



# Study on neutron-gamma discrimination methods based on GMM-KNN and LabVIEW implementation

Ting-Meng Ding<sup>1</sup> · Yu-Hang Jiang<sup>1</sup> · Xuan-Xi Wang<sup>1</sup> · Xiao-Fei Jiang<sup>1</sup>

Received: 23 November 2023 / Revised: 2 April 2024 / Accepted: 7 April 2024 / Published online: 12 October 2024

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2024

## Abstract

Machine learning algorithms are considered as effective methods for improving the effectiveness of neutron-gamma ( $n$ - $\gamma$ ) discrimination. This study proposed an intelligent discrimination method that combined a Gaussian mixture model (GMM) with the K-nearest neighbor (KNN) algorithm, referred to as GMM-KNN. First, the unlabeled training and test data were categorized into three energy ranges: 0–25 keV, 25–100 keV, and 100–2100 keV. Second, GMM-KNN achieved small-batch clustering in three energy intervals with only the tail integral  $Q_{\text{tail}}$  and total integral  $Q_{\text{total}}$  as the pulse features. Subsequently, we selected the pulses with a probability greater than 99% from the GMM clustering results to construct the training set. Finally, we improved the KNN algorithm such that GMM-KNN realized the classification and regression algorithms through the LabVIEW language. The outputs of GMM-KNN were the category or regression predictions. The proposed GMM-KNN constructed the training set using unlabeled real pulse data and realized  $n$ - $\gamma$  discrimination of  $^{241}\text{Am}$ -Be pulses using the LabVIEW program. The experimental results demonstrated the high robustness and flexibility of GMM-KNN. Even when using only 1/4 of the training set, the execution time of GMM-KNN was only 2021 ms, with a difference of only 0.13% compared with the results obtained on the full training set. Furthermore, GMM-KNN outperformed the charge comparison method in terms of accuracy, and correctly classified 5.52% of the ambiguous pulses. In addition, the GMM-KNN regressor achieved a higher figure of merit (FOM), with FOM values of 0.877, 1.262, and 1.020, corresponding to the three energy ranges, with a 32.08% improvement in 0–25 keV. In conclusion, the GMM-KNN algorithm demonstrates accurate and readily deployable real-time  $n$ - $\gamma$  discrimination performance, rendering it suitable for on-site analysis.

**Keywords**  $n$ - $\gamma$  discrimination · GMM · KNN · LabVIEW · Classification · Regression

## 1 Introduction

Scintillation detectors are typically used for neutron detection [1–3]. In scintillation detectors, neutron and gamma-ray signals have different time decay constants, with neutron signals having longer decay times than gamma-ray signals [4]. Thus, neutrons and gamma rays have different pulse shapes, and pulse shape discrimination (PSD) [5] utilizes this difference to discriminate between the two categories.

In conventional  $n$ - $\gamma$  discrimination algorithms, the charge comparison method (CCM) uses the PSD factor as a discriminating index [6], which is the ratio of the tail integral of the pulse ( $Q_{\text{tail}}$ ) to the total integral ( $Q_{\text{total}}$ ) [7, 8]. In general, the PSD factor corresponding to neutrons is larger than that corresponding to gamma rays. The CCM is simple and easy to use; however, it is weak in case of  $n$ - $\gamma$  discrimination in the low-energy range.

Currently, machine learning algorithms are widely used for  $n$ - $\gamma$  discrimination [9, 10]. Machine learning algorithms include unsupervised and supervised learning algorithms. Unsupervised learning algorithms cluster data according to their distribution in the feature space, which does not rely on pre-labeled samples and can identify abnormal pulse events [11, 12]. The Gaussian mixture model (GMM), which is commonly used in unsupervised learning, has demonstrated good performance in  $n$ - $\gamma$

---

This work was supported by National Science Fund for Distinguished Young Scholars of China (No.12205062).

✉ Xiao-Fei Jiang  
xfjiang1@gzu.edu.cn

<sup>1</sup> Big Data and Information Engineering, GuiZhou University, Guiyang 550025, China

discrimination [13, 14]. However, the direct clustering of complete pulse data by the GMM still faces challenges. First, clustering high-dimensional data directly leads to the "curse of dimensionality." Second, when performing direct clustering on massive amounts of data using the GMM, a large number of significant errors occur, with neutrons incorrectly identified as  $\gamma$  rays within clusters with clear boundaries. Furthermore, GMM clustering is suitable only for fixed data [15, 16].

To overcome the limitations of the GMM in  $n$ - $\gamma$  discrimination, Liu et al. [14] proposed a method combining principal component analysis (PCA) and GMM clustering. They extracted three features using PCA and then applied the GMM clustering algorithm. The results indicated that the PCA-GMM provided a higher figure of merit (FOM) for  $n$ - $\gamma$  discrimination than the CCM. However, this method has strict data requirements, necessitating close pulse peak positions and including only pulse tails while disregarding the differences in the overall pulse.

Wang et al. [17] proposed a method for identifying neutrons and  $\gamma$  rays using a small-batch GMM clustering algorithm. This method yields a higher FOM than the CCM. However, in case of a large mismatch between the test pulse and the trained model, the exponent yields a large negative value, indicating that this method is susceptible to outliers.

Supervised learning algorithms can classify or regress unknown signals; however, they depend on prior knowledge. Common supervised learning algorithms include K-nearest neighbors (KNN) [18–20], support vector machines (SVM) [21, 22], and linear discriminant analysis (LDA) [23]. The KNN algorithm performs classification or regression prediction by calculating the distance between the test data and training samples. It is a simple and portable algorithm that accurately performs classification and regression tasks based on existing samples. However, the real-time performance of the KNN algorithm remains debatable, and its stability under different conditions requires further exploration.

Durbin et al. [18] proposed a new method that uses KNN regression to improve the PSD performance. This approach enabled direct comparison with conventional PSD methods using the FOM. However, this study did not consider the runtime of the algorithm, which is a critical indicator of real-time performance [24–27].

All the studies mentioned above concentrated on using machine learning algorithms to improve the  $n$ - $\gamma$  discrimination capabilities. However, they did not address the algorithmic portability or real-time performance issues. Thus, this study proposed a combined method of GMM and KNN algorithms (GMM-KNN) to overcome the limitations of a single machine learning algorithm in  $n$ - $\gamma$  discrimination. The proposed method constructs a training set from unlabeled data to discriminate unknown pulses.

## 2 Method

In this study, the GMM-KNN algorithm combines supervised learning algorithm (GMM clustering) with an unsupervised learning algorithm (KNN classification and regression). Further, the GMM-KNN achieves pulse discrimination using the LabVIEW program.

### 2.1 GMM clustering

The GMM is a probabilistic model that describes a dataset comprising multiple Gaussian distributions [28]. For each Gaussian component, the probability density function is expressed as:

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})}, \mathbf{x} \in \mathbb{R}, \quad (1)$$

where  $n$  denotes the dimensions of the pulse data;  $\boldsymbol{\mu}$  denotes the  $n$ -dimensional mean vector; and  $\Sigma$  denotes the covariance matrix of  $n \times n$ . Obviously, the Gaussian distribution is determined by the mean vector  $\boldsymbol{\mu}$  and the covariance matrix  $\Sigma$ . The initial pulse has 248 features; however, such a high dimension leads to the "curse of dimensionality." To reduce the number of pulse features, this study used only  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  as the two-dimensional (2-D) features of the pulse data, that is  $n = 2$ .

Neglecting pulse stacking in the  $n$ - $\gamma$  discrimination, the GMM has only two components: neutrons and gamma rays. For a Gaussian mixture distribution with two mixed components, the probability density is expressed as:

$$f_M(\mathbf{x}) = \sum_{i=1}^2 \alpha_i \cdot f(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i), \quad (2)$$

where  $\alpha_i$  is the "mixture coefficient", which is the probability of selecting the  $i$ th component Gaussian. Here,  $\alpha_i > 0$ , and  $\sum_{i=1}^2 \alpha_i = 1$ .

The model parameters  $\alpha_i$ ,  $\boldsymbol{\mu}_i$ , and  $\Sigma_i$  must be solved by the iterative optimization of the expectation-maximum (EM) algorithm [29, 30]. Each iteration of the EM algorithm comprises two steps: step E, which involves estimating the expectation of the hidden variables based on the current parameters; and step M, which involves using the computational results from step E to update the model parameters based on maximum likelihood estimation.

A large number of errors occurred when GMM clustering was performed directly on the entire training dataset. To enhance the accuracy of GMM clustering, we divided the data into three energy ranges [17]: 0–25 keV, 25–100 keV, and 100–2100 keV. The GMM soft clustering output

provides the probability that a pulse belongs to either neutrons or gamma rays. If the probability of a pulse belonging to neutrons exceeded 50%, it was classified as a neutron; otherwise, it was classified as a gamma ray. The primary objective of GMM clustering was to produce a dependable training set that is subsequently utilized by the KNN algorithm. However, this classification method may encounter ambiguous pulses, which can reduce the accuracy of KNN classification.

## 2.2 KNN classification and regression algorithm

GMM clustering can only cluster a fixed dataset; therefore, supervised learning algorithms must be used for real-time discrimination. This study used supervised learning to accurately represent the distribution of the training set samples, which followed a Gaussian mixture distribution comprising two components. The goal of supervised learning was to ensure that the test set data exhibited classification/clustering results similar to those of the training set. Among the supervised learning algorithms, such as SVM and LDA, KNN was selected for its simplicity and ease of implementation in LabVIEW programs. The KNN algorithm is known for its simplicity and accuracy, ensuring a generalization error that is not more than twice the error rate of the Bayes optimal classifier. To optimize the KNN algorithm, it is important to determine the optimal value of  $K$ , select an appropriate distance metric, and specify the decision rule.

First, the key to the KNN algorithm is the determination of the optimal  $K$  value. When the  $K$  value is excessively large, the distant points affect the prediction results, resulting in underfitting; conversely, if  $K$  is excessively small, the model is less tolerant of noise and prone to overfitting. In this study, the method used to determine the optimal  $K$  value was tenfold cross-validation [31]. A tenfold cross-validation means that the training dataset  $D$  was divided into ten equal parts, nine of which were used to train the model, and the remaining 1 part of the data was used to compute the test accuracy. This process was repeated for each part of the data to obtain the average accuracy. There exists a maximum value for the results of the tenfold cross-validation, where the accuracy value first increases with increasing  $K$  and then decreases after reaching the maximum value. The  $K$  value corresponding to the highest accuracy average is the optimal  $K$  value. The dataset  $D$  must contain a sufficient number of samples, and simultaneously, the result of the GMM clustering can obtain a reliable training set. Therefore, GMM-KNN uses the training set obtained from GMM clustering as dataset  $D$ .

In addition, the KNN algorithm must calculate the distance between an unknown pulse and each pulse in the training set (in this study, the distance is the Euclidean distance).

For example, for points  $X$  and  $Y$  with  $n$  features, the distance is calculated as

$$D(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}. \quad (3)$$

Because we use  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  as the 2 features of pulses, the above equation has  $n = 2$ . Calculating the distance between the two pulses simply involves replacing  $x$  and  $y$  with  $Q_{\text{tail}}$  and  $Q_{\text{total}}$ , respectively.

Finally, the KNN algorithm outputs the classification or regression results for the pulses in the test set. For the KNN classification task, the result is the category that occurs most frequently in the nearest neighboring  $K$  instances [32]. For the KNN regression task, the result is the average value of each feature over the nearest neighboring  $K$  instances [33].

## 2.3 GMM-KNN classification and regression

Both GMM and KNN are effective methods for n- $\gamma$  discrimination. GMM clustering performs well in PSD analysis. However, the results of GMM clustering are limited to the current dataset, and prior knowledge obtained from the training set cannot be directly applied to the test set. In contrast, KNN accurately captures the sample distribution and can be easily implemented on hardware, providing strong real-time performance and the potential for real-time n- $\gamma$  discrimination. However, the drawback of KNN is that it requires prestorage of the sample set.

In this study, the goal of GMM clustering was to construct a reliable training set, and KNN utilized this training set to discriminate unknown pulses. To integrate the GMM and KNN, we proposed improvements to both methods. When GMM clusters the data across the entire energy range, a significant number of misclassifications are obtained. Therefore, in GMM-KNN, the data were divided into three energy partitions to enhance the clustering performance. In addition, the GMM clustering results often include confusing pulses (low-probability events). Hence, we selected only the pulses with classification probabilities greater than 99% as the training set. The test set of GMM-KNN comprised a large number of unknown pulses, and each pulse must calculate its distance from all samples in the training set. Using the complete training set yielded the most accurate classification results but significantly increased the computational cost of the algorithm.

In the context of GMM-KNN classification, reducing the size of the training dataset has a minimal impact on the classification results, but significantly decreases the algorithm complexity. This facilitates a flexible selection of the sample quantity within the training set for the GMM-KNN classification. However, for GMM-KNN regression, a complete training set must be used to ensure the accuracy of the regression predictions. To assess the real-time performance

of our method, a unified programming language or first-in, first-out (FIFO) data transfer with other devices must be employed. The LabVIEW program can call or directly read the acquired data, which facilitates the real-time implementation of the GMM-KNN for pulse discrimination. Using parallel computing, the LabVIEW program can concurrently calculate two decision values, resulting in simpler decision logic and faster computation. In addition, the real-time performance of the algorithm must consider the runtime and memory footprint of the trained model. Thus, this study proposed the GMM-KNN algorithm that employed only two features,  $Q_{\text{tail}}$  and  $Q_{\text{total}}$ , for both GMM clustering and KNN classification and regression. These features reduced data dimensionality in GMM clustering and significantly decreased the computational cost and memory footprint of the trained model.

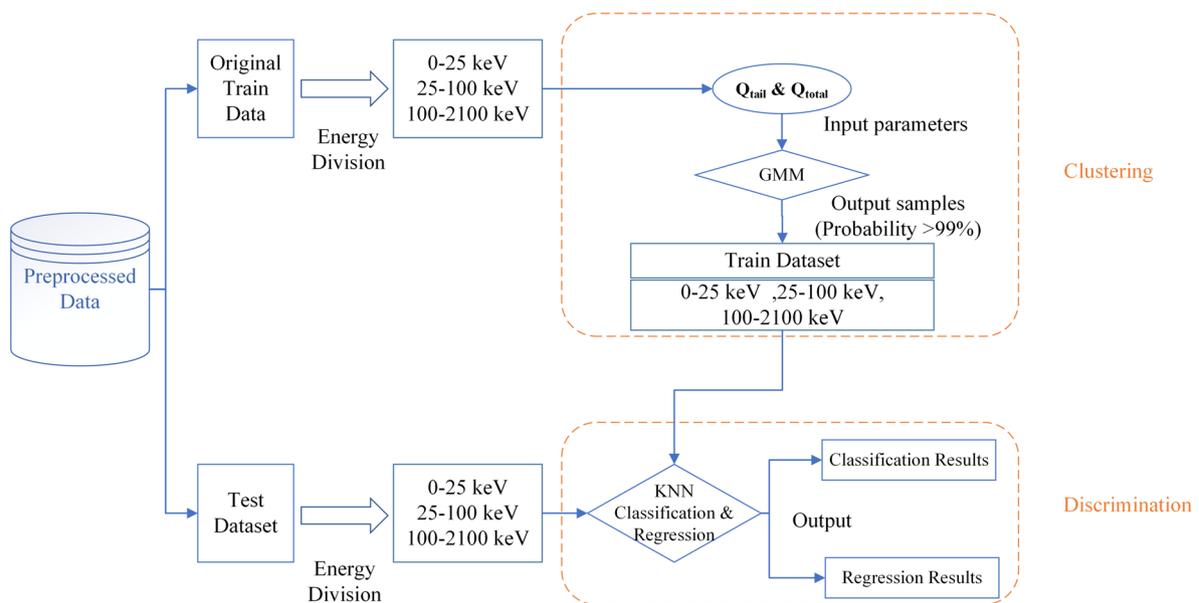
A block diagram of the procedure of GMM-KNN algorithm is shown in Fig. 1. First, the method divides the pre-processed training and test data into three parts separately. These three energy ranges are 0–25 keV, 25–100 keV, and 100–2100 keV, respectively. Second, GMM takes  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  as pulse features and performs small-batch clustering in three energy ranges. Consequently, this method selects a portion of the data of clustering results (probability > 99%) as the training set of KNN. Finally, GMM-KNN implements the classification and regression algorithms with LabVIEW programming and subsequently outputs the category and regression prediction values.

## 2.4 Evaluation metrics

The output of the GMM-KNN classification was binary, with zero representing gamma rays and one representing neutrons. Both pulse types exhibited an elliptical distribution in the feature space. Comparing the difference between the output and ground truth facilitated the qualitative assessment of the effectiveness of the n- $\gamma$  discrimination. The outputs of the GMM-KNN regression comprised the average values of the nearest  $K$  pulses for  $Q_{\text{tail}}$  and  $Q_{\text{total}}$ , as well as their ratios. We used this ratio to calculate the FOM, where a higher FOM indicated better n- $\gamma$  discrimination performance.

There is a difference in the ratio of slow charge to total charge between the two types of pulses, neutrons and gamma rays; therefore, the CCM selects the ratio of the tail integral ( $Q_{\text{tail}}$ ) to the total integral ( $Q_{\text{total}}$ ) as the PSD factor [34] and then calculates the FOM as the discrimination metric. In this study,  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  were adjusted to obtain the best FOM.  $Q_{\text{tail}}$  was considered as 68 ns and  $Q_{\text{total}}$  as 124 ns (Fig. 2). For the GMM-KNN regression, the pulse features are the predicted regression values, thus,  $\text{PSD} = (\text{Regressed } Q_{\text{tail}}) / (\text{Regressed } Q_{\text{total}})$ . After calculating the PSD values for all pulses, the GMM-KNN regression used the updated FOM as the evaluation metric, which can use the FOM to evaluate the n- $\gamma$  discrimination effectiveness further.

Two Gaussian peaks were observed in the PSD histogram after Gaussian fitting (Fig. 3), and the FOM is calculated as follows:



**Fig. 1** Block diagram of n- $\gamma$  discrimination based on GMM-KNN

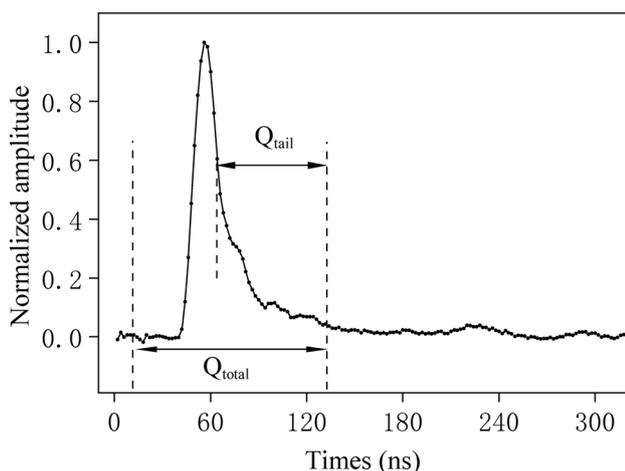


Fig. 2 Diagram of the tail integral  $Q_{tail}$  and the total integral  $Q_{total}$

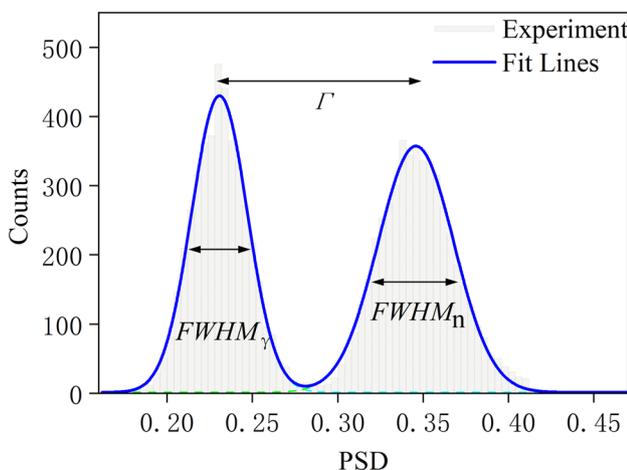


Fig. 3 Definition of FOM

$$FOM = \frac{\Gamma}{FWHM_{\gamma} + FWHM_n}, \tag{4}$$

where  $\Gamma$  is the distance between the two Gaussian peaks and  $FWHM_{\gamma}$  and  $FWHM_n$  are the full width at half maximum of the gamma and neutron peaks, respectively. A larger FOM value implies a higher separation between the two Gaussian peaks. Thus, the algorithm n/γ discrimination was more effective.

### 3 LabVIEW implementation of GMM-KNN

LabVIEW is a graphical programming language used mainly in the fields of control, measurement, and data acquisition [35]. The nuclear signal output from the detector was digitized and stored using the LabVIEW host computer program and

the signals were processed using discrimination algorithms. To evaluate the real-time performance of the algorithm for data acquisition, we used the unified LabVIEW programming language. KNN classification using the LabVIEW program first presorted the training set. Consequently, the distance values were completely independent of the index values, which simplified judgment logic. The array operations in KNN regression facilitated regression values for all features to be calculated simultaneously, greatly simplifying the LabVIEW program. In the LabVIEW program for real-time n-γ discrimination, the unknown pulses were computed as arrays. In this study, the GMM-KNN classification and GMM-KNN regression algorithms were applied to the same test set for discrimination, and the output results were saved as a.csv file.

#### 3.1 Energy division

The pulses in the test set were randomly collected in the real experiments. The pulses were divided into three parts based on three energy ranges: 0–25 keV, 25–100 keV, and 100–2100 keV. Figure 4 shows the LabVIEW program diagram for pulses division based on energy. The conditional diagram uses energy as the determinant, and the program counts the pulses in each energy range. Thereafter, by sorting the energy in ascending order and setting the index of "array subset vi" based on the count value, we can obtain three sub-arrays.

#### 3.2 GMM-KNN classification algorithm

A program block diagram of the GMM-KNN classification is shown in Fig. 5, GMM-KNN classification to determine the category of a pulse included three steps:

Step 1 - Training set composition: In the GMM-KNN algorithm, the training set processed by GMM clustering contained 26172 pulses (probability >99%), and the test pulses calculated the distances with each pulse of the training set. It is more computationally intensive when using the full training set directly; therefore, we sorted the data within the training set by probability value and take one sample every three values. These samples formed a 2-D array training set  $M_c$ . The columns of  $M_c$  represent the two features of a pulse,  $Q_{tail}$  and  $Q_{total}$ , and each row represents a pulse. The  $m$  rows in front of the 2-D array are the gamma rays and the remaining  $k$  rows are neutrons. The expression for  $M_c$  is as follows:



$$M_c = \begin{bmatrix} \gamma_1(\text{train}) \\ \gamma_2(\text{train}) \\ \gamma_m(\text{train}) \\ \vdots \\ n_1(\text{train}) \\ n_2(\text{train}) \\ \vdots \\ n_k(\text{train}) \end{bmatrix} = \begin{bmatrix} Q_{\text{tail}}(\gamma_1) & Q_{\text{total}}(\gamma_1) \\ Q_{\text{tail}}(\gamma_2) & Q_{\text{total}}(\gamma_2) \\ \vdots & \vdots \\ Q_{\text{tail}}(\gamma_m) & Q_{\text{total}}(\gamma_m) \\ Q_{\text{tail}}(n_1) & Q_{\text{total}}(n_1) \\ Q_{\text{tail}}(n_2) & Q_{\text{total}}(n_2) \\ \vdots & \vdots \\ Q_{\text{tail}}(n_k) & Q_{\text{total}}(n_k) \end{bmatrix} \quad (5)$$

Step 2 - Distance values computation: This involved computing the distance values between the test pulse feature vector  $\mu = [Q_{\text{tail}}(\text{test}) + Q_{\text{total}}(\text{test})]$  and all pulses in the training set. These distance values formed a one-dimensional (1-D) array  $D(M_c, \mu)$ . The elements of  $D(M_c, \mu)$  are the Euclidean distance values between each row of the array within  $M_c$  and feature vector  $\mu$ :

$$D(M_c, \mu) = \begin{bmatrix} D(\gamma_1(\text{train}), \mu) \\ D(\gamma_2(\text{train}), \mu) \\ \vdots \\ D(\gamma_m(\text{train}), \mu) \\ D(n_1(\text{train}), \mu) \\ D(n_2(\text{train}), \mu) \\ \vdots \\ D(n_k(\text{train}), \mu) \end{bmatrix} \quad (6)$$

$$= D_\gamma(M_c, \mu) + D_n(M_c, \mu)$$

Step 3 - Pulse classification: The distance subsets  $D_\gamma(M_c, \mu)$  and  $D_n(M_c, \mu)$  corresponded to gamma rays and neutrons, respectively. The smallest  $K$  values were obtained from the distance subsets  $D_\gamma(M_c, \mu)$  and  $D_n(M_c, \mu)$ . The result

of the tenfold cross-validation indicated a maximum accuracy of 99.317 % at  $K = 5$ ; thus,  $K$  was set as 5. The two distance subsets were sorted in ascending order separately, and the smallest five gamma rays and five neutrons in each of  $D_\gamma(M_c, \mu)$  and  $D_n(M_c, \mu)$  were named as  $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5; n_1, n_2, n_3, n_4, n_5$ , respectively.

When  $\gamma_3 < n_3$ , we have

$$\begin{cases} n_1 < n_2 < n_3, \\ \gamma_1 < \gamma_2 < \gamma_3 < n_3. \end{cases} \quad (7)$$

The above equation holds for  $\gamma_4 < n_3$  and  $\gamma_5 < n_3$ . In these cases, the number of gamma rays must be greater than the number of neutrons in the five nearest neighbor values.

When  $n_3 < \gamma_3$ , we have

$$\begin{cases} \gamma_1 < \gamma_2 < \gamma_3, \\ n_1 < n_2 < n_3 < \gamma_3. \end{cases} \quad (8)$$

The above equation holds for  $n_4 < \gamma_3$  and  $n_5 < \gamma_3$ . In these cases, the number of neutrons must be greater than the number of gamma rays among the five nearest neighbor values.

In summary, the 3rd value of each of the subsets ( $\gamma_3$  and  $n_3$ ) was the judgment value. When  $\gamma_3 < n_3$ , the pulses were determined to be gamma rays. Conversely, when  $n_3 < \gamma_3$ , the pulses were considered as neutrons. In classification, the traditional KNN algorithm must perform five judgments and five counts. However, the GMM-KNN classification requires only one judgment, which improves algorithm efficiency and reduces algorithm complexity.

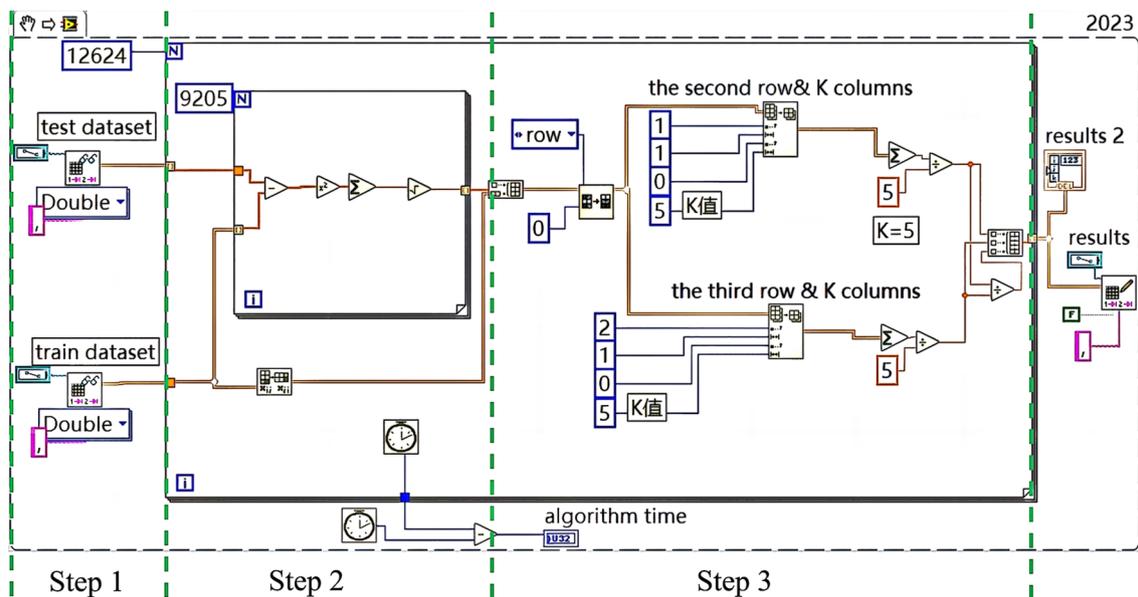


Fig. 6 Block diagram of LabVIEW program for GMM-KNN regression algorithm

### 3.3 GMM-KNN regression algorithm

GMM-KNN regression requires determining the nearest neighbor of the  $K(K = 5)$  values and calculating the average of the five values. The algorithm does not divide the training set into two groups but requires a complete training set. Similar to the GMM-KNN classification, the implementation of the GMM-KNN regression with LabVIEW involves three steps (Fig. 6):

Step 1: The distance values between the test pulse  $\mu = [Q_{\text{tail}}(\text{test}) + Q_{\text{total}}(\text{test})]$  and the  $n$  pulses in the training set  $M_r$  were calculated. Consequently, a 1-D array of distance values  $D(M_r, \mu)$  was obtained.

$$M_r = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} Q_{\text{tail}}^{(1)} & Q_{\text{total}}^{(1)} \\ Q_{\text{tail}}^{(2)} & Q_{\text{total}}^{(2)} \\ \vdots & \vdots \\ Q_{\text{tail}}^{(m)} & Q_{\text{total}}^{(m)} \end{bmatrix} \quad (9)$$

$$D(M_r, \mu) = \begin{bmatrix} D(v_1, \mu) \\ D(v_2, \mu) \\ \vdots \\ D(v_n, \mu) \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix} \quad (10)$$

Step 2: The 1-D array  $D(M_r, \mu)$  was combined with the 2-D array  $M_r$  of the training set into a new 2-D array  $M_n$ . The first row of  $M_n$  represented the distance value, the second row represented the  $Q_{\text{tail}}$  value of the pulses in the training set, and the third row represented the  $Q_{\text{total}}$  value of the pulses in the training set. We then sorted  $M_n$  in ascending order according to the first row and intercepted the first five columns to obtain  $M_5$ .

$$M_n = \begin{bmatrix} D_1 & D_2 & \dots & D_n \\ Q_{\text{tail}}^{(1)} & Q_{\text{tail}}^{(2)} & \dots & Q_{\text{tail}}^{(n)} \\ Q_{\text{total}}^{(1)} & Q_{\text{total}}^{(2)} & \dots & Q_{\text{total}}^{(n)} \end{bmatrix} \quad (11)$$

$$M_5 = \begin{bmatrix} D_{\text{min}1} & D_{\text{min}2} & \dots & D_{\text{min}5} \\ Q_{\text{tail}}(1) & Q_{\text{tail}}(2) & \dots & Q_{\text{tail}}(5) \\ Q_{\text{total}}(1) & Q_{\text{total}}(2) & \dots & Q_{\text{total}}(5) \end{bmatrix}$$

Step 3: The regression values of KNN were calculated. The first eigenvalue  $E_1$  was the average of the first five columns in row 2, the second eigenvalue  $E_2$  was the average of the first five columns in row 3, and the third eigenvalue  $PSD_{\text{GMM-KNN}}$  was the division of the first two eigenvalues.

$$E_1 = \frac{\sum_{i=1}^5 Q_{\text{tail}}(\text{mini})}{5}, \quad E_2 = \frac{\sum_{i=1}^5 Q_{\text{total}}(\text{mini})}{5} \quad (12)$$

$$PSD_{\text{GMM-KNN}} = E_1/E_2$$

The PSD values of all pulses were stored in.csv files, and the GMM-KNN regression used PSD as a discriminating index. The PSP values were used to compute the FOM, which can evaluate the n- $\gamma$  discrimination ability of the GMM-KNN regression. The GMM-KNN regression first calculated the distance value of the pulses and subsequently calculated the regression value of the nearest  $K$  value. The regression values for all pulses did not need to be calculated. Moreover, the regression values of  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  were calculated in parallel, which significantly reduced the computational complexity.

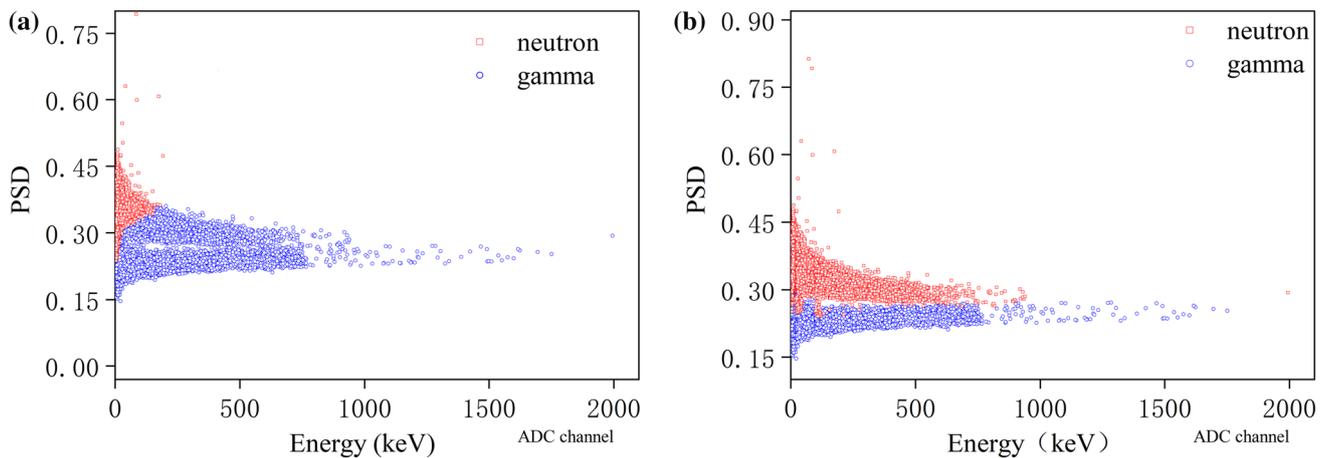


Fig. 7 (Color online) Results of GMM clustering. **a** Results of the direct GMM clustering. **b** Results of the small-batch GMM clustering after segmentation by energy

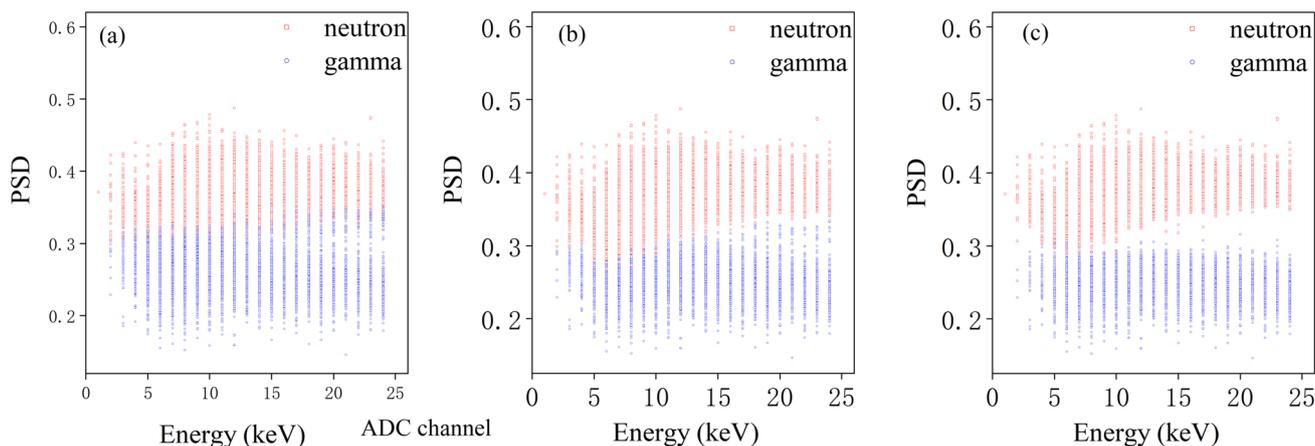


Fig. 8 (Color online) n-γ distribution in the training set within the 0–25 keV at different probabilities

Table 1 Distribution of pulses in three energy domains at intercepts of different probabilities

Probability					
Energy (keV)	Original dataset	90%	95%	99%	99.9%
0–25	12570	11002	10442	9177	7074
25–100	9576	9422	9357	9170	8867
100–2100	7854	7826	7810	7755	7671

### 4 Results and discussion

The neutron source used in this experiment was an <sup>241</sup>Am-Be source, the detector was an organic liquid scintillation detector (EJ-301) [36, 37], and the digitizer was DT5730B. The detector collected current pulses, which were digitized to obtain raw data. After preprocessing steps, such as smoothing, filtering, normalization, and baseline recovery, the raw data were transformed into initial data, which were stored in the computer. The preprocessed dataset of 60,000 pulses was divided into two parts. Of these, 30,000 pulses were used for the GMM clustering to obtain a reliable training set. The remaining 30,000 pulses were reserved to test the feasibility of the GMM-KNN algorithm.

#### 4.1 GMM clustering

Depending on whether the probability value exceeded 50%, we classified the GMM clustering results into two categories. Figure 7a shows the results of direct clustering, where the red squares and blue circles represent neutrons and gamma rays, respectively. There were significant errors in the pulse discrimination, with numerous neutrons

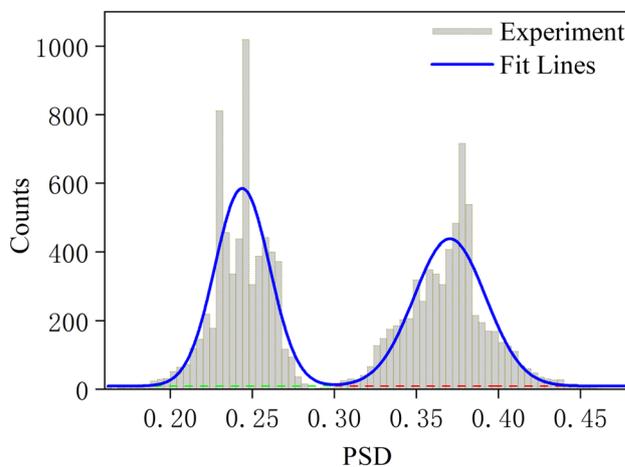


Fig. 9 PSD histogram of the GMM-KNN regression, when the training set comprises pulses with a probability greater than 99.9%

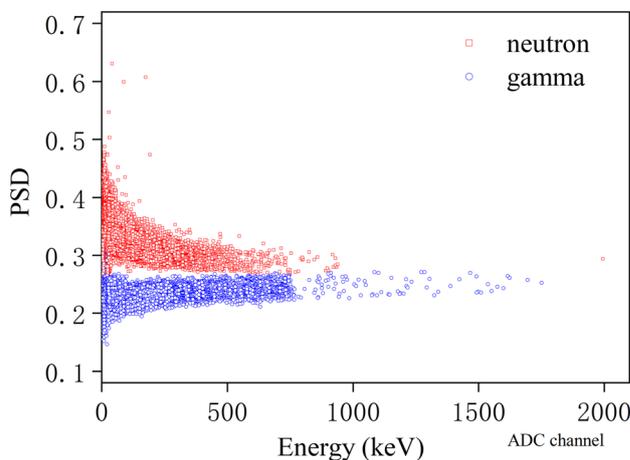


Fig. 10 Training set comprising pulses with probability above 99%

misclassified as gamma rays. Figure 7b shows the results of small-batch clustering in 3 energy ranges, where only a few ambiguous pulses were observed. Small-batch clustering reduced the misclassification rate and improved the effectiveness of  $n$ - $\gamma$  discrimination.

Although small-batch clustering yielded better results than direct clustering, Fig. 7 exhibits the red downward protrusion at the energy segmentation boundary (25 keV and 100 keV). To obtain a reliable training set, these ambiguous pulses must be removed. Within 0–25 keV, we progressively increased the probability of pulses in the training set (Fig. 8). Figures 8a–c corresponds to complete pulses, pulses with a probability above 90%, and pulses with a probability above 99%, respectively. Figure 8 demonstrates that a higher probability results in a better separation of pulses in the training set; however, it also decreases the number of pulses in 0–25 keV.

Table 1 presents the quantity distributions in the training set for different probabilities. At 25–2100 keV, there was minimal variation in the number of pulses. Most of the ambiguous pulses were concentrated at 0–25 keV. As the probability increased, the number of pulses in 0–25 keV decreased significantly. When the training set comprised pulses with a probability exceeding 99.9%, the number of pulses in the 0–25 keV range was even lower than that in the 25–2100 keV. In addition, the PSD histogram of the test set exhibited three distinct peaks and poor fitting performance at this stage (Fig. 9). Therefore, this study used a training set composed of pulses with a probability exceeding 99%.

Figure 10 shows the training set comprising pulses with a probability above 99%. In this case, the samples with PSD values below the threshold ( $PSD_{\text{threshold}}$ ) were removed at the energy segmentation boundaries. At this stage, the neutrons and gamma rays are completely separated, and the GMM-KNN algorithm only needed to select the remaining pulses from this portion to construct the training set. Once the training set was constructed, the GMM-KNN can flexibly select a subset of data from the training set and implement  $n$ - $\gamma$  discrimination using LabVIEW program.

## 4.2 GMM-KNN classification

In the context of real-time  $n$ - $\gamma$  discrimination, the algorithmic efficiency is a crucial factor that is influenced by both the size of the training set and the number of pulse features.

**Table 2** Algorithmic time consumption and error rate of different sample sizes

Sample sizes	Sample sizes					
	1/1	1/2	1/3	1/4	1/5	1/6
Total time (ms)	8828	4131	2712	2021	1648	1291
Error rate (%)	0.000	0.080	0.126	0.130	0.196	0.240

**Table 3** Algorithmic times of different ranges or Method

Methods	Energy (keV)	Times (ms)
GMM-KNN	0–25	2021
GMM-KNN	25–100	1791
GMM-KNN	100–2100	1446
64-points KNN	0–25	3425

This study investigated the time consumption and error rates of the GMM-KNN algorithm by employing average sampling techniques to select subsets of the complete training set with proportions of 1/2, 1/3, 1/4, 1/5, and 1/6 (Table 2). By comparing the time consumption and error rates of differently sized training sets, it was observed that when only 1/4 of the complete dataset was used, the classifier's time consumption was reduced to approximately 1/4 of the original, resulting in an average execution time per pulse of only 67  $\mu$ s compared to 294.27  $\mu$ s for the complete training set. The discriminative results between the reduced dataset and the complete training set exhibited a difference of only 0.13% (41 pulses, with 21 falling in the 0–25 keV). This sampling method significantly reduced the computational costs while ensuring reliable discrimination results. This facilitated flexible selection of the training set based on the experimental latency requirements.

KNN algorithms typically use the nonzero-amplitude portion of pulses as feature points, which encompass 64 points. Table 3 shows that the time consumption of the KNN algorithm was 4325 ms in the 0–25 keV. In contrast, GMM-KNN employed only two pulse features,  $Q_{\text{tail}}$  and  $Q_{\text{total}}$ , resulting in a reduced execution time of 2021 ms for the same test volume. The program was parallelized, and the selection of  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  significantly improved the processing speed. Notably, even when using the 64-points KNN algorithm, the execution time remained at 4325 ms, indicating that the algorithm was not very demanding in terms of the number of pulse features and could adapt to different pulse types and experimental requirements.

An imbalanced classification is a significant challenge for classification algorithms. In this study, we applied GMM clustering to partition a test dataset containing 10,000 gamma rays and 10,000 neutrons into 20 subsets, each comprising 1,000 pulses. We varied the gamma/neutron ( $\gamma/n$ ) ratios from 10:1 to 10:10 and from 10:10 to 1:10, resulting in 19 different ratios. We compared the results of the

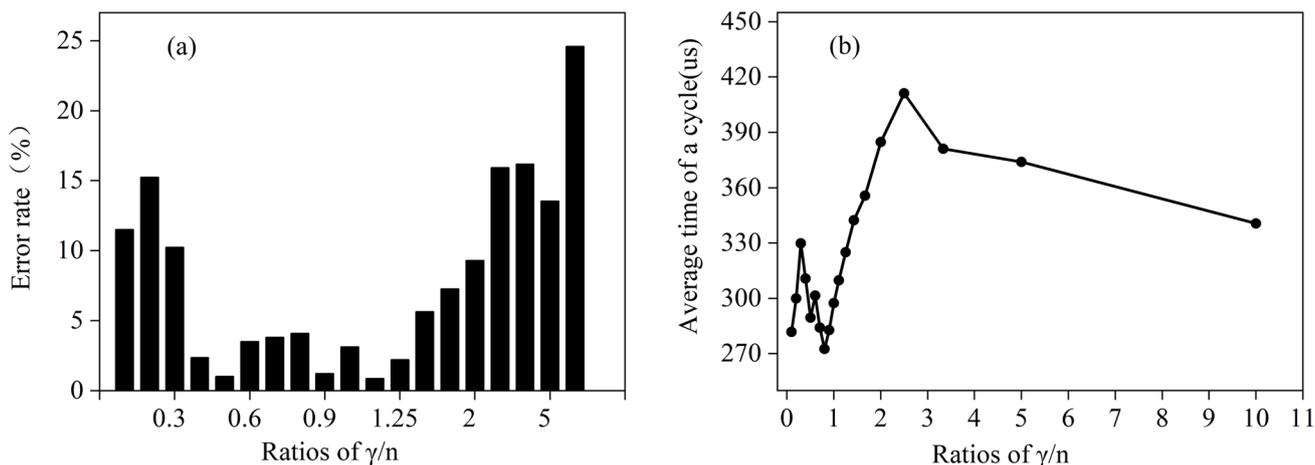


Fig. 11 Error rate and runtime for various gamma/neutron ratios

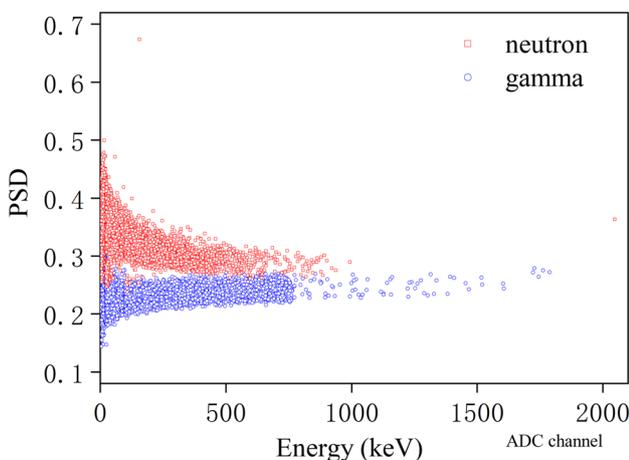


Fig. 12 (Color online) Classification results of GMM-KNN in a 2-D space comprising PSD and Energy

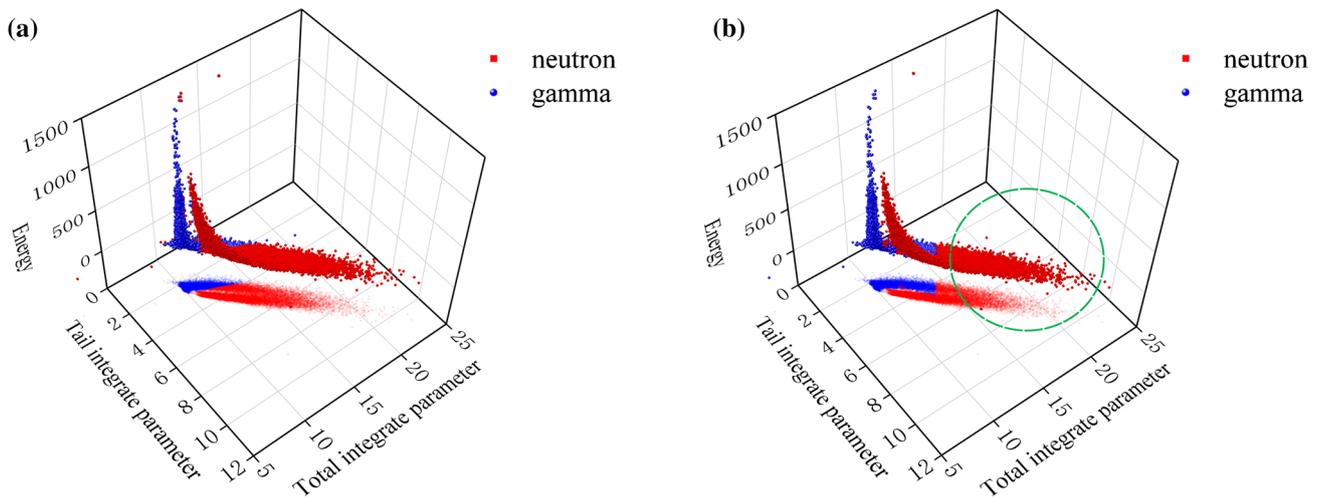
GMM-KNN classifier with the prior knowledge to obtain the error rates and average execution times per pulse for each ratio, as shown in Fig. 11. All seven ratios ranging from 10:8 to 6:10 exhibited error rates below 5%. The lowest error rate of 1.01% (1.23%) and average execution time of a cycle of 301.84  $\mu$ s (289.60  $\mu$ s) were achieved at a ratio of 9:10 (10:9), which is consistent with the  $\gamma/n$  ratio of approximately 9:10 (12,542 gamma rays to 13,619 neutrons) observed in the training set. The GMM-KNN classifier demonstrated excellent real-time performance, strong adaptability, and robustness. Thus, it exhibits great potential for real-time discrimination and can be utilized for on-site analysis, serving as a reference for offline n- $\gamma$  discrimination analysis.

Using the complete training set, the GMM-KNN performed the classification task simultaneously in three different energy ranges (0–25 keV, 25–100 keV, and 100–2100 keV). The discrimination results in these three

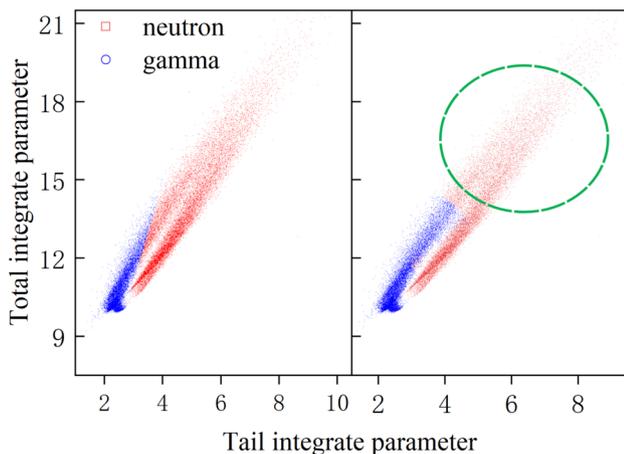
energy ranges were concatenated to obtain a discrimination result across the complete energy range (as shown in Fig. 12). The classification exhibited a small error at the energy boundaries, which is consistent with the GMM clustering results. This indicated that the test results accurately reflected the distribution of the data in the training set. In contrast to the CCM, which judges the pulse categories using the threshold  $PSD_{\text{threshold}}$ , the results of the GMM-KNN classification were more consistent with the Gaussian mixture distributions of the two components. In the low-energy range, an overlap between neutrons and gamma rays was observed, and we could not directly observe the classification results of these two pulse types in the low-energy region in Fig. 12. Only by comparing the classification effects of CCM and GMM-KNN in the feature space containing  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  can we effectively evaluate their classification performances.

Using  $Q_{\text{tail}}$ ,  $Q_{\text{total}}$  and energy as the X, Y, and Z axes, respectively, we obtained a three-dimensional (3-D) plot of the classification results in the feature space (Fig. 13). The red cubes and blue spheres represent neutrons and gamma rays, respectively. Figure 13a shows the classification results of CCM. Figure 13 shows the classification results of GMM-KNN. As evident, for the majority of pulses, the neutrons and gamma rays exhibited distinct cone-shaped distributions, which rendered them easy to distinguish. However, for certain lower-energy pulses, there was no clear boundary between the neutrons and gamma rays (indicated by the flattened red region in Fig. 13), resulting in their mixture and making differentiation difficult. To further compare the classification results of the CCM and GMM-KNN in the low-energy range, we focused on the performance of the two classification algorithms in the feature space.

Figure 14 shows the projection of the 3-D visualization from Fig. 13 onto the X – Y plane. In the 2-D feature



**Fig. 13** (Color online) Scatterplot in 3-D space comprising eigenvalues and energies. (a) CCM classification results. (b) GMM-KNN classification results



**Fig. 14** (Color online) Classification results of CCM and GMM-KNN in 2-D feature space. The left and right sides present the CCM and GMM-KNN classification results, respectively

space, we can observe the classification results of the CCM (left) and GMM-KNN (right). The two types of pulses exhibited approximately elliptical distributions, with blue representing gamma rays and red representing neutrons. The CCM determined the pulse categories based on the threshold,  $PSD_{\text{threshold}}$ . It constructed a histogram of the pulse PSD and fit it with a Gaussian distribution, thereby facilitating the separate fitting of peaks for neutrons and gamma rays. The midpoint between the two peaks was considered as the threshold  $PSD_{\text{threshold}}$ , which served as the criterion for distinguishing between neutrons and gamma rays. In Fig. 14,  $PSD_{\text{threshold}}$  is represented by a simple straight line (the equation of the line is  $y = \frac{1}{PSD_{\text{threshold}}}x$ ). In contrast, GMM worked better for ellip-

tical clustering, such that GMM-KNN classified 1657 (5.52%) gamma rays correctly. Although GMM-KNN still cannot completely discriminate neutrons and gamma rays in the low-energy range, it performs better classification compared to CCM. In the feature space, the GMM-KNN had a larger number of discriminable pulses.

For the portion of pulses that is difficult to distinguish for both methods (the flattened red region in the 3-D visualization), both the CCM and GMM-KNN tended to classify these ambiguous pulses as neutron pulses. This is because gamma rays exhibit a more concentrated distribution, resulting in a higher peak in the PSD histogram corresponding to gamma rays. As the  $PSD_{\text{threshold}}$  moved toward the gamma peak, more pulses are classified as neutrons. In the case of the GMM-KNN classification method, the training set was also generated by the GMM, resulting in classification preferences similar to those of the CCM.

A quantitative comparison of the classification results of CCM and GMM-KNN revealed that GMM-KNN improved the n- $\gamma$  discrimination. For quantitative analysis, we used the GMM-KNN regression algorithm.

### 4.3 GMM-KNN regression

To obtain a more generalizable metric, GMM-KNN employs regression to compute a quantifiable FOM value. This is because the difference between the two types of pulses, neutrons and gamma rays, increases with higher energies. Consequently, the FOM values vary across the three energy ranges.

For the CCM, we created a histogram of the PSD (Fig. 15). Figure 15a–c correspond to the test results within 0–25 keV, 25–100 keV, and 100–2100 keV, respectively.

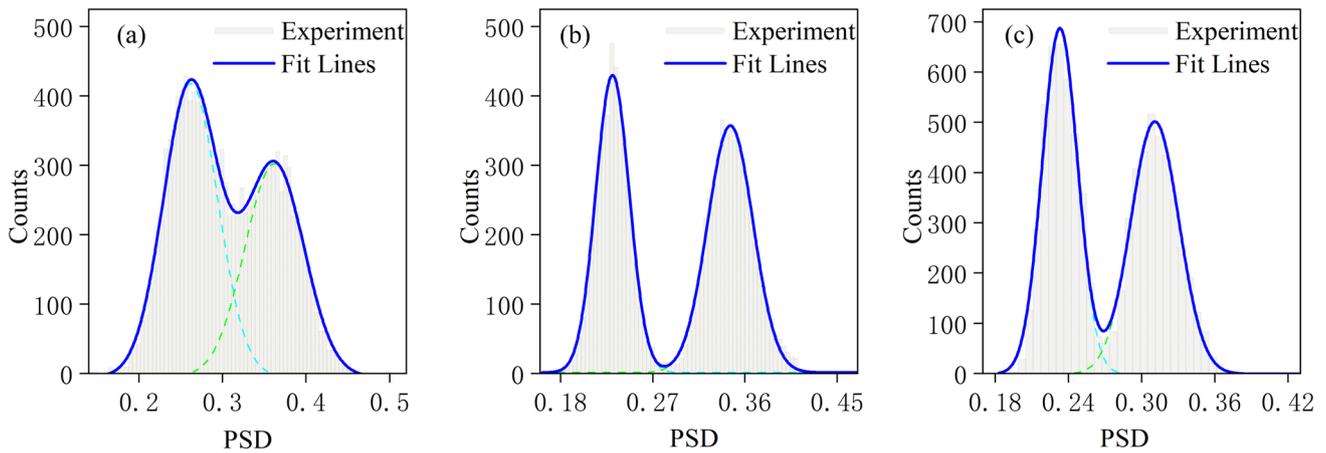


Fig. 15 Histograms of PSD from the CCM for the three energy levels

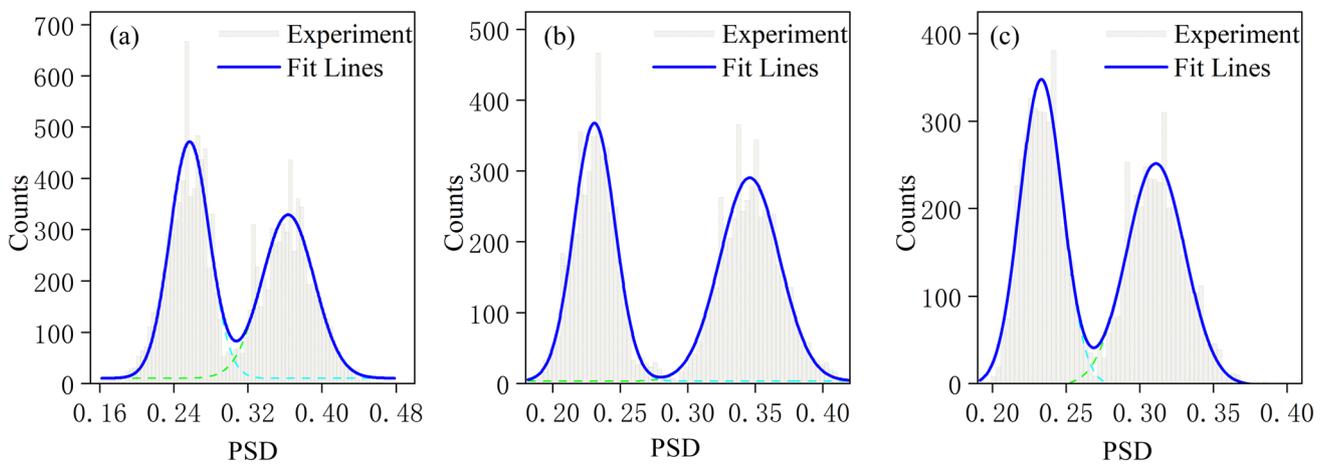


Fig. 16 Histograms of PSD from the GMM-KNN for the three energy levels

Table 4 FOM of CCM and GMM-KNN regression

Energy (keV)			
Method	0–25	25–100	100–2100
CCM	0.664	1.053	0.967
GMM-KNN	0.877	1.262	1.020
Increasing rate	32.08 %	19.85 %	5.48 %

Figure 15a corresponds to the lowest 0–25 keV range, where the two Gaussian-fit peaks were least separated. In this energy range, neutrons and gamma rays were the most difficult to discriminate, and the number of misclassified pulses was the highest. As the energy increased, the difference between the distributions of the two types of pulses increased, and the separation of the two Gaussian-fit peaks increased, as shown in Fig. 15b and c. These two parts of the pulse can be easily distinguished. The CCM is a simple

and effective discrimination method in the energy range of 25–2100 keV; therefore, the CCM can discriminate pulses with high energies well. However, at lower energy levels, there are numerous ambiguous pulses, and the level of ambiguity increases as the energy decreases.

For the GMM-KNN regression, we generated PSD histograms (Fig. 16). Figure 16a–c correspond to the test results within 0–25 keV, 25–100 keV, and 100–2100 keV, respectively. As shown in Fig. 16a, for GMM-KNN, the separation of the two Gaussian fitting peaks increased, and the FOM value greatly improved compared with that in case of CCM. The discrimination ability of the GMM-KNN was significantly better than that of the CCM in the energy range of 0–25 keV. However, in the higher energy ranges (25–100 keV and 100–2100 keV), the FOM values improved less and the separation of the two Gaussian-fitted peaks became slightly larger. When the effectiveness of n-γ discrimination improved, we observed that the PSD is highly

concentrated in certain bins. This is attributed to the fact that the training set was not a standard Gaussian distribution, but there were several small protrusions of varying heights. After the KNN regression, pulses in certain intervals became more concentrated, leading to a more pronounced peak in the histogram.

The results of the CCM and GMM-KNN regressions are presented in Table 4. In the 0–25 keV, neither method could completely separate the two types of pulses. However, the FOM of the GMM-KNN method improved by 32.08%, thereby significantly enhancing the discrimination between neutrons and gamma rays. This method facilitated a further reduction in the energy threshold for discriminable pulses, implying that neutrons and gamma rays can be distinguished in a larger energy range. In addition, the improvement in FOM by GMM-KNN decreased as the energy increased. At 25–2100 keV, the CCM could achieve basic separation between neutrons and gamma rays, thereby diminishing the additional benefits of machine learning methods.

## 5 Conclusion

We designed a new intelligent discrimination algorithm called GMM-KNN. This method can be used to construct a training set from unlabeled data and achieve the n- $\gamma$  discrimination of unknown pulses. GMM-KNN selected  $Q_{\text{tail}}$  and  $Q_{\text{total}}$  as pulse features to reduce the dimensions and flexibly select samples from the training set, which significantly reduced algorithm complexity. GMM-KNN also used the LabVIEW program to achieve KNN classification and regression. The LabVIEW program can be executed in parallel; however, the memory consumption of the array should be strictly controlled. We improved the KNN algorithm, particularly for KNN classification. The improved KNN algorithm significantly enhanced the running speed. Compared to the conventional KNN algorithm, the GMM-KNN classifier required only half the time to test the same dataset. Moreover, GMM-KNN could flexibly choose the number of training set samples based on specific experimental delay requirements. When only a quarter of the sample set data were used, the GMM-KNN classifier required only approximately 1/4 of the time (2021 ms) required for the full dataset, whereas the discrimination results differed by only 0.13% (41 pulses). Further, this method maintained a stable performance over a wide range of gamma/neutron ratios, rendering it suitable for different experimental data. Prior to running the GMM-KNN classifier, we presorted the two types of pulse samples in the training set and executed the LabVIEW parallel calculation, thereby reducing the complexity of the judgment logic by using only the size of the two judgment values to determine the pulse category. The GMM-KNN regression first calculated the distance value

and then synthesized it with the original pulse features into a 2-D array. This facilitated the calculation of the average of  $K$  values rather than all pulse regression values. To evaluate the n- $\gamma$  discrimination effect of the GMM-KNN algorithm, we qualitatively analyzed the scatter distribution and quantitatively calculated the FOM value. In the feature space, the GMM-KNN classification could better fit near-elliptical distributions, correctly classifying approximately 5.52% of the gamma rays. The FOM values for the GMM-KNN regression in the three energy ranges were 0.877, 1.262, and 1.020, respectively. Compared with the CCM, this method exhibited higher discrimination factors in each energy range, particularly in the low-energy domain (< 25 keV), where  $FOM_{\text{GMM-KNN}}$  increased by 32.08% compared with  $FOM_{\text{CCM}}$ . This study verified the effectiveness and feasibility of the GMM-KNN method, which exhibited excellent real-time performance, strong adaptability, robustness, and great potential for real-time discrimination. This renders it suitable for on-site analysis and providing new ideas and methods for the n- $\gamma$  discrimination field. In future work, the GMM-KNN will be deployed on an NI field-programmable gate array (FPGA) to form a composite device with the host computer, achieving high-accuracy real-time n- $\gamma$  discrimination.

**Author Contributions** All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Yu-Hang Jiang, Xuan-Xi Wang and Ting-Meng Ding. The first draft of the manuscript was written by Ting-Meng Ding and Xiao-Fei Jiang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Data Availability Statement** The data that support the findings of this study are openly available in Science Data Bank at <https://cstr.cn/31253.11.sciencedb.j00186.00235> and <https://www.doi.org/10.57760/sciencedb.j00186.00235>.

## Declarations

**Conflict of interest** The authors declare that they have no Conflict of interest.

## References

1. T. Alharbi, Distance metrics for digital pulse-shape discrimination of scintillator detectors. *Radiat. Phys. Chem.* **156**, 205–209 (2019). <https://doi.org/10.1016/j.radphyschem.2018.11.014>
2. R.A. Winyard, J.E. Lutkin, G.W. McBeth, Pulse shape discrimination in inorganic and organic scintillators. I. *Nucl. Instrum. Methods* **95**, 141–153 (1971). [https://doi.org/10.1016/0029-554X\(71\)90054-1](https://doi.org/10.1016/0029-554X(71)90054-1)
3. G. Liu, M.D. Aspinall, X. Ma et al., An investigation of the digital discrimination of neutrons and  $\gamma$  rays with organic scintillation detectors using an artificial neural network. *Nucl. Instrum. Methods Phys. Res. A* **607**, 620–628 (2009). <https://doi.org/10.1016/j.nima.2009.06.027>
4. G. Tian, X.P. Ou-Yang, H.G. Liang et al., Digital n/ $\gamma$  discrimination measurement of low intensity pulsed neutron. *Nucl. Tech.*

- (in Chinese) **38**, 060204 (2015). <https://doi.org/10.11889/j.0253-3219.2015.hjs.38.060204>
5. K.N. Li, X.P. Zhang, Q. Gui et al., Characterization of the new scintillator  $\text{Cs}_2\text{LiYCl}_6:\text{Ce}^{3+}$ . *Nucl. Sci. Tech.* **29**, 11 (2018). <https://doi.org/10.1007/s41365-017-0342-4>
  6. J.L. Cai, D.W. Li, P.L. Wang et al., Fast pulse sampling module for real-time neutron-gamma discrimination. *Nucl. Sci. Tech.* **30**, 84 (2019). <https://doi.org/10.1007/s41365-019-0595-1>
  7. J.M. Adams, G. White, A versatile pulse shape discriminator for charged particle separation and its application to fast neutron time-of-flight spectroscopy. *Nucl. Instrum. Methods.* **156**, 459–476 (1978). [https://doi.org/10.1016/0029-554X\(78\)90746-2](https://doi.org/10.1016/0029-554X(78)90746-2)
  8. J.H. Heltsley, L. Brandon, A. Galonsky et al., Particle identification via pulse-shape discrimination with a charge-integrating ADC. *Nucl. Instrum. Methods A* **263**, 441–445 (1988). [https://doi.org/10.1016/0168-9002\(88\)90984-9](https://doi.org/10.1016/0168-9002(88)90984-9)
  9. T. Sanderson, C. Scott, M. Flaska et al., Machine learning for digital pulse shape discrimination. *IEEE Nucl. Sci. Symp. Med. Imaging Conf. Rec.* **89**, 1–4 (2012). <https://doi.org/10.1109/NSSMIC.2012.6551092>
  10. P. Raj, A. Raman, *Handbook of Research on Cloud and Fog Computing Infrastructures for Data Science*, 1st edn. (IGI Global, Pennsylvania, 2018), pp. 1–400. <https://doi.org/10.4018/978-1-5225-5972-6>
  11. C. Fu, A. Di Fulvio, S.D. Clarke et al., Artificial neural network algorithms for pulse shape discrimination and recovery of piled-up pulses in organic scintillators. *Ann. Nucl. Energy* **120**, 410–421 (2018). <https://doi.org/10.1016/j.anucene.2018.05.054>
  12. S. Conroy, G. Ericsson, J. Nyberg et al., An artificial neural network based neutron-gamma discrimination and pile-up rejection framework for the BC-501 liquid scintillation detector. *Nucl. Instrum. Methods A* **610**, 534–539 (2009). <https://doi.org/10.1016/j.nima.2009.08.064>
  13. G. Andrew, C. Qi, A.D. Kaplan et al., Pulse pileup rejection methods using a two-component Gaussian mixture model for fast neutron detection with pulse shape discriminating scintillator. *Nucl. Instrum. Methods A* **988**, 164905 (2021). <https://doi.org/10.1016/j.nima.2020.164905>
  14. L.F. Liu, H. Shao, Study on neutron-gamma discrimination method based on the KPCA-GMM. *Nucl. Instrum. Methods A* **1056**, 168604 (2023). <https://doi.org/10.1016/j.nima.2023.168604>
  15. B. Blair, C. Chen, A. Glenn et al., Gaussian mixture models as automated particle classifiers for fast neutron detectors. *Stat. Anal. Data Min.* **12**, 479–488 (2019). <https://doi.org/10.1002/sam.11432>
  16. L.F. Liu, H. Shao, Study on neutron-gamma discrimination method based on the KPCA-GMM-ANN. *Radiat. Phys. Chem.* **203**, 110602 (2023). <https://doi.org/10.1016/j.radphyschem.2022.110602>
  17. F.P. Wang, M.H. Yang, J.Y. Wang et al., A comparison of small-batch clustering and charge-comparison methods for  $n/\gamma$  discrimination using a liquid scintillation detector. *Nucl. Instrum. Methods A* **1028**, 166379 (2022). <https://doi.org/10.1016/j.nima.2022.166379>
  18. M. Durbin, M.A. Wonders, M. Flaska et al., K-nearest neighbors regression for the discrimination of gamma rays and neutrons in organic scintillators. *Nucl. Instrum. Methods A* **987**, 164826 (2021). <https://doi.org/10.1016/j.nima.2020.164826>
  19. Y. Qian, W. Zhou, J. Yan et al., Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Remote Sens.* **7**, 153–168 (2015). <https://doi.org/10.3390/rs70100153>
  20. J.G. Wang, P. Neskovic, L.N. Cooper, Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern. Recogn.* **39**, 417–423 (2006). <https://doi.org/10.1016/j.patcog.2005.08.009>
  21. H. Arahmane, E.M. Hamzaoui, B. Maissa et al., Neutron-gamma discrimination method based on blind source separation and machine learning. *Nucl. Sci. Tech.* **32**, 18 (2021). <https://doi.org/10.1007/s41365-021-00850-w>
  22. X. Yu, J. Zhu, S. Lin et al., Neutron/gamma discrimination based on the support vector machine method. *Nucl. Instrum. Methods A* **777**, 80–84 (2015). <https://doi.org/10.1016/j.nima.2014.12.087>
  23. J. Wen, J. Zhu, T. Xue et al., Performance of linear classification algorithms on  $n/\gamma$  discrimination for  $\text{LaBr}_3:\text{Ce}$  scintillation detectors with various pulse digitizer properties. *J. Instrum.* **15**, P02004 (2020). <https://doi.org/10.1088/1748-0221/15/02/P02004>
  24. A. Fernandes, N. Cruz, B. Santos et al., FPGA code for the data acquisition and real-time processing prototype of the ITER radial neutron camera. *IEEE Trans. Nucl. Sci.* **66**, 1318–1323 (2019). <https://doi.org/10.1109/TNS.2019.2903646>
  25. N. Cruz, B. Santos, A. Fernandes et al., The design and performance of the real-time software architecture for the ITER radial neutron camera. *IEEE Trans. Nucl. Sci.* **32**, 1310–1317 (2019). <https://doi.org/10.1109/TNS.2019.2907056>
  26. A. Fernandes, R.C. Pereira, B. Santos et al., New FPGA based hardware implementation for JET gamma-ray camera upgrade. *Fusion. Eng. Des.* **128**, 188–192 (2018). <https://doi.org/10.1016/j.fusengdes.2018.02.038>
  27. L.M. Simms, B. Blair, J. Ruz et al., Pulse discrimination with a Gaussian mixture model on an FPGA. *Nucl. Instrum. Methods A* **900**, 1–7 (2018). <https://doi.org/10.1016/j.nima.2018.05.039>
  28. B. Blair, C. Chen, A. Glenn et al., Gaussian mixture models as automated particle classifiers for fast neutron detectors. *Stat. Anal. Data Min.* **12**, 479–488 (2019). <https://doi.org/10.1002/sam.11432>
  29. S. Beni, D. Krstic, D. Nikezic et al., Studies on unfolding energy spectra of neutrons using maximum-likelihood expectation-maximization method. *Nucl. Sci. Tech.* **30**, 134 (2019). <https://doi.org/10.1007/s41365-019-0662-7>
  30. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **39**, 1–22 (1977). <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>
  31. X.X. Wang, Q.L. Tang, X.F. Jiang, Simulation of  $n-\gamma$  pulse signal discrimination based on KNN classification algorithm. *Electron. Measurement Technol.* **45**, 164–170 (2022). <https://doi.org/10.19651/j.cnki.emt.2209025>
  32. S. Zhang, X. Li, M. Zong et al., Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans. Neur. Net. Lear.* **29**, 1774–1785 (2018). <https://doi.org/10.1109/TNNLS.2017.2673241>
  33. J. Goldberger, S. Roweis, G. Hinton et al., Neighbourhood components analysis. *Adv. Neural Inf. Process. Syst.* **17**, 513–520 (2005). <https://doi.org/10.1109/TCSVT.2013.2242640>
  34. Y.Q. Zhang, L.Q. Hu, G.Q. Zhong et al., Development of a high-speed digital pulse signal acquisition and processing system based on MTCA for liquid scintillator neutron detector on EAST. *Nucl. Sci. Tech.* **34**, 150 (2023). <https://doi.org/10.1007/s41365-023-01318-9>
  35. L. Chang, Y.D. Liu, L. Du et al., Pulse shape discrimination and energy calibration of EJ301 liquid scintillation detector. *Nucl. Tech.* **38**, 020501 (2015). <https://doi.org/10.11889/j.0253-3219.2015.hjs.38.020501>. (in Chinese)
  36. L. Stevanato, D. Cester, G. Nebbia et al., Neutron detection in a high gamma-ray background with EJ-301 and EJ-309 liquid scintillators. *Nucl. Instrum. Methods A* **690**, 96–101 (2012). <https://doi.org/10.1016/j.nima.2012.06.047>

37. G.Q. Zhong, In: Dissertation, Investigation of fusion neutron diagnostic technology on EAST device. University of Science and Technology of China (2017) **(in Chinese)** author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the