiological Protection. ICRP Publication 1.’, *Ann. ICRP*, no. 1, pp. iii–v, 1959, doi: [10.1016/S0074-27402880016-X](https://doi.org/10.1016/S0074-27402880016-X).
- [7] D. Kardamakis, S. Baatout, M. Bourguignon, N. Foray, and Y. Socol, ‘History of Radiation Biology’, in *Radiobiology Textbook*, Springer, 2023, pp. 1–24.
- [8] International Commission on Radiological Protection, ‘Recommendations of the International Commission on Radiological Protection. ICRP publication 26.’, *Ann. ICRP*, 1977.
- [9] M. Zucchetti, ‘Radiation protection and safety of nuclear plants’, Turin, 2022.
- [10] W. Jacobi, ‘The concept of the effective dose a proposal for the combination of organ doses’, *Radiat. Environ. Biophys.*, vol. 12, no. 2, pp. 101–109, Jun. 1975, doi: [10.1007/BF01328971](https://doi.org/10.1007/BF01328971).
- [11] International Commission on Radiation Units and Measurements, ‘Determination of Dose Equivalents Resulting from External Radiation Sources. ICRU Report 39.’, *Rep. Int. Comm. Radiat. Units Meas.*, vol. os-20, no. 2, pp. 1–1, 1985, doi: [10.1093/jicru_os20.2.1](https://doi.org/10.1093/jicru_os20.2.1).
- [12] International Commission on Radiation Units and Measurements *et al.*, ‘Determination of Dose Equivalents from External Radiation Sources – Part II. ICRU Report 43.’, *Rep. Int. Comm. Radiat. Units Meas.*, vol. os-22, no. 2, pp. 1–2, 1988, doi: [10.1093/jicru_os22.2.1](https://doi.org/10.1093/jicru_os22.2.1).
- [13] International Commission on Radiation Units and Measurements *et al.*, ‘Fundamental quantities and units for ionizing radiation. ICRU Report 85a.’, *Rep. Int. Comm. Radiat. Units Meas.*, vol. 11, no. 1, 2011, doi: [10.1093/jicru_ndr001](https://doi.org/10.1093/jicru_ndr001).
- [14] D. Andisco, S. Blanco, and A. E. Buzzi, ‘Dosimetry in Radiology; Dosimetría en radiología’, *Rev. Argent. Radiol. Online*, vol. 78, 2014. Available online: <https://www.osti.gov/etdweb/biblio/22331386> (Accessed: Mar. 12, 2024)
- [15] International Commission on Radiological Protection, A. Ulanovsky, D. Copplestone, and J. Vives i Batlle, ‘Dose coefficients for nonhuman biota environmentally exposed to radiation. ICRP Publication 136.’, *Ann. ICRP*, vol. 46, no. 2, 2017. Available online: <https://www.icrp.org/publication.asp?id=ICRP%20Publication%20136>
- [16] International Commission on Radiological Protection *et al.*, ‘Conversion Coefficients for Radiological Protection Quantities for External Radiation Exposures. ICRP Publication 116.’, *Ann. ICRP*, vol. 40, no. 2–5, 2010, Available online: <https://www.icrp.org/publication.asp?id=ICRP%20Publication%20116>

- [17] International Commission on Radiological Protection, '1990 Recommendations of the International Commission on Radiological Protection. ICRP Publication 60.', *Ann. ICRP*, vol. 21, no. 1–3, 1991, Available online: <https://www.icrp.org/publication.asp?id=ICRP%20Publication%2060>
- [18] M. Arthur, 'Dosimeter', US-2036072-A, Mar. 31, 1936
- [19] M. E. M. Abdelrahman, 'Personal dosimetry of workers without a physical dosimeter using computational methods', PhD Thesis, ULiège - Université de Liège, 2020.
- [20] F. Daniels, C. A. Boyd, and D. F. Saunders, 'Thermoluminescence as a Research Tool', *Science*, vol. 117, no. 3040, pp. 343–349, 1953, doi: 10.1126/science.117.3040.343.
- [21] International Commission on Radiological Protection, 'General Principles for the Radiation Protection of Workers. ICRP Publication 75.', *Ann. ICRP*, vol. 27, 1997, Available online: <https://www.icrp.org/publication.asp?id=ICRP%20Publication%2075>
- [22] M. G. Stabin, Ed., 'Radiation Protection Practice/Evaluation', in *Radiation Protection and Dosimetry*, New York, NY: Springer New York, 2007, pp. 244–308. doi: 10.1007/978-0-387-49983-3_11.
- [23] J. Akhter, M. Munir, K. Ahmad, M. M. Mahmood, and P. Akhter, 'Occupational exposure from external radiation used in medical practices in Pakistan by film badge dosimetry', *Radiat. Prot. Dosimetry*, vol. 140, pp. 396–401, Apr. 2010, doi: 10.1093/rpd/ncq134.
- [24] J. Shapiro, *Radiation protection: a guide for scientists and physicians*, Third. Cambridge, MA (US); Harvard Univ. Press, 1990. Available online: <https://www.osti.gov/biblio/5639436> (Accessed: Mar. 12, 2024)
- [25] E. G. Yukihara *et al.*, 'Luminescence dosimetry', *Nat. Rev. Methods Primer*, vol. 2, no. 1, p. 26, Apr. 2022, doi: 10.1038/s43586-022-00102-0.
- [26] K. D. Balliet, 'Analysis of Inverse-Square Law and Dose Rate Effects with LiF Thermoluminescent Dosimetry using 109 Cd X-rays', Jan. 2006. Available online: https://www.researchgate.net/profile/Kyle-Balliet-2/publication/317389336_Analysis_of_Inverse-Square_Law_and_Dose_Rate_Effects_with_LiF_Thermoluminescent_Dosimetry_using_109_Cd_X-rays/links/59388b18a6fdcc58ae617bfb/Analysis-of-Inverse-Square-Law-and-Dose-Rate-Effects-with-LiF-Thermoluminescent-Dosimetry-using-109-Cd-X-rays.pdf (Accessed: Mar. 12, 2024)
- [27] W. Huda and R. M. Slone, *Review of radiologic physics*, vol. 41. Lippincott Williams & Wilkins, 1996.
- [28] G. F. Knoll, *Radiation detection and measurement*. John Wiley & Sons, 2010. Available online: https://books.google.com/books?hl=it&lr=&id=4vTJ7UDeI5IC&oi=fnd&pg=PA1&dq=radiation+detection+and+measurement+knoll&ots=VyzODRDJaC&sig=_hOwaTbJuhTrzO3pbRzkosgMLHY (Accessed: Mar. 13, 2024)
- [29] C. Wernli, A. Fiechtner, and J. Kahilainen, 'Neutron dosimetry with ion chamber-based DIS system', *Radiat. Prot. Dosimetry*, vol. 66, no. 1–4, pp. 459–462, 1996.
- [30] A. Parisi, 'Setting up a new dosimetry system at SCK-CEN: testing the Instadose 2 dosimeter', 2015.
- [31] International Atomic Energy Agency, 'Assessment of Occupational Exposure due to External Radiation Sources - Active personal dosimeters',
- [32] R. Khadka, 'Machine Learning Types #2', Medium. Available online: <https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1> (Accessed: Dec. 21, 2023)

- [33] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An introduction to statistical learning: With applications in python*. Springer Nature, 2023. Available online: <https://books.google.com/books?hl=it&lr=&id=ygzJEAAAQBAJ&oi=fnd&pg=PR7&dq=an+introduction+to+statistical+learning+with+applications+in+python&ots=Wz12jPG1Tg&sig=k9IDDtZEMgDVHY9H9G2aYSfNyPI> (Accessed: Mar. 13, 2024)
- [34] J. Morimoto and F. Ponton, ‘Virtual reality in biology: could we become virtual naturalists?’, *Evol. Educ. Outreach*, vol. 14, May 2021, doi: 10.1186/s12052-021-00147-x.
- [35] P. Libin, ‘Introduction to Artificial Intelligence and Machine Learning for Biomedical Research Module 4: Reinforcement learning’,
- [36] D. P. Green and H. L. Kern, ‘Modeling heterogeneous treatment effects in large-scale experiments using Bayesian Additive Regression Trees’, in *The annual summer meeting of the society of political methodology*, 2010. Available online: https://www.researchgate.net/profile/Donald-Green-5/publication/242667854_Modeling_heterogeneous_treatment_effects_in_large-scale_experiments_using_Bayesian_Additive_Regression_Trees/links/0046352d68bdde5ceb000000/Modeling-heterogeneous-treatment-effects-in-large-scale-experiments-using-Bayesian-Additive-Regression-Trees.pdf (Accessed: Mar. 13, 2024)
- [37] H. Al-Behadili, K. Ku-Mahamud, and R. Sagban, ‘Rule pruning techniques in the ant-miner classification algorithm and its variants: A review’, presented at the 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2018. doi: 10.1109/ISCAIE.2018.8405448.
- [38] ‘Deep Learning: How Will It Change Healthcare? | OrboGraph’. Available online: <https://orbograph.com/deep-learning-how-will-it-change-healthcare/> (Accessed: Dec. 21, 2023)
- [39] ‘SCK CEN Instadose brochure’. SCK CEN. Available online: <https://www.sckcen.be/en/diensten/medische-toepassingen/persoonlijke-omgevingsdosimetrie>
- [40] ‘What is SQL? - Structured Query Language (SQL) Explained - AWS’, Amazon Web Services, Inc. Available online: <https://aws.amazon.com/what-is/sql/> (Accessed: Feb. 22, 2024)
- [41] ‘pandas - Python Data Analysis Library’. Available online: <https://pandas.pydata.org/> (Accessed: Feb. 28, 2024)
- [42] A. Stevenson, *Oxford dictionary of English*. Oxford University Press, USA, 2010. Available online: <https://books.google.com/books?hl=it&lr=&id=anecAQAAQBAJ&oi=fnd&pg=PR5&dq=oxford+languages+dictionary&ots=T0cC9pjHDT&sig=JkQF2vxHnsbE1rEnAc408X4hQFE> (Accessed: Feb. 28, 2024)
- [43] S. Ken, ‘Pearson’s correlation coefficient’, *Encyclopedia Britannica*, Feb. 2024, Available online: <https://www.britannica.com/topic/Pearsons-correlation-coefficient> (Accessed: Feb. 21, 2024)
- [44] F. Bertrand, ‘Sweetviz: A pandas-based library to visualize and compare datasets.’ Available online: <https://github.com/fbdesignpro/sweetviz> (Accessed: Feb. 21, 2024)
- [45] J. Hauke and T. Kossowski, ‘Comparison of values of Pearson’s and Spearman’s correlation coefficients on the same sets of data’, *Quaest. Geogr.*, vol. 30, no. 2, pp. 87–93, 2011.
- [46] F. Pedregosa *et al.*, ‘Scikit-learn: Machine Learning in Python’, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [47] A. Gholamy, V. Kreinovich, and O. Kosheleva, ‘Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation’, *Dep. Tech. Rep. CS*, Feb. 2018, Available online: https://scholarworks.utep.edu/cs_techrep/1209
- [48] V. C. Molina, P. Romojaro, and A. H. Solis, ‘ANICCA fuel cycle irradiation models: a machine learner predictor as a function of initial fuel composition’, Jul. 2022. Available online:

<https://researchportal.sckcen.be/en/publications/anicca-fuel-cycle-irradiation-models-a-machine-learner-predictor-> (Accessed: Mar. 07, 2024)

- [49] X. Ying, ‘An Overview of Overfitting and its Solutions’, *J. Phys. Conf. Ser.*, vol. 1168, no. 2, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.
- [50] D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek, ‘Scalable Global Optimization via Local Bayesian Optimization’. arXiv, Feb. 24, 2020. doi: 10.48550/arXiv.1910.01739.
- [51] R. Rodríguez-Pérez and J. Bajorath, ‘Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions’, *J. Comput. Aided Mol. Des.*, vol. 34, no. 10, pp. 1013–1026, Oct. 2020, doi: 10.1007/s10822-020-00314-0.
- [52] M. Cilimkovic, ‘Neural networks and back propagation algorithm’, *Inst. Technol. Blanchardstown Blanchardstown Road North Dublin*, vol. 15, no. 1, 2015, Available online: <https://www.academia.edu/download/51924347/NeuralNetworks.pdf> (Accessed: Mar. 13, 2024)
- [53] L. Tucci, ‘What is Machine Learning and How Does It Work? In-Depth Guide’, *Enterp. AI*. Available online: <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML> (Accessed: Mar. 07, 2024)
- [54] J. Brownlee, ‘A Gentle Introduction to the Rectified Linear Unit (ReLU)’, *MachineLearningMastery.com*. Available online: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (Accessed: Feb. 26, 2024)
- [55] Nik, ‘Tanh Activation Function for Deep Learning: A Complete Guide • datagy’, *datagy*. Available online: <https://datagy.io/tanh-activation-function/> (Accessed: Mar. 05, 2024)
- [56] A. Zaras, N. Passalis, and A. Tefas, ‘Chapter 2 - Neural networks and backpropagation’, in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds., Academic Press, 2022, pp. 17–34. doi: 10.1016/B978-0-32-385787-1.00007-5.
- [57] D. P. Kingma and J. Ba, ‘Adam: A Method for Stochastic Optimization’, Jan. 2017, doi: 10.48550/arXiv.1412.6980.
- [58] A. Devarakonda, M. Naumov, and M. Garland, ‘AdaBatch: Adaptive Batch Sizes for Training Deep Neural Networks’, Feb. 2018, doi: 10.48550/arXiv.1712.02029.
- [59] J. Brownlee, ‘A Gentle Introduction to Imbalanced Classification’, *MachineLearningMastery.com*. Available online: <https://machinelearningmastery.com/what-is-imbalanced-classification/> (Accessed: Mar. 06, 2024)