



A machine learning-based methodology for pulse classification in dual-phase xenon time projection chambers

P. Brás^{1,a}, F. Neves¹, A. Lindote¹, A. Cottle², R. Cabrita¹, E. Lopez Asamar¹, G. Pereira¹, C. Silva¹, V. Solovov¹, M. I. Lopes¹

¹ Laboratório de Instrumentação e Física Experimental de Partículas (LIP), University of Coimbra, 3004 516 Coimbra, Portugal

² Department of Physics, University of Oxford, Oxford OX1 3RH, UK

Received: 19 January 2022 / Accepted: 7 June 2022 / Published online: 24 June 2022
© The Author(s) 2022

Abstract Machine learning techniques are now well established in experimental particle physics, allowing detector data to be analyzed in new and unique ways. The identification of signals in particle observatories is an essential data processing task that can potentially be improved using such methods. This paper aims at exploring the benefits that a dedicated machine learning approach might provide to the classification of signals in dual-phase noble gas time projection chambers. A full methodology is presented, from exploratory data analysis using Gaussian mixture models and feature importance ranking to the construction of dedicated predictive models based on standard implementations of neural networks and random forests, validated using unlabeled simulated data from the LZ experiment as a proxy to real data. The global classification accuracy of the predictive models developed in this work is estimated to be $>99.0\%$, which is an improvement over conventional algorithms tested with similar data. The results from the clustering analysis were also used to identify anomalies in the data caused by miscalculated signal properties, showing that this methodology can also be used for data monitoring.

1 Introduction

Dual-phase noble element time projection chambers (TPCs) are excellent rare event observatories due to their low background, low energy threshold, good energy and position resolutions, and ability to scale their target mass [1]. These detectors have a long history in direct searches for dark matter (DM) in the form of Weakly Interacting Massive Particles (WIMPs) [2–6] and new, multi-tonne scale detectors using this technology are already or will soon start collecting data in the near future [7–9]. A dual-phase noble element TPC

consists of a scintillating liquid target volume (usually argon or xenon) in equilibrium with a gaseous layer on top, both observed by arrays of light sensors placed at the top of the gas phase and/or at the bottom of the liquid phase. An interaction with the target material in the TPC will excite and ionize some atoms at the interaction site, producing both scintillation light and ionization charge [1]. The scintillation light is promptly detected by the light sensors as the primary signal (S1). An electric field across the liquid phase drifts the ionization electrons that did not recombine towards the liquid-gas interface, where they are extracted to the gas phase and accelerated in a stronger field to produce a larger electroluminescence light signal (S2). The distribution of the S2 light over the light sensor arrays is used to reconstruct the position of the initial interaction in the horizontal plane, while the time between the S1 and S2 signals indicates the depth of the interaction. Furthermore, the relative size of these signals can be used to distinguish between electron recoils (ER) and nuclear recoils (NR) and infer the nature of the interacting particle [10].

In order to fully characterize an interaction in both position and energy, all signals resulting from the different processes in the detector must be correctly identified. This is usually performed by a specialized classification algorithm. Such classifiers need not only to correctly identify the main signals (S1 and S2) that characterize an event in a dual-phase TPC but also avoid misclassifying spurious signals as relevant ones. A high efficiency classifier is imperative in order to ensure that no event is misreconstructed and that rare signal events are accurately identified. Dual-phase noble element TPCs share the same overall pulse shape characteristics for a wide range of detector parameters, like electric field strength, light collection efficiency or the thermodynamic parameters of xenon/argon [11], and thus also share many of the same challenges related to signal identification.

Several techniques can be deployed to classify the signals recorded in these detectors, from simple, human-built,

^a e-mail: pbras@coimbra.lip.pt (corresponding author)

heuristic methods like decision trees [12,13] to advanced data analysis techniques such as machine learning (ML). It has been demonstrated that a classification efficiency for S1 and S2 signals above 90% is achievable using conventional heuristic methods [5,6], with some signal-level processing methods showing efficiencies $>98\%$ and $>99\%$ for small S1 and S2 signals, respectively [14]. The implementation of classification methods based on ML could improve this signal identification efficiency while also providing important information for the identification of spurious pulses. These methods have the advantage of handling high dimensional data more efficiently than conventional methods and can uncover unique and insightful information about intrinsic properties of the data. This work will explore a ML-based methodology for the development of high-accuracy and minimally-biased pulse classification tools tailored for dual-phase noble element TPCs.

There is an underlying risk in training a ML classifier model with non-representative data, most notably if the training is performed in a supervised manner. Supervised learning methods often resort to simulated data for training, and their results are therefore dependent on the accuracy of those simulations. This motivates the usage of unsupervised learning methods that do not rely on approximated models of the real data. In these approaches, the data features used for training are obtained directly from the target dataset, guaranteeing that all the available information and any possible data trends are present with no underlying hidden biases.

A particularly convenient set of tools used to process unlabeled datasets are clustering algorithms. Clustering analysis is very often combined with classification efforts because it naturally partitions the data based on its intrinsic properties in a robust, minimally-biased way [15]. In this work, a clustering analysis using Gaussian mixture models (GMMs) [16] is used to build two distinct predictive models designed for classification, those being random forests (RFs) [17,18] and an ensemble of neural networks (NN) [19]. This clustering method was chosen for its ability to handle large density inhomogeneities in the data [15]. The predictive models were selected due to their simplicity, robustness and excellent performance as classifiers, as well as their availability across many different ML software packages and implementations. The RF model was also selected due to its ability to extract information about the best discriminant features in the data [15].

Section 2 will give an overview of the data used in this analysis (unlabeled simulated data from the LUX-ZEPLIN (LZ) experiment [8,20]). Section 3 will present the methodology developed for processing the data and constructing classification tools based on ML: a clustering analysis is presented in Sect. 3.1, that will then be the basis for the development of predictive models based on RFs presented in Sect. 3.2 and a NN ensemble model presented in Sect. 3.3. A discus-

sion of the results and main issues can be found in Sect. 3.4. Section 4 will present some final remarks regarding the different models and an overview of the methodology developed in this work.

2 Overview of the data

This work was developed using LZ simulated data [20] from a simulated dataset created on October 2018 by the collaboration to validate its analysis tools. Simulated LZ data was chosen in order to keep this work relevant for current and next generation experiments searching for rare events over a wide range of energies. This data was also readily available and extensively tested. To ensure that the simulated data is analogous to real data, Monte Carlo truth information was not used for this study.

LZ will use a 7 tonne xenon TPC observed by 494 photomultiplier tubes (PMTs) distributed between two arrays (253 at the top and 241 at the bottom) [21]. An event is comprised of digitized waveforms originating from the readout of the TPC PMTs. The waveforms are digitized at 100 M samples per second and are only recorded when the voltage response of the PMT is above a threshold of 0.16 photoelectrons (PEs) [8]. These are then combined across all PMT channels to produce a summed waveform, which will contain several distinguishable structures that can be isolated in time and are expected to correlate to particular signals in the TPC. Each such structure will be referred to as a pulse. The event data is processed in a typical modular analysis approach, starting with pulse-level data processing and then performing event-level reconstruction: after some initial timing synchronization calibrations and baseline corrections, a dedicated pulse finding algorithm uses difference of Gaussian (DoG) filters to identify transients in the summed waveforms and isolate individual pulses in time [22]. The widths of the Gaussian filters are tuned to match the average frequency response of S1 and S2 pulses, and by setting thresholds in the responses of these filters the algorithm can identify the boundaries of a pulse (start and end times). Next, the parameterization algorithms calculate a set of pulse-shape and channel distribution quantities using both the summed and individual PMT channel waveforms within the pulse boundaries. These quantities are then used by the classifier algorithms to identify the physical origin of each individual pulse (S1, S2, etc). The quantities relevant for the pulse classification analysis explored in this work are presented in Sect. 2.2. After this pulse-level analysis, other dedicated algorithms use the full context of the event to characterize the interaction in the detector (in both energy and position).

Errors along the data processing chain, from the electronics readout to the pulse-level analysis algorithms, can produce spurious pulses that contaminate the data. The correct

identification and characterization of these spurious pulses is critical for the physics analysis, as well as to mitigate them at their origin.

2.1 Signals in a dual-phase noble element TPC

Figure 1 displays some examples of pulses in the dataset used that are expected to be found in a dual-phase noble element TPC. The main pulse types are the aforementioned S1 and S2, that are represented in Fig. 1a, b, respectively.

In the presence of an electric field, most of the scintillation light is produced by the decay of xenon excimers, which have fast decay times of 2.2 ns and 27 ns for transitions from singlet or triplet electronic excited states to the ground state, respectively [1]. Due to a combination of the pulse response of the PMT amplifier chain and S1 photon flight time, which can become greater than the decay times of the xenon excimers in large TPCs [23], a typical S1 pulse has a length in time of the order of 100 ns FWHM, rising quickly (~ 50 ns) and falling exponentially (~ 500 ns). The S1 signals will also have, on average, larger amplitudes in the bottom PMT channels due to the internal reflection of scintillation in the liquid-gas interface.

The S2 signal is proportional to the number of electrons extracted into the gas phase, with each electron producing hundreds of scintillation photons [1]. The shape in time of the S2 signal is dictated by the electron transit time across the gas gap in the extraction region and by the charge distribution of the electron cloud drifted from the interaction site to the liquid-gas interface [24]. The drifting electrons will experience diffusion (in both the longitudinal and transverse directions) along the travel path, making the S2 signals from interactions deeper in the TPC wider than those from interactions closer to the liquid-gas boundary. These effects result in S2 pulses with a length in time of the order of ~ 2 μ s FWHM, considering an extraction field of $\gtrsim 10$ kV/cm and a gas gap of 8 mm (electron transit time of 1.2 μ s) [8, 24]. Due to electron diffusion, the S2 pulses tend to have an almost symmetrical shape in time, akin to a normal distribution. Most of the S2 light is detected by the top PMT array due to its proximity to the liquid boundary and extraction region.

A pulse that results from the detection of a single photon by a PMT is called a single photoelectron (SPE). The smallest S1 pulses are composed of SPEs in a small set of PMT channels within a short time window. Random coincidence of PMT dark counts (spontaneous and spurious PMT pulses inherently indistinguishable from SPEs) can mimic small S1 pulses. To minimize this effect, a channel coincidence (i.e., the number of PMT channels that record signal within the time window of a pulse) of 3 or more is required for a pulse to be considered an S1 in LZ (3-fold coincidence) [21]. In order to preserve potential smaller S1 signals in low-energy analyses, the channel coincidence requirement for S1 pulses

was reduced to 2-fold in this work. This wider acceptance of pulses with lower channel coincidence can be reverted at a later analysis stage, if needed.

PMT afterpulses (APs) are typically caused by residual gas ionized by the electrons accelerated in the charge multiplication stages of the PMT. These ions are then drifted back to the photocathode where they release more electrons and produce a delayed signal at characteristic times [25, 26]. Figure 1c shows an S1 pulse followed by five APs. An example of a signal generated by the extraction of a single electron (SE) from the liquid into the gas phase is presented in Fig. 1d (left-most pulse) and in Fig. 1f (see discussion below for more details on the latter example). An S2 pulse is, in its essence, the overlap of several SE signals. The shape of SE pulses display great variability, with characteristic spike structures caused by the random nature of the detection of electroluminescence photons. As mentioned previously, the extraction of a single electron can produce hundreds of photons [1], with the average number of photons detected per extracted electron depending on the fixed strength of the extraction field, photon yield of the gas and light collection efficiency of the detector. Therefore, SE pulses have a very characteristic size that is used to calibrate the response of the detector [8]. Much like the channel coincidence criteria for S1 pulses, S2 pulses usually need to have areas several times larger than the average SE area in order to be considered valid, excluding false S2 pulses caused by random SE pileup. However, in order to preserve the classification of the smallest S2 pulses, the pileup of two or more SE pulses will be considered a valid S2 pulse in this work.

S2 pulses are generally followed by a tail of delayed electron emissions that can result in SE pileup and thus mimic an S2 signal, as seen in Fig. 1e. These delayed emissions typically produce an immediate continuous signal within some tens of μ s after the S2 signal (dubbed “S2 tail”) and a disperse trail of SE pulses that can last several ms after the S2 (“e-trains”) [27, 28]. These emission phenomena tend to scale with the size of the main S2 signal. The S2 tail is most likely the result of photoionization of impurities in the liquid bulk caused by xenon luminescence, and by photoelectric effect on the field grids. The e-trains and other SE delayed emissions with longer time scales are likely produced by the spontaneous emission of electrons trapped at the liquid boundary and by the capture and subsequent release of drifting electrons by impurities [27, 28].

Spurious pulses, i.e., pulses that do not directly correlate to real light-yielding processes in the TPC or that present unusual or unexpected properties, are the result of errors at any stage of data handling, from the electronics readout to the pulse-level analysis algorithms. These pulses often display similarities with either S1 or S2 pulses, contaminating the data and leading to misclassification errors. Figure 1f shows two SE pulses, with the leftmost being incorrectly split into

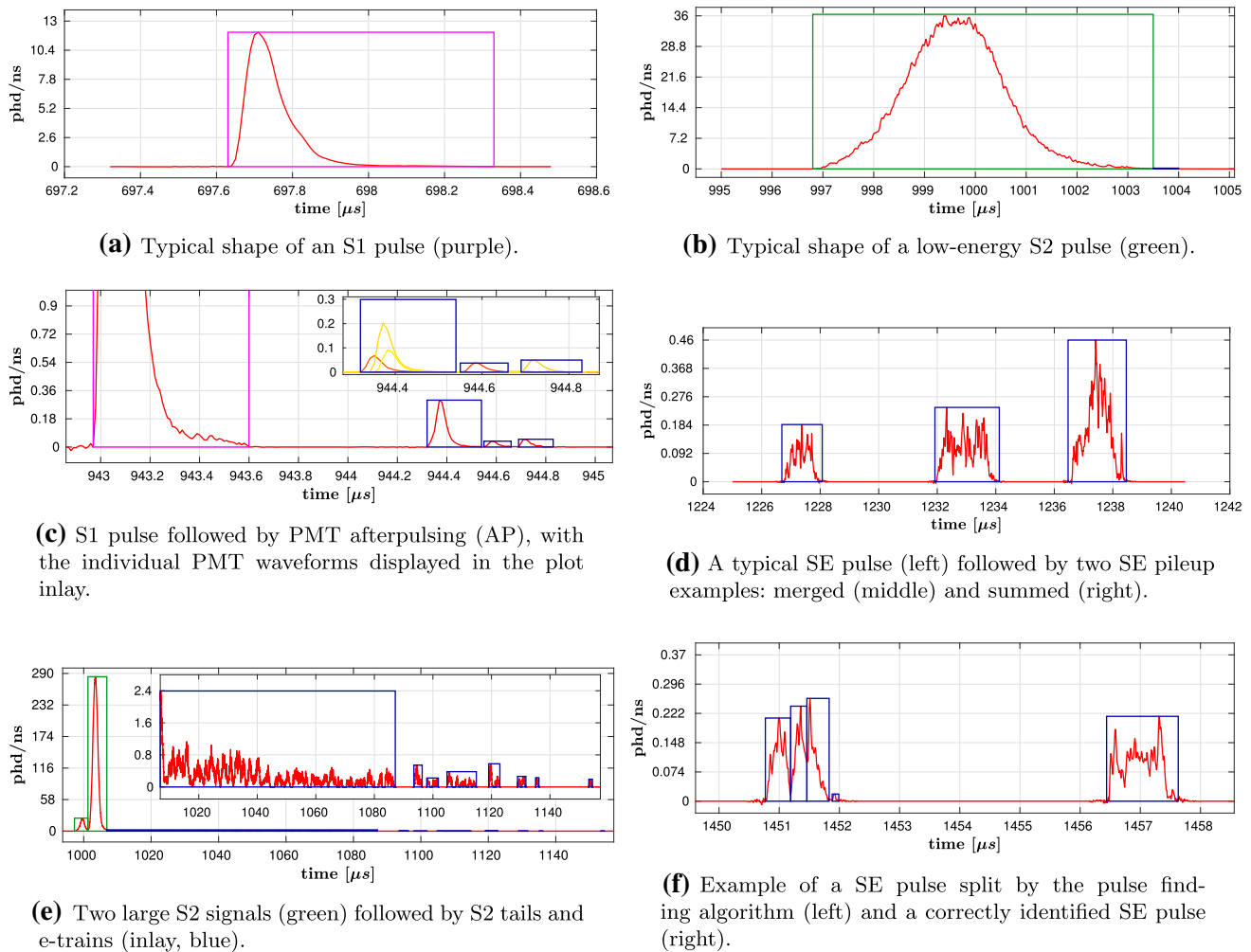


Fig. 1 Examples of typical pulses expected in dual-phase noble element TPCs, obtained from LZ simulated data. Pulse amplitudes are converted to units of photons detected (phd) per nanosecond. The coloured boxes represent the pulse boundaries calculated by a simple pulse finding algorithm, the purple and green boxes mark the main S1 and S2

several non-physical substructures by the pulse finding algorithm. These individual structures resemble S1 pulses in both shape and timing, and constitute a significant challenge for pulse-level classification in this dataset.

The main pulse classes that are usually considered in the low-level analysis of dual-phase noble-element TPCs are the aforementioned S1, S2, SE and SPE classes. Pulses that do not belong to any of the previously mentioned classes or that have non-physical properties (i.e., spurious pulses) are labeled as “Other” in this work.

2.2 Pulse features and data preprocessing

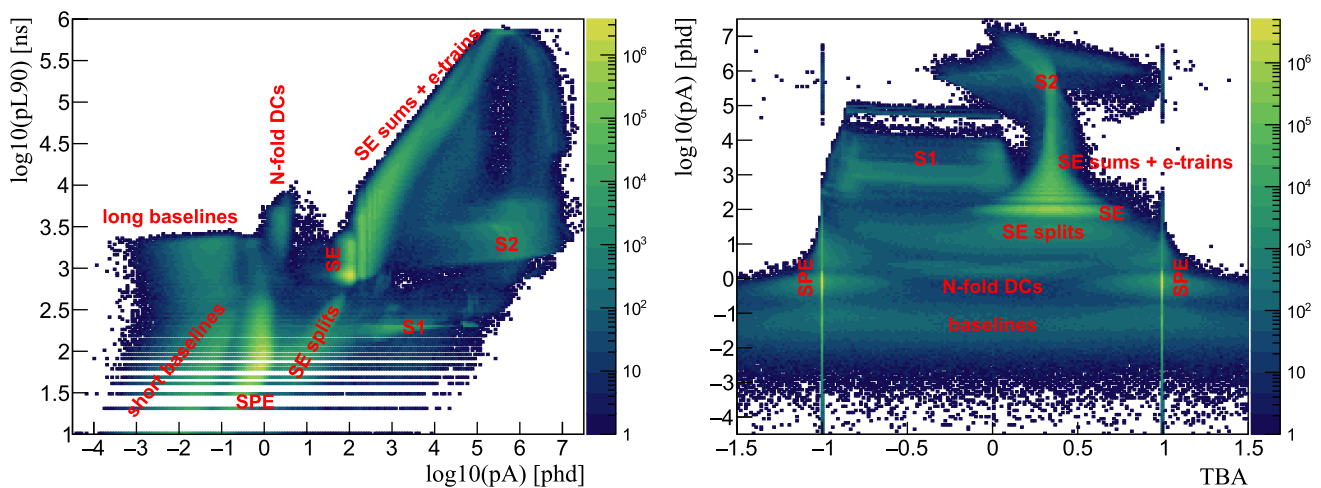
Pulses are identified by the classification algorithms based on their characteristic geometrical features. Some analyses use the raw shape of the pulse directly to identify the pulse (e.g.,

pulses of an event, respectively. The splitting of single electron (SE) pulses, like the example presented in **f**, is particularly challenging to the classification, since these spurious pulses have characteristics that are very similar to S1 pulses and PMT afterpulsing (AP)

using convolutional neural networks [29] or deep learning [30]) but typically an intermediate parameterization step is used to calculate relevant shape-related quantities, such as integrated areas, lengths, fit parameters and other detector-specific traits, that are then used by the classification algorithms. This work will follow the latter approach. Of the available pulse features in the LZ dataset used, dubbed “reduced quantities” (RQs), 11 were selected for this analysis and are summarized in Table 1. These RQs were selected based on past experience on heuristic pulse classification algorithms developed for the LUX and LZ experiments. The $pL90$ RQ provides a better estimation of the length of a pulse compared to the time difference between the pulse start and pulse end boundaries set by the pulse finder algorithm, since the latter length has larger variance for smaller pulses. The four prompt fraction RQs ($pF50$, $pF100$, $pF200$ and $pF1k$) are

Table 1 Some of the pulse-level RQs available in LZ simulated data chosen for the classification analysis in this work

Name [unit]	Type	Description
pA [phd]	Float	Total integrated area from the start to the end of the pulse
pH [phd/ns]	Float	Pulse maximum amplitude
pHTL	Float	Fraction of pulse length time at which the pulse reaches maximum amplitude
pL90 [ns]	Int	Pulse length time at 90% area, from 5 to 95% integrated area time
pRMSW [ns]	Int	Pulse root mean square (RMS) width
pF50	Float	Fraction of the pulse area integrated in a 50 ns time window starting 10 ns before the 5% integrated area time
pF100	Float	Same as <i>pF50</i> but for a 100 ns integration window
pF200	Float	Same as <i>pF50</i> but for a 200 ns integration window
pF1k	Float	Same as <i>pF50</i> but for a 1 μ s integration window
TBA	Float	Top-bottom asymmetry: difference between the top PMT area fraction and bottom PMT area fraction
Coincidence	Int	Number of PMT channels that record signal within pulse boundaries

**Fig. 2** Marginal distributions of the main pulse populations in pulse area (*pA*) vs pulse length at 90% area (*pL90*) space (left) and top-bottom asymmetry (*TBA*) vs *pA* space (right). Not all isolated populations are labeled, only the cases presented in Fig. 1 have been highlighted. The

populations labeled “baselines” consist of random noise that is isolated by the pulse finding algorithm. The population labeled “N-fold DC” contains spurious pulses caused by random coincidences of PMT dark counts (DCs) in *N* PMT channels

somewhat correlated, but they were included in this work in order to identify which one has the strongest discriminant power (Sect. 3.2.1).

A pre-selection of pulses was performed to ensure the quality of the dataset before the classification analysis, excluding pulses with non-physical RQ values, e.g., negative total areas, height or length. Pulses with *coincidence*=1 are also excluded *a priori* from this analysis since these can be automatically identified as either SPE or Other pulses, both of which are not critical for the description of the event. However, the value of the *coincidence* RQ is often overestimated in this dataset due to accidental partial overlap of pulses across different channels, especially for SPE pulses and afterpulsing. Even though these pulses are expected to be measured in a single PMT channel, negligible contributions from baseline fluctuations or small portions of partially overlapping pulses in other channels within the boundaries of the

pulse can lead to higher *coincidence* RQ values. This results in some SPEs and APs having *coincidence* > 1 and thus are eligible to be classified as S1 pulses, contaminating the RQ dataset. This, however, does not affect the final physics analyses since these distinctions are intentionally delayed to a dedicated event identification algorithm that uses the context provided by the remaining pulses in the event. This separation between pulse-level and event-level processing allows for the deployment of algorithms that are highly specialized for each task.

After the quality selections mentioned above, a total of 10^6 pulses are randomly selected from the remaining. The full dataset corresponds to an average of 25400 distinct events, corresponding to roughly 10 minutes and 35 seconds of exposure considering the expected average background event rate of 40 Hz for LZ [8]. Figure 2 shows a rough labeling of the main pulse populations in the dataset, obtained using tradi-

tional pulse identification methods and visual inspection. The goal of Fig. 2 is to act as a reference for the upcoming results in Sects. 3.1, 3.2 and 3.3. The populations are not fully separable in any 2-dimensional representation of pulse features but their general distributions can be intuitively inferred.

3 Methods

The following sections explain in detail the implementations and results of the ML methods explored in this analysis. A clustering analysis of the data is performed first, using Gaussian mixture models, and the results are then used to train two predictive models based in tree ensembles (RFs) and NNs to perform pulse classification and to extract useful information from the RQ dataset. These ML methods are extensively used in the physical sciences [31]. The potential of tree ensemble methods like RFs and boosted decision trees (BDTs) for data analysis in physics is well established, from searches for beyond the Standard Model physics at particle colliders [32,33] to dark matter direct detection experiments [6,34,35].

Much like the tree ensemble methods, NNs are extensively used in dark matter searches at colliders [33,36–38], direct detection experiments [39–41], cosmology and astrophysics surveys [42–44], and in other rare event searches [45,46]. However, this work will take a different approach to the traditional implementation of a single feed-forward dense NN by implementing instead an ensemble of NN classifiers trained individually to classify each valid pulse class in the data in a One-vs-All configuration [16]. Ensembles of NNs have been around for at least three decades [47–50], but the implementations are often based on generalizations of boosting or bagging strategies applied to NN models. In this work, the NN ensemble is instead composed of specialized binary classification models that work in parallel and whose results are combined to obtain a final classification.

Throughout this work the dataset will be partitioned into four distinct classes, corresponding to the class labels

$$\mathcal{S} = \{s_1, s_2, s_3, s_4\} = \{S1, S2, SE, \text{Other}\}.$$

The S1 pulse class will include 2-fold coincidence pulses in an effort to preserve the classification efficiency for low-energy searches. The distinction between S2 and SE pulses, despite both having the same physical origin, is expected to help the classification algorithms distinguishing between larger S1 pulses and the plethora of S2-like pulses, such as S2 tails and SE pileup, during training.

3.1 Clustering analysis

GMMs are a multi-component probabilistic distribution where a finite ensemble of Gaussian distributions are assumed

to originate the observed data [16]. The GMM implementation of the public python package *scikit-learn* [51] was used for the cluster analysis¹ considering $K = 67$ components and full covariance freedom between the different GMM components. The number of components of the mixture model was roughly estimated using an implementation of the kernel-based algorithm described in Reference [52]. This number was left purposefully larger than the number of categorical pulse classes being considered by the classifier module since each global pulse class contains several distinct pulse populations. This increases the efficiency of the clustering analysis by allowing a more “fine-grained” model that can account for the hidden stratification of the data [53]. Also, over-partitioning allows for a deeper understanding of the differences between seemingly degenerate populations, possibly allowing for the detection of outlier populations or pathological structures that were not identified in the preceding steps. Furthermore, after the clustering analysis, the different components of the mixture model can be collapsed into the same categorical classes corresponding to the pulse types expected in the data. This is a form of hierarchical processing that greatly accelerates learning on the subsequent classification algorithms [54]. However, some of the choices made in the assignment of pulse labels to the GMM components may not be ideal and will introduce biases in the following processing steps. As with many aspects of cluster analysis, and ML in general, a heuristic approach is somewhat inevitable [15].

Figure 3 displays the results of the GMM clustering of the RQ dataset, considering $K = 67$ components and using the following subset of RQs: pA , pH , $pL90$, $pF100$ and TBA . These RQs were selected based on their ubiquity in the analysis and with the goal of reducing the dimensionality of the data. Despite the probabilistic nature of the GMM algorithm, each pulse is categorically associated to the Gaussian component with the highest likelihood of having generated it [16]. Every GMM component is checked for spurious features in these and other marginal distributions and any strange population is handscanned, i.e., a sample of pulse waveforms are analyzed by eye, to determine their constituents. The first noticeable feature is the green population of S2-like pulses at high pA and trailing to higher $pL90$ values that clearly displays a pathological behavior of TBA (range outliers), with fluctuations that are larger than expected for regular S2 pulses (see Fig. 2). Investigation prompted by this observation determined that this issue was caused by a misreconstruction of the TBA RQ by the pulse parametrization algorithms. Another notable result from the GMM model is the identification of a distinct population of SE signals that are followed by either an afterpulsing or a coincident dark count, making them similar to small S2 pulses in the RQ space and resulting in a

¹ <https://gitlab.com/PauloBras/PClassGMMClustering.git>.

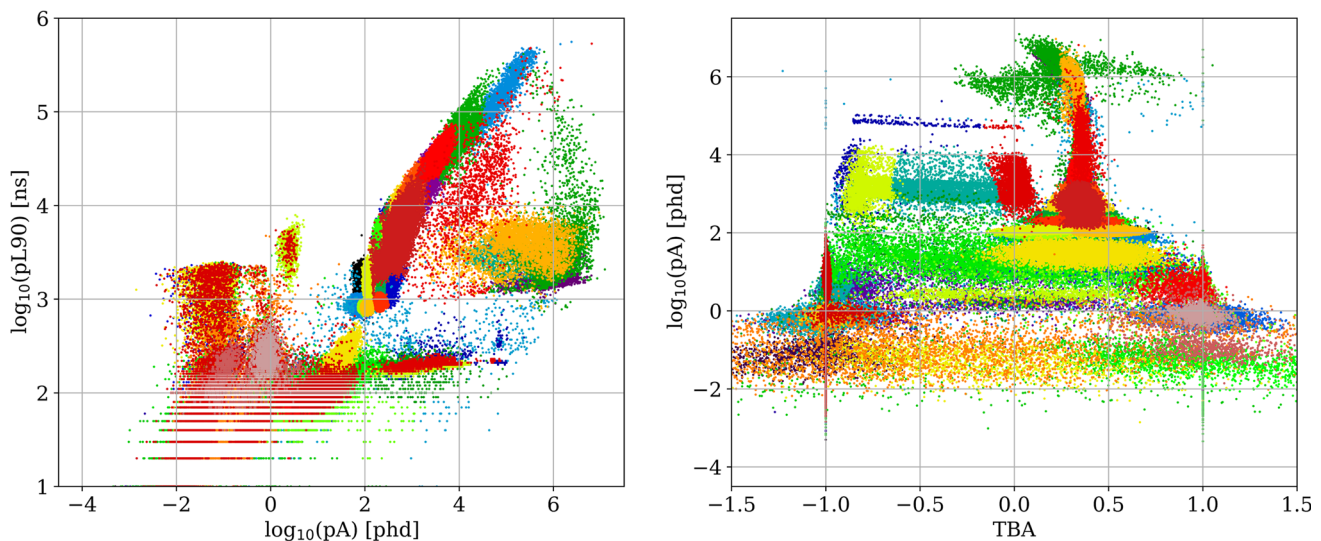


Fig. 3 Scatter plots of the GMM components for pulse area (pA) vs pulse length ($pL90$) RQs (left) and top-bottom asymmetry (TBA) vs pulse area RQs (right). The size and density of the 67 components vary significantly

small mixture between these signals in some GMM components. This, however, is inconsequential to the physics analysis since S2 and SE pulses can be discriminated by area alone in a subsequent processing step.

Each GMM component is now assigned a categorical pulse class based on the type of pulses that it contains. The contents of each component are inferred by carefully characterizing their parameter distributions and by handscanning a larger sample of pulses. Two GMM components were identified as containing most of the SE split population with low contamination from other pulse species. However, handscans of other GMM components seem to indicate that the clustering analysis cannot fully resolve afterpulsing and the contaminant SPE pulses from other spurious pulses like SE splits or baseline fluctuations. The components with noticeable contamination are assigned to the categorical pulse class in the majority. A discussion of the effect that these results and choices have in the training of the predictive models is provided in Sect. 3.4.

A rough classification of the data has already been performed by identifying the contents of each GMM component. Even though mixture models can be used as predictive models, alternative ML methods such as tree ensembles and neural networks are often better at generalizing and handling novel data. In the absence of labeled data, the results from the GMM clustering analysis can be used as a prior estimate of the pulse classes and were used to train the predictive models described in Sects. 3.2 and 3.3. The accuracy of these models will be calculated using the results obtained in this Section as targets, together with careful monitoring of their results via data handscans.

3.2 The RFClassifier pulse classification tool

The simplicity and robustness of random forests motivated the development of the RFClassifier² pulse classification tool. This tool has been developed using the *scikit-learn* implementation of the *RandomForestClassifier* model [51]. The methodology followed here aims to provide a deeper understanding of how to best separate bad pulse populations overlapping with the main populations and to determine which RQs, parametric thresholds, and sequences of selection criteria yield the most efficient partitioning of the dataset (i.e., feature importance ranking) [15, 18].

The RQs selected for benchmarking the model were pA , pH , $pL90$, $pRMSW$, $pF50$, $pF100$, $pF200$, $pF1k$, TBA and $pHTL$. The *coincidence* RQ was not selected due to being miscalculated for smaller pulses, as explained in Sect. 2.2. Despite being highly correlated, the four prompt fraction RQs ($pF50$, $pF100$, $pF200$ and $pF1k$) were included in the input data to determine which one has the strongest discriminant power among them.

The results obtained in the clustering analysis in Sect. 3.1 are used to train and benchmark the RFClassifier model. The number of trees in the model, their individual depth, and the minimal sample size to allow the data to be split, were estimated recursively by monitoring the performance of the model. The final model is composed of 101 learners with no limitation of growth and minimum number of samples to split a branch set at 2. The RQ dataset was divided into a training set and test set with 80–20% splitting ratio. No additional selection of data was performed and all classes are considered to have the same importance. It is worth noting that some

² <https://gitlab.com/PauloBras/PClassRFClassifier.git>.

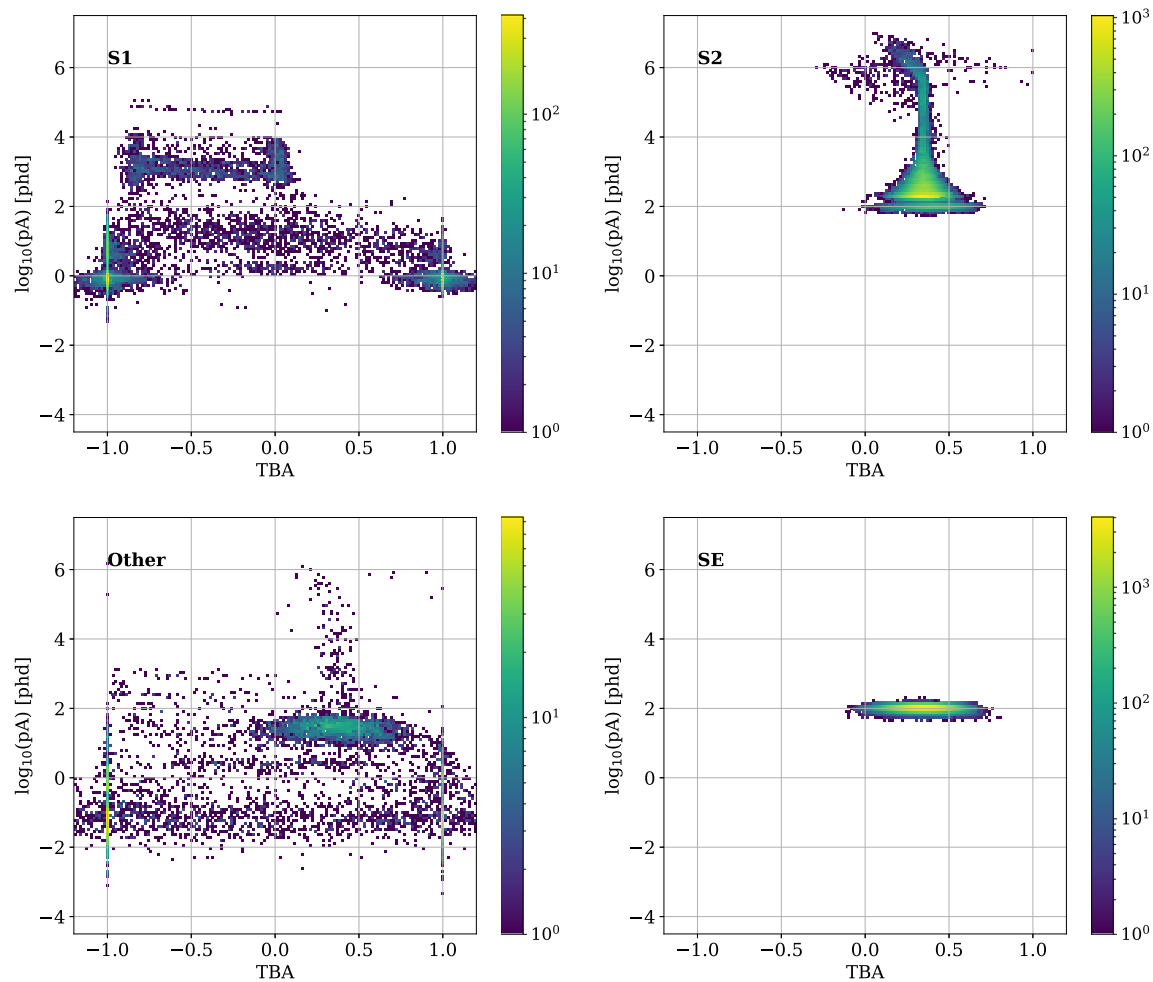


Fig. 4 Distribution of the pulse populations in the marginal distribution TBA vs pA after being processed by the RFClassifier. The top-left plot displays the distribution of pulses classified as S1. The top right plot

displays the distribution of S2 pulses, while the bottom plots display the Other population on the left and the SE population on the right

classes are more common than others in this dataset, namely SE pulses. Bootstrap aggregating (bagging) may induce some class bias if some classes are more represented than others in a multi-class dataset. This effect is amplified if bootstrapping is done without replacement, which is the case here [15, 18]. However, the asymmetry on the abundances of the different species is not too severe in this work and all classes are assumed to be sufficiently well represented, with the least prevalent class (S1 pulses, corresponding to around 10% of all pulses in the dataset) being around 2 orders of magnitude larger than the number of learners used.

Figures 4 and 5 display the results of the predictions of the RFClassifier model for the test dataset, represented in the marginal distributions $\log_{10}(pA)$ vs $\log_{10}(pL90)$ and TBA vs $\log_{10}(pA)$, respectively. These distributions can be compared to the plots in Fig. 2, that displays the distribution of all pulse populations conveniently labeled. The population of SE split pulses is clearly visible at the center of the bottom-left plot of

Fig. 4, that displays the distribution of the pulses classified as Other by the RFClassifier. This module successfully tagged most of the SE split pulses as Other, leaving only a small number of these spurious pulses in the distribution of pulses classified as S1, displayed on the top-left plot of Fig. 4.

Table 2 displays the confusion matrix of the RFClassifier for the test set of the RQ dataset, using the GMM results as the target class. The overall validation accuracy (acc) of the RFClassifier model, calculated using the GMM results as the target classification, is $acc = 99.38\%$. Considering only the classification of S1-like and S2-like pulses, i.e., not considering the mixing of S2 and SE pulse labels to be a misclassification, the validation accuracy is $acc^{S1S2} = 99.67\%$.

The main failure mode present in these results is the classification of Other pulses as S1-like pulses, mainly from SE splits. Conversely, some S1-like pulses are also being identified as Other pulses, hinting at some level of label mixing in the GMM results (see discussion in Sect. 3.4).

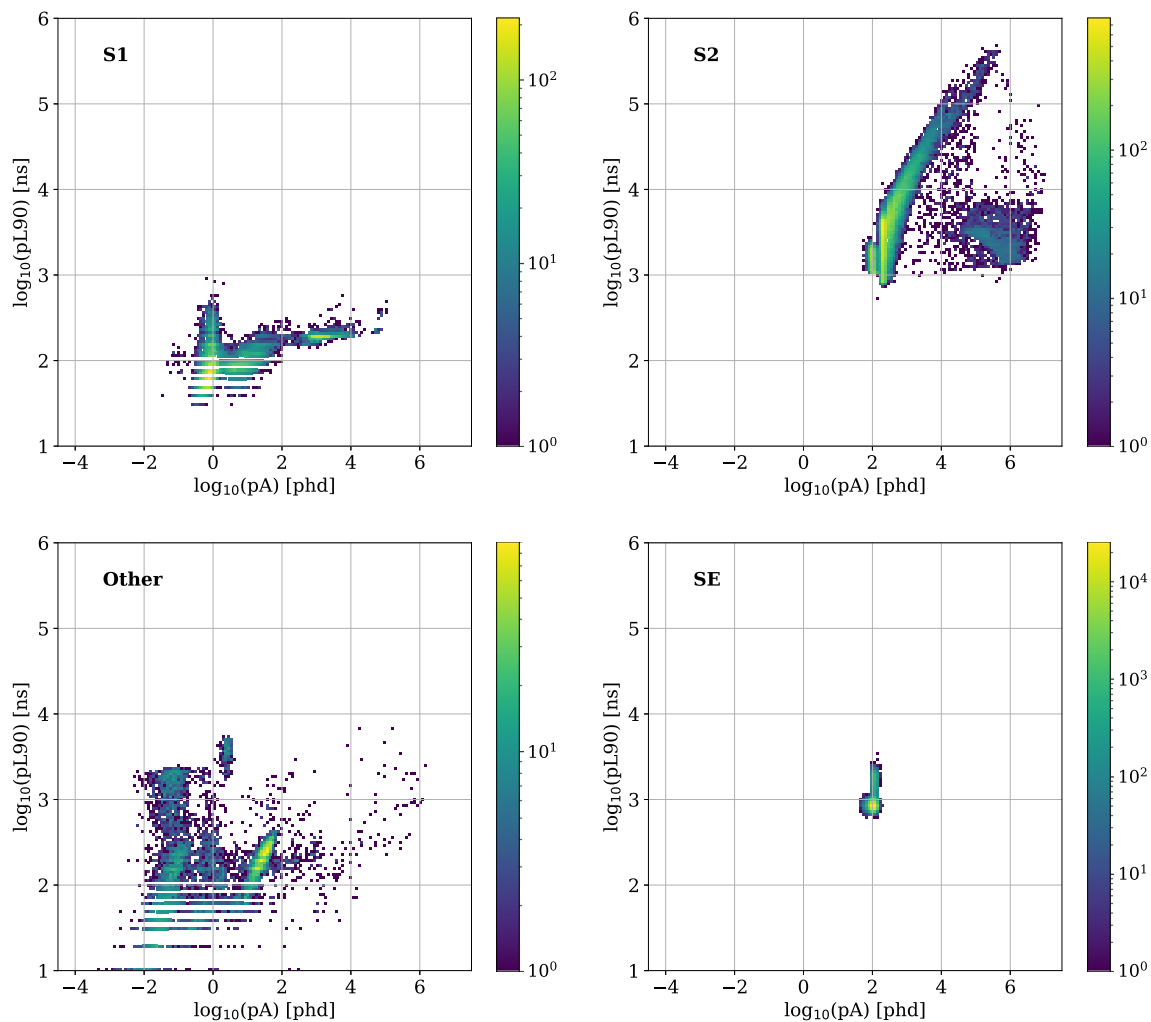


Fig. 5 Distribution of the pulse populations in the marginal distribution pA vs $pL90$ after being processed by the RFClassifier. The top-left plot displays the distribution of pulses classified as S1. The top right plot

displays the distribution of S2 pulses, while the bottom plots display the Other population on the left and the SE population on the right

3.2.1 Feature importance ranking

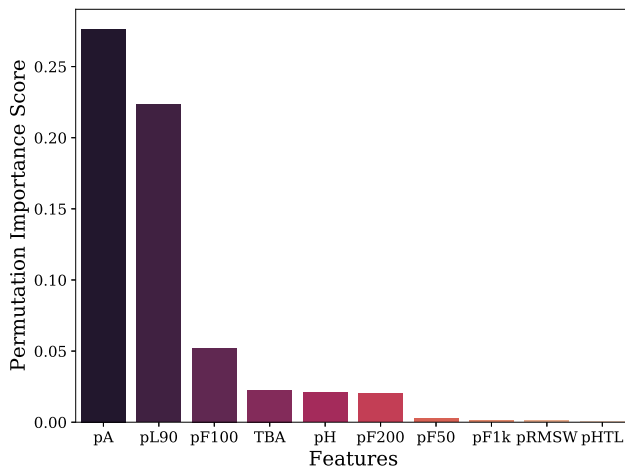
When working with classification of data with large dimensionality it is useful to rank the data features by their usefulness in partitioning the data. This allows ML models like those explored in this work to be trained with a smaller subset of features that ranked higher, and by doing so greatly improving training efficiency without compromising classification performance [15], assuming that this feature ranking is ubiquitous across different models. Even for data with lower dimensionality, the identification of the best discriminant features is an extremely useful step in the development of traditional heuristic classification algorithms, or in providing important information for the development of more efficient online data monitoring tools.

The permutation importance score method was used in this work to determine which RQs are the best overall discriminants for this dataset. This method evaluates the decrease in a model score when a single feature value is randomly shuffled, and is more reliable than the typical variable importance ranking methods based on impurity indices and variable frequency since it is less sensitive to highly correlated variables and to asymmetric representation of class labels [18,55].

Figure 6 shows the permutation importance score for each RQ considered in the analysis. The scores were obtained using the fully trained RFClassifier model. The pA and $pL90$ RQs are the best overall discriminants, with $pF100$, $pF200$, pH and TBA all roughly equal as the next strongest features. There is also an apparent preference for the $pF100$ RQ among the prompt fraction RQs. The $pF50$, $pF1k$, $pRMSW$ and $pHTL$

Table 2 Confusion matrix of the RFClassifier results over the test dataset, using the GMM results as the target labels

GMM class	RFClassifier predicted class				Total	
	S1	S2	SE	Other		
S1	11,590	0	0	261	11,851	5.9%
S2	0	51,103	348	4	51,455	25.7%
SE	0	228	128,371	0	128,599	64.4%
Other	385	13	5	7692	8095	4.0%
Total	11,975	51,344	128,724	7957	200,000	

**Fig. 6** Permutation importance ranking obtained with the RFClassifier

RQs scored the lowest on the permutation importance ranking.

The feature ranking provided by the RFClassifier was used to inform the training of the NN ensemble method presented in Sect. 3.3. The low dimensionality of the data used in this study, with only 10 independent features, means that training times are manageable and reducing the number of features will not impact it significantly. However, excluding the features with lower discrimination power increases the effectiveness of training and can help the model reach a higher accuracy. This was verified by comparing the accuracy of the same model trained with all features and with only the top 6 performing ones. For that reason, the *pF50*, *pF1k*, *pRMSW* and *pHTL* RQs were excluded during the training of the NN ensemble method presented in Sect. 3.3.

These results were also used to inform other non-ML classification algorithms that were proposed for LZ.

3.3 TriNet classifier

It was established during this work that an ensemble of dense NNs as binary classifiers, each trained in a One-vs-All configuration, returned better overall results than a single multi-class NN classifier model. In a One-vs-All problem, each of

the NNs focuses on learning how to distinguish a single class from the remaining.

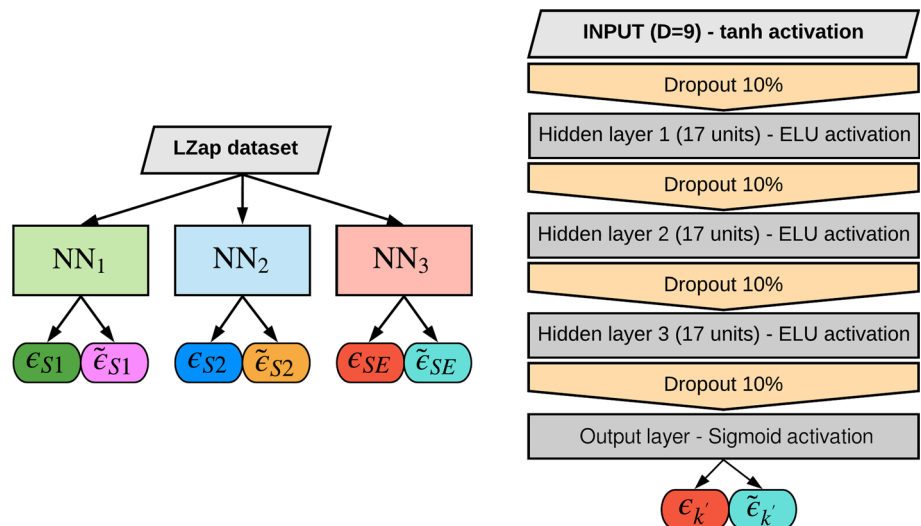
The NN ensemble model developed in this work combines several independent learners, each trained separately to identify a single given categorical pulse class. The prediction of the ensemble of K' NN classifiers is represented by a function $\hat{y} = f(\mathbf{x})$, with $\hat{y} = \{\hat{y}_1; \hat{y}_2; \dots; \hat{y}_{k'}; \dots; \hat{y}_{K'}\}$. The terms $\hat{y}_{k'}$ are the output vectors of the k^{th} NN classifier in the ensemble, labeled here as $\text{NN}_{k'}$. These terms can be defined as independent functions such that $\hat{y}_{k'} = f_{k'}(\mathbf{x})$, with $f_{k'}$ being the function fitted by $\text{NN}_{k'}$. The terms $\hat{y}_{k'}$ are two-vectors of the form $\hat{y}_{k'} = (\epsilon_{k'}, \tilde{\epsilon}_{k'})$. The component $\hat{y}_{k'}^1 = \epsilon_{k'}$ is the “response” of $\text{NN}_{k'}$, and should approach 1 when a pulse is of the class designated to the NN. Conversely, the term $\hat{y}_{k'}^2 = \tilde{\epsilon}_{k'}$ represents the response of $\text{NN}_{k'}$ for all the other classes, here named the “anti-response”. Since the response and anti-response are anti-correlated, only the response is used in this work. The output of each NN is continuous and bounded, and can be viewed as the likelihood of a given pulse being from the respective target class. Therefore, this method provides a quasi-probabilistic approach to the classification problem. Obtaining probabilistic information instead of a categorical result can provide more information about the nature of the pulse and the degree of ambiguity of the prediction.

In this work the Other pulses are considered the “exception” class to the remaining target pulse classes and thus they do not need to be learned explicitly by the NN ensemble model. For that reason, the number of target classes with a dedicated NN classifier in the ensemble is $K' = 3$, those classes being S1, S2 and SE, as mentioned at the end of Sect. 3. Therefore, this model was named *TriNet Classifier*,³ and was implemented and trained using the Keras library [56] with a Tensorflow backend.

The architecture of the TriNet Classifier is represented in the left side of Fig. 7. The ensemble model is composed of three independently trained NN classifiers schematically depicted in the right side of Fig. 7. Each NN classifier has $L = 3$ fully connected hidden layers, with $P = 17$ hidden units (neurons) each for a total of $h = 51$ neurons. These

³ <https://gitlab.com/PauloBras/PClassTriNetClassifier.git>.

Fig. 7 Simplified schematic of the TriNet Classifier model (left). The model is composed of an ensemble of three NN models (right) each trained to separate the three main pulse classes in the data: S1, S2 and SE pulses. The outputs of each NN classifier, $\hat{y}_{k'} = \{\epsilon_{k'}, \tilde{\epsilon}_{k'}\}$, represent the response and anti-response of the classifier to its respective class



values were selected by benchmarking the performance of the individual models for different values of L and P . All hidden neurons have an exponential linear activation (ELU) [56]. The input layer and all hidden layers have an associated dropout layer that will randomly shut down neurons with a 10% probability per neuron in each training batch in order to prevent overtraining [19,56].

The output of the TriNet Classifier is given by the ensemble of the individual outputs of each NN classifier, $\hat{y} = \{\hat{y}_1; \hat{y}_2; \hat{y}_3\}$, with $\hat{y}_{k'} = (\epsilon_{k'}, \tilde{\epsilon}_{k'})$ the output of the individual NN classifier assigned to class label k' . For simplicity, the output \hat{y} is explicitly written as a matrix of the type

$$\hat{y} = \begin{bmatrix} \epsilon_1 & \tilde{\epsilon}_1 \\ \epsilon_2 & \tilde{\epsilon}_2 \\ \epsilon_3 & \tilde{\epsilon}_3 \end{bmatrix}$$

with the row vectors representing the individual outputs of the $NN_{k'}$ classifiers and the column vectors representing the overall response and anti-response of the ensemble, hereby defined by $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)^T$ and $\tilde{\epsilon} = (\tilde{\epsilon}_1, \tilde{\epsilon}_2, \tilde{\epsilon}_3)^T$, respectively. Since the sum of the elements of each individual output $\hat{y}_{k'}$ is approximately unity, the sum of all elements of the TriNet Classifier output is approximately K' ($=3$ in this work), i.e.,

$$\sum_{k'=1}^{K'} (\epsilon_{k'} + \tilde{\epsilon}_{k'}) \approx K'. \quad (1)$$

For each of the pulse classes present in the data, the output \hat{y} of the TriNet Classifier ensemble is expected to be asymptotically equivalent to

$$f(\mathbf{x} | y = 1) \sim \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix};$$

$$f(\mathbf{x} | y = 2) \sim \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad (2)$$

$$f(\mathbf{x} | y = 3) \sim \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

with the training label $y = k'$ representing the pulse class $s_{k'} \in \{S1, S2, SE\}$.

The responses of the individual classifiers of the TriNet ensemble to a pulse that resembles neither an S1 nor an S2 nor an SE pulse, i.e., an Other pulse, are expected to be small, i.e., the output of the TriNet Classifier model associated to an Other pulses is expected to be asymptotically equivalent to

$$f(\mathbf{x} | y \neq \{1, 2, 3\}) \sim \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \quad (3)$$

It is convenient to quantify the global strength of the responses ϵ and anti-responses $\tilde{\epsilon}$ in order to evaluate if there is any type of ambiguity on the predictions. The *confidence* on the result, Γ_ϵ , can be expressed by the sum of the elements of ϵ , and an equivalent quantity can be defined for $\tilde{\epsilon}$, designated by $\Gamma_{\tilde{\epsilon}}$.

$$\Gamma_\epsilon = \sum_{k'=1}^{K'} \epsilon_{k'} \quad (\text{confidence}) \quad (4)$$

$$\Gamma_{\tilde{\epsilon}} = \sum_{k'=1}^{K'} \tilde{\epsilon}_{k'} \quad (\text{confidence complement}) \quad (5)$$

The parameter $\Gamma_{\tilde{\epsilon}}$ can be seen as the complement to the confidence score Γ_ϵ , since the result in Eq. (1) implies that $\Gamma_{\tilde{\epsilon}} \approx K' - \Gamma_\epsilon$.

The output of the TriNet model can be converted into a probabilistic vector, $p(k)$, using a simple set of rules using the response vector ϵ and the confidence score Γ_ϵ . Since there is no representation of spurious pulses in training, the probability vector should have $K' + 1$ number of terms, one for each K' primary class plus an additional term assigned to the remaining classes. In the case of the TriNet classifier, the $K' = 3$ primary classes will return $K' + 1 = 4$ probability results. For $K' = 3$, the simplest parametrization of the output of the model is given by Eqs. (6) and (7), for when the confidence Γ_ϵ is less than or equal to 1 or when its larger than 1, respectively. The latter case is dubbed “overconfidence” and automatically indicates that there is some conflict between the responses of the classifiers, which can be the result of some pulses having mixed properties or hinting towards problems in the data.

1. If $\Gamma_\epsilon \leq 1$:

$$\begin{aligned} p(k \in K') &= \epsilon_{k'} \\ p(k = 4) &= 1 - \Gamma_\epsilon \quad (\text{Other}) \end{aligned} \quad (6)$$

2. If $\Gamma_\epsilon > 1$ (overconfidence):

$$\begin{aligned} p(k = 4) &= \frac{1}{2} (\Gamma_\epsilon - 1) \quad (\text{Other}) \\ p(k \in K') &= \epsilon_{k'} \frac{1 - p(k = 4)}{\Gamma_\epsilon} \end{aligned} \quad (7)$$

Using the information obtained by feature importance ranking with the RFClassifier (see Sect. 3.2.1), the following list of RQs were selected as input: pA , pH , $pL90$, $pF100$, $pF200$, and TBA . These inputs were normalized and mean-centered when necessary to cover similar ranges, in order to avoid loss of information and neuron “death” [19]. The mean-centering of the pA and $pL90$ RQs is done with respect to the mean values for SE pulses since these are very well defined in the data and provide a natural middle point for the S1-like and S2-like phase-spaces. The pH , pA and $pL90$ RQs were projected to a logarithmic representation since they span several orders of magnitude. The TBA RQ and the prompt fraction family of RQs do not need normalization.

The dataset was divided into a training set and test set with 80–20% splitting ratio. The training is optimized using the *RMSprop* algorithm [56] with an initial learning rate $\alpha = 0.001$ and batch size $n = 128$ samples. The choice of the hyperparameters of the model was performed *a-priori* by monitoring its generalization accuracy, loss over the generalization data and average training time. The training is monitored using the accuracy score of each model over the validation dataset, with an early stopping of the training if the validation accuracy does not improve for 10 consecutive

epochs. To avoid overfitting, the model is saved at the state it was at the beginning of the 10 final training epochs.

The class labels used to train the TriNet Classifier were obtained using the GMM clustering analysis explained in Sect. 3.1. As mentioned before, there is some degree of mixture between SE split pulses and S1 afterpulsing in the GMM results, which is expected to influence the performance of the TriNet Classifier in some way. If the mixture is not too severe, each NN is expected to overcome the label impurity under the *Classification Without Labels* (CWoLA) paradigm [57]. However, the contamination of SPE pulses with *coincidence* larger than 1 seems to be uniformly distributed across the S1 and Other pulse classes in the GMM results, which may lead to some confusion during training (see discussion below).

In order to represent the pulse populations for different classes obtained with the TriNet Classifier, the \hat{y} output, and in turn the probabilistic vector of the pulse classes obtained using Eqs. (6) and (7), are converted to categorical class labels corresponding to the 4 possible pulse classes considered in this work. Here, the categorical classifications are obtained by setting threshold values to the elements of the probabilistic vector, with each threshold being tuned to minimize the misclassification errors (false positives and false negatives) for its respective class. This conversion step can be achieved in many different ways, and the one used in this work is not assumed to be optimal.

Figures 8 and 9 display the results of the predictions of the TriNet classifier model for the test dataset after converting them into categorical classifications, represented in the marginal distributions $\log_{10}(pA)$ vs $\log_{10}(pL90)$ and TBA vs $\log_{10}(pA)$, respectively. These distributions can be compared to the plots in Fig. 2, as well as with the results obtained with the RFClassifier model in Figs. 4 and 5. The TriNet model is able to separate the main pulses classes with high efficiency, comparable to the results from the RFClassifier model presented in Sect. 3.2. Similarly to those results, the TriNet classifier model managed to separate the majority of the SE split pulses, clearly visible at the center of the bottom-left plot of Fig. 8, that displays the distribution of the pulses classified as Other. The classification of some SE pulses with larger pulse length as S2 pulses is a consequence of both models being trained with the results from the GMM clustering analysis, discussed in Sect. 3.1. The sub-population of SPE pulses with *coincidence* RQ larger than 2 (easily identified by having $TBA \approx \pm 1$) is present in both the distribution of pulses classified as S1 (top left plot of Fig. 8) and in the distribution of pulses classified as Other (bottom left plot of Fig. 8). This indicates that some mixing in the classification is taking place as mentioned before. A more detailed discussion is presented in Sect. 3.4. Finally, it has been demonstrated that the TriNet model is able to identify spurious pulses without having explicitly learned from examples of these pulses during training, which may provide a strong method to identify

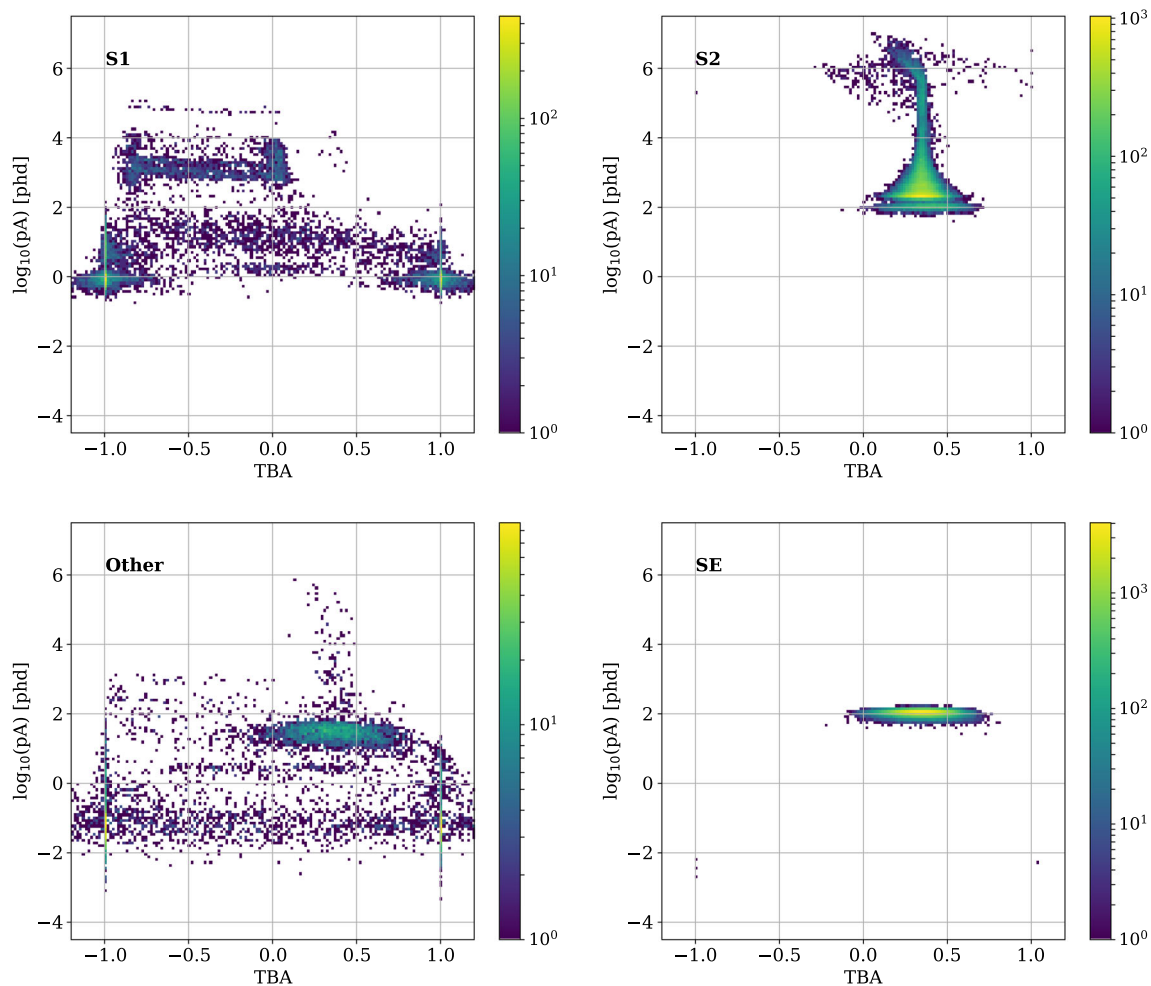


Fig. 8 Distribution of the pulse populations in the marginal distribution TBA vs pA after being processed by the TriNet Classifier trained with the GMM results, and converted to categorical classifications. Top-

left: pulses classified as S1. Top right: pulses classified as S2. Bottom-right: pulses classified as SE. Bottom-left: distribution of the remaining pulses, i.e., classified as Other

unexpected pathological behavior in the data by looking at pulses classified as Other.

Table 3 compares the results of the TriNet Classifier with the labels obtained in the GMM clustering analysis developed in Sect. 3.1. From Table 3, the overall validation accuracy of the TriNet Classifier model, calculated over the GMM labels, is estimated to be $acc = 99.06\%$. Considering only the classification of S1-like and S2-like pulses, i.e., not considering the mixing of S2 and SE pulse labels to be a misclassification, the validation accuracy becomes $acc^{S1S2} = 99.47\%$. Similarly to the RFClassifier results in Sect. 3.2, the main contribution to the accuracy loss of the TriNet model is the confusion between S1 pulses and Other pulses. This issue is discussed in detail in the next section.

3.4 Discussion

The RFClassifier and the TriNet models show some tension with the results from the GMM clustering, especially in the

mixing of S1 and Other labels. A detailed handscan focused on these cases was performed in order to determine their cause. This exercise indicated that the major contributions to the loss of accuracy on both the TriNet and RFClassifier models are the misclassification of SPE pulses with incorrect channel coincidence values, and a strong confusion between S1 afterpulsing and the SE split pulses.

Roughly 55% of the pulses labeled as S1 in the GMM analysis, and 43% of those labeled as Other, have coincidence lower than 3. The majority of these cases are SPE pulses with miscalculated coincidence, with a minor contribution from random noise (baselines). From the handscan, it was determined that these cases contribute to 64% of the pulses misclassified as S1 and 24% of those misclassified as Other by the TriNet model. If these pulses are excluded, the accuracy of the TriNet model increases up to 99.3%.

The confusion between S1 afterpulsing and the SE split pulses seems to be caused by some mixture on the GMM analysis results, which influenced the TriNet and RFClassi-

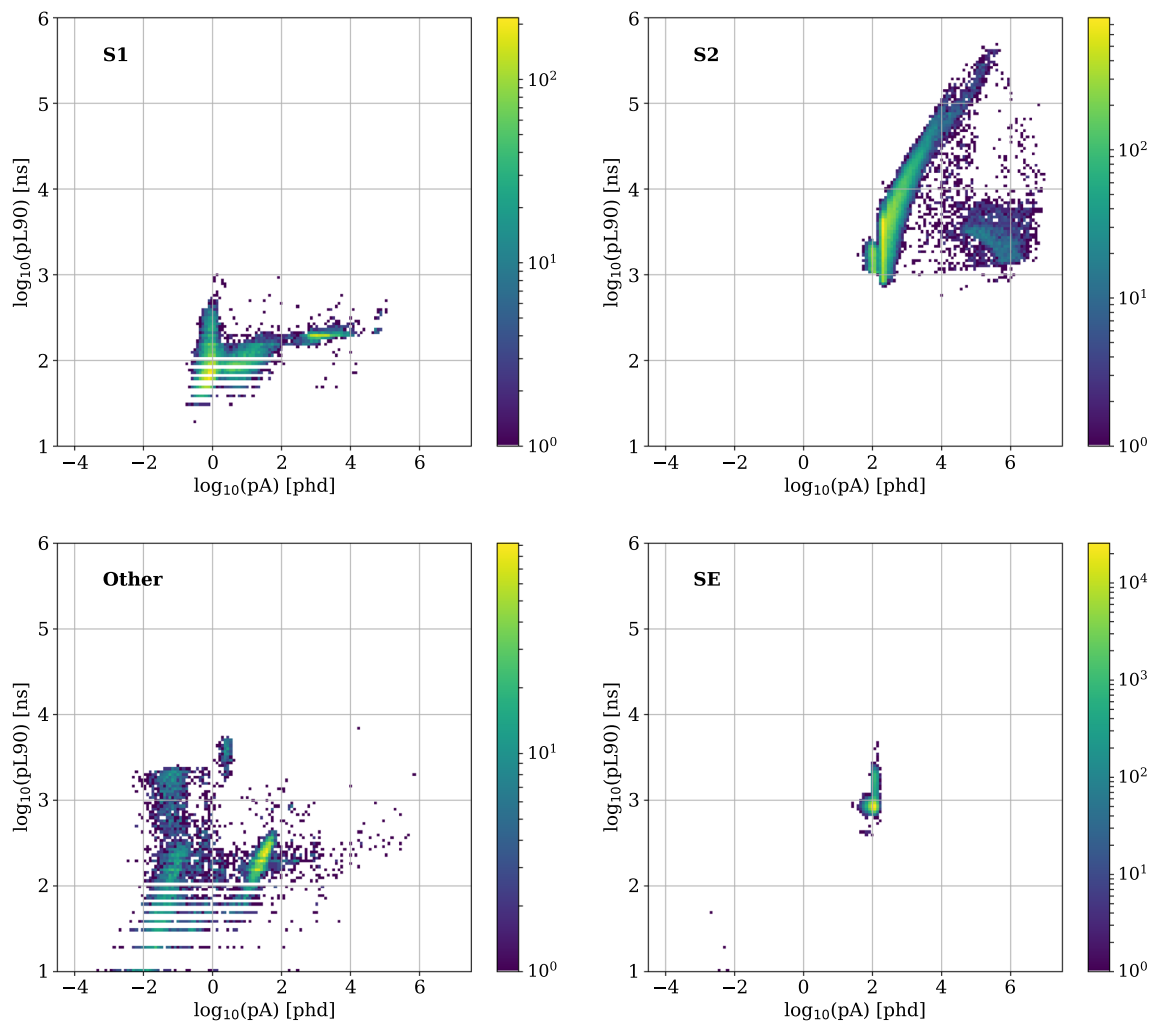


Fig. 9 Distribution of the pulse populations in the marginal distribution pA vs $pL90$ after being processed by the TriNet Classifier trained with the GMM results, and converted to categorical classifications. Top-

left: pulses classified as S1. Top right: pulses classified as S2. Bottom-right: pulses classified as SE. Bottom-left: distribution of the remaining pulses, i.e., classified as Other

Table 3 Confusion matrix of the results from the TriNet Classifier, trained with the GMM results as labels, compared with the GMM results obtained in Sect. 3.1

GMM class	TriNet predicted class				Total	
	S1	S2	SE	Other		
S1	11,571	0	0	280	11,851	5.9%
S2	0	51,001	444	10	51,455	25.7%
SE	0	380	128,211	8	128,599	64.4%
Other	698	38	28	7331	8095	4.0%
Total	12,269	51,419	128,683	7629	200,000	

fier models. The degree of confusion indicates that the label mixture for these two pulse types is high, as mentioned previously in Sect. 3.1, which most likely prevented the models from fully learning to distinguish these two pulse types [57]. Very often these pulses share the same RQ phase-space and have very similar waveforms. In fact, it became apparent during the handscan that discriminating by eye between some

SE split pulses and S1 afterpulsing was remarkably difficult without some context from the rest of the event. Considering that the classification is performed at the pulse level without any information from the rest of the event (and only in the RQ space) it seems reasonable that the GMM analysis, and consequently the predictive models, could not fully separate these two populations.

Most of the classification issues mentioned above only involve pulses that are not critical to the reconstruction of an event, while the accuracy for identifying the primary S1 and S2 pulses correctly – the pulses that are essential to fully describe an interaction – is expected to be much higher than the values of the global accuracy presented here. For this reason, the impact of these misclassifications on the performance of the detector is expected to be negligible.

4 Conclusions

The methodology presented here demonstrates how one can develop new or improve existing algorithms aimed at the classification of signals in dual-phase noble element TPCs using standard ML methods. Even though this work was performed using simulated data, the omission of truth information and the realism of the simulations used ensure that this data is analogous to real detector data, which indicates that this analysis can accomplish the same results and overcome similar challenges when applied to real data.

The initial clustering analysis and the feature importance ranking methods presented here provide vital information, directly from detector data, that can then be used to train dedicated ML predictive models capable of competing with common *ad hoc* and heuristic methods. The GMM clustering analysis explored here provides a robust and minimally biased way of characterizing detector data directly, in order to train pulse classification algorithms while avoiding any dependencies on simulations. This technique will be tested with real detector data once LZ starts data taking.

The RFClassifier model achieved an overall classification accuracy of 99.38% and was able to successfully tag known spurious pulses in the data. The TriNet model also achieved an excellent global accuracy of 99.04% and demonstrated that it is possible to teach an ensemble of NNs to identify spurious pulses while maintaining a high global classification accuracy, even if the spurious pulses are not explicitly learned. Both models, when used as standalone pulse classification tools, perform at least on par with conventional methods [5, 6, 14] having also outperformed a conventional classification algorithm developed and tested by the LZ collaboration with an overall accuracy of 98.6%, estimated via handscans, for the same data used in this work. It was also demonstrated that these models can be used as auxiliary tools to identify systemic issues in the data processing stages preceding the classification. The TriNet Classifier model is also able to deliver probabilistic information, providing this classifier with a higher classification flexibility than the remaining methods studied here.

The results presented in this work were obtained whilst training the classification models with impure data. This contamination of the training labels originated from the GMM

clustering analysis, which in turn resulted from errors in preceding data handling stages, e.g., miscalculated *coincidence* RQ or split pulses. These uncertainties in the data are to be expected during the first stages of development of a processing framework or when a detector begins taking data. Despite these challenges, the methods developed in this work outperformed traditional methods without resorting to *a priori* knowledge of the data, such as truth information from simulations during validation. Thus, this methodology is able to effectively overcome these potential sources of bias. Furthermore, these methods can be used to diagnose potential problems in preceding data handling stages, which has been demonstrated in this work with some level of success. In particular the GMM clustering analysis has revealed issues with the *TBA* RQ calculation by isolating the corresponding outlier population. This property can help to identify potential issues in real data as well.

Finally, besides the challenges contemplated in this work (mainly caused by data handling stages between data acquisition and processing) additional problems are to be expected when analysing real detector data (from LZ or otherwise). However, since this methodology uses a data-driven approach, it is expected to overcome any data-specific challenges with at least the same level of success as that of traditional methods, regardless of the nature of these challenges.

Acknowledgements The authors thank the LUX-ZEPLIN Collaboration for access to the simulated data used in this work, obtained using the BACCARAT simulation package, the Detector Electronics Response (DER) simulation package and the LZap data processing package.

Funding The funding has been received from Fundação para a Ciência e a Tecnologia with Grant nos. PD/BD/114114/2015, POCI/01-0145-FEDER-029147, PTDC/FIS-PAR/2831/2020, PTDC/FIS-PAR/29147/2017.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The data that support the findings of this study are available from the LZ Collaboration but restrictions apply to the availability of these data, which were used under licence for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of the LZ Collaboration.]

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended

use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funded by SCOAP³.

References

1. V. Chepel, H. Araujo, *JINST* **8**, R04001 (2013). <https://doi.org/10.1088/1748-0221/8/04/R04001>
2. D.Y. Akimov et al., *Astropart. Phys.* **27**, 46–60 (2007). <https://doi.org/10.1016/j.astropartphys.2006.09.005>
3. D.S. Akerib et al. [LUX], *Phys. Rev. Lett.* **118**(2), 021303 (2017). <https://doi.org/10.1103/PhysRevLett.118.021303>
4. P. Agnes et al. [DarkSide], *Phys. Rev. Lett.* **121**(8), 081307 (2018). <https://doi.org/10.1103/PhysRevLett.121.081307>
5. E. Aprile et al. [XENON], *Phys. Rev. Lett.* **121**(11), 111302 (2018). <https://doi.org/10.1103/PhysRevLett.121.111302>
6. Q. Wang et al. [PandaX-II], *Chin. Phys. C* **44**(12), 125001 (2020). <https://doi.org/10.1088/1674-1137/abb658>
7. J. Aalbers et al. [DARWIN], *JCAP* **11**, 017 (2016). <https://doi.org/10.1088/1475-7516/2016/11/017>
8. D.S. Akerib et al. [LZ], *Nucl. Instrum. Meth. A* **953**, 163047 (2020). <https://doi.org/10.1016/j.nima.2019.163047>
9. E. Aprile et al. [XENON], *JCAP* **11**, 031 (2020). <https://doi.org/10.1088/1475-7516/2020/11/031>
10. D.S. Akerib et al. [LUX], *Phys. Rev. D* **102**(11), 112002 (2020). <https://doi.org/10.1103/PhysRevD.102.112002>
11. D. Gonzalez-Diaz, F. Monrabal, S. Murphy, *Nucl. Instrum. Meth. A* **878**, 200–255 (2018). <https://doi.org/10.1016/j.nima.2017.09.024>
12. D.S. Akerib et al. [LUX], *Phys. Rev. D* **97**(10), 102008 (2018). <https://doi.org/10.1103/PhysRevD.97.102008>
13. E. Aprile et al. [XENON], *Eur. Phys. J. C* **77**(12), 881 (2017). <https://doi.org/10.1140/epjc/s10052-017-5326-3>
14. D.S. Akerib et al. [LUX], *Phys. Rev. Lett.* **112**, 091303 (2014). <https://doi.org/10.1103/PhysRevLett.112.091303>
15. T. Hastie, R. Tibshirani, J. Friedman, *Elements of Statistical Learning*, 2nd edn. (Springer-Verlag, New York, 2009). <https://doi.org/10.1007/978-0-387-84858-7>
16. C.M. Bishop, *Pattern recognition and machine learning*, (Springer New York, 2006) ISBN:978-0-387-31073-2
17. T. K. Ho, *Proceedings of 3rd international conference on document analysis and recognition*. **v1**, 278–282 (1995). <https://doi.org/10.1109/ICDAR.1995.598994>
18. L. Breiman, *Mach. Learn.* **45**, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
19. C.C. Aggarwal, *Neural networks and deep learning*, vol.1, XXIII, 497 (Springer, Cham, 2018). <https://doi.org/10.1007/978-3-319-94463-0>
20. D.S. Akerib et al. [LUX-ZEPLIN], *Astropart. Phys.* **125**, 102480 (2021). <https://doi.org/10.1016/j.astropartphys.2020.102480>
21. D.S. Akerib et al. [LUX-ZEPLIN], *Phys. Rev. D* **101**(5), 052002 (2020). <https://doi.org/10.1103/PhysRevD.101.052002>
22. H.A.E. El-Sennary et al., *Am. J. Comput. Sci. Technol.* **2**(3), 35–47 (2019). <https://doi.org/10.11648/j.ajcst.20190203.11>
23. M. Moongweluwan [LUX], *JINST* **11**(02), C02036 (2016). <https://doi.org/10.1088/1748-0221/11/02/C02036>
24. B.J. Mount et al. *arXiv:1703.09144* [physics.ins-det] (2017)
25. P.A. Amaudruz et al. [DEAP], *Nucl. Instrum. Meth. A* **922**, 373–384 (2019). <https://doi.org/10.1016/j.nima.2018.12.058>
26. B. López Paredes, H.M. Araújo, F. Froborg, N. Marangou, I. Olcina, T.J. Sumner, R. Taylor, A. Tomás, A. Vacheret, *Astropart. Phys.* **102**, 56–66 (2018). <https://doi.org/10.1016/j.astropartphys.2018.04.006>
27. P. Sorensen, K. Kamdin, *JINST* **13**(02), P02032 (2018). <https://doi.org/10.1088/1748-0221/13/02/P02032>
28. D.S. Akerib et al. [LUX], *Phys. Rev. D* **102**(9), 092004 (2020). <https://doi.org/10.1103/PhysRevD.102.092004>
29. J. Griffiths, S. Kleinegesse, D. Saunders, R. Taylor, A. Vacheret, *arXiv:1807.06853* [physics.ins-det] (2018)
30. P. Holl, L. Hauertmann, B. Majorovits, O. Schulz, M. Schuster, A.J. Zsigmond, *Eur. Phys. J. C* **79**(6), 450 (2019). <https://doi.org/10.1140/epjc/s10052-019-6869-2>
31. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, *Rev. Mod. Phys.* **91**(4), 045002 (2019). <https://doi.org/10.1103/RevModPhys.91.045002>
32. G. Bertone, M.P. Deisenroth, J.S. Kim, S. Liem, R. Ruiz de Austri, M. Welling, *Phys. Dark Univ.* **24**, 100293 (2019). <https://doi.org/10.1016/j.dark.2019.100293>
33. G. Bertone, N. Bozorgnia, J.S. Kim, S. Liem, C. McCabe, S. Otten, R. Ruiz de Austri, *JCAP* **03**, 026 (2018). <https://doi.org/10.1088/1475-7516/2018/03/026>
34. E. Armengaud et al. [EDELWEISS], *JCAP* **05**, 019 (2016). <https://doi.org/10.1088/1475-7516/2016/05/019>
35. D.S. Akerib et al. [LUX], *Phys. Rev. D* **104**(1), 012011 (2021). <https://doi.org/10.1103/PhysRevD.104.012011>
36. A. Dey, J. Lahiri, B. Mukhopadhyaya, *JHEP* **09**, 004 (2019). [https://doi.org/10.1007/JHEP09\(2019\)004](https://doi.org/10.1007/JHEP09(2019)004)
37. C.K. Khosa, V. Sanz, M. Soughton, *Sci. Post Phys.* **10**(6), 151 (2021). <https://doi.org/10.21468/SciPostPhys.10.6.151>
38. S.K. Shirobokov, A.E. Ustyuzhanin, A.I. Golutvin, *J. Phys. Conf. Ser.* **1525**(1), 012087 (2020). <https://doi.org/10.1088/1742-6596/1525/1/012087>
39. R. Agnese et al. [SuperCDMS], *Phys. Rev. D* **92**(7), 072003 (2015). <https://doi.org/10.1103/PhysRevD.92.072003>
40. B. Matusch et al. [PICO], *arXiv:1811.11308* [physics.comp-ph] (2018)
41. C.K. Khosa, L. Mars, J. Richards, V. Sanz, *J. Phys. G* **47**(9), 095201 (2020). <https://doi.org/10.1088/1361-6471/ab8e94>
42. J. Brehmer, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer, *Astrophys. J.* **886**(1), 49 (2019). <https://doi.org/10.3847/1538-4357/ab4c41>
43. S. Alexander, S. Gleyzer, E. McDonough, M.W. Toomey, E. Usai, *Astrophys. J.* **893**, 15 (2020). <https://doi.org/10.3847/1538-4357/ab7925>
44. C. Escamilla-Rivera, M.A.C. Quintero, S. Capozziello, *JCAP* **03**, 008 (2020). <https://doi.org/10.1088/1475-7516/2020/03/008>
45. S. Delaquis et al. [EXO], *JINST* **13**(08), P08023 (2018). <https://doi.org/10.1088/1748-0221/13/08/P08023>
46. M. Kekic et al. [NEXT], *JHEP* **01**, 189 (2021). [https://doi.org/10.1007/JHEP01\(2021\)189](https://doi.org/10.1007/JHEP01(2021)189)
47. L.K. Hansen, P. Salamon, *IEEE Trans. Pattern Anal. Mach. Intell.* **v12**, 10, 993–1001 (1990). <https://doi.org/10.1109/34.58871>
48. Z.-H. Zhou, J. Wu, W. Tang, *Artif. Intell.* **137**(1–2), 239–263 (2002). [https://doi.org/10.1016/S0004-3702\(02\)00190-X](https://doi.org/10.1016/S0004-3702(02)00190-X)
49. M.A.H. Akhand, K. Murase, *Artificial Neural Networks - ICANN 2007*, 98–108 (Springer Berlin Heidelberg, 2007). https://doi.org/10.1007/978-3-540-74690-4_11
50. S. Tao, *arXiv:1904.05488* [cs.LG] (2019)
51. F. Pedregosa et al., *J. Mach. Learn. Res.* **12**, 2825–2830 (2011) jmlr.org/papers/v12/pedregosa11a.html
52. V. Sindhwani, P. Niyogi, M. Belkin, In *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, (Association for Computing Machinery, New York, NY, USA, 824–831, 2005). <https://doi.org/10.1145/1102351.1102455>

53. N.S. Sohoni, J.A. Dunnmon, G. Angus, A. Gu, C. Ré, [arXiv:2011.12945](https://arxiv.org/abs/2011.12945) [cs.LG] (2020)
54. W. Ertel, Introduction to Artificial Intelligence, **2**, XIV, 356 (Springer International Publishing, 2017). <https://doi.org/10.1007/978-3-319-58487-4>
55. C. Strobl, A.L. Boulesteix, A. Zeileis et al., BMC Bioinf. **8**, 25 (2007). <https://doi.org/10.1186/1471-2105-8-25>
56. F. Chollet et al., Keras, (2015). <https://keras.io>
57. E.M. Metodiev, B. Nachman, J. Thaler, JHEP **10**, 174 (2017). [https://doi.org/10.1007/JHEP10\(2017\)174](https://doi.org/10.1007/JHEP10(2017)174)