IL NUOVO CIMENTO **48 C** (2025) 50
DOI 10.1393/ncc/i2025-25050-1

COLLOQUIA: IWM-EC 2024

Charged-particle identification with advanced artificial intelligence approaches

D. DELL'AQUILA(1)(2) and M. RUSSO(3)(4)

(1) *Dipartimento di Fisica "Ettore Pancini", Università degli Studi di Napoli "Federico II" Napoli, Italy*
(2) *INFN, Sezione di Napoli - Napoli, Italy*
(3) *Dipartimento di Fisica e Astronomia "Ettore Majorana", Università degli Studi di Catania Catania, Italy*
(4) *INFN, Sezione di Catania - Catania, Italy*

received 26 November 2024

Summary. — Modern nucleus-nucleus collision experiments require the use of advanced particle identification techniques. However, similar tasks are often time-consuming, enhancing the complexity of the data analysis process. We develop a novel approach capable to automatically identify charge and mass of detected ions with almost zero human supervision. Our method uses evolutionary computing and clustering algorithms and exploits previously developed analytical functionals to provide physics constraints. The new algorithm is successfully tested on ΔE-E telescopes based on annular silicon strip detectors and could be integrated in online and offline analysis pipelines of existing detection arrays.

1. – Introduction

The physics interpretation of nuclear collision data often relies on the detailed knowledge of the *collision event*, which, in turn, requires detecting and *identifying* particles and fragments produced in the collision. In charged particle experiments, the identification task consists in assigning a given reconstructed particle its precise nature, *e.g.*, charge (Z) and mass (A) of an ion. This is equivalent to grouping data into some meaningful physics classes: particle-identification (PID) in nuclear physics can be therefore seen as a *data classification* problem. However, PID procedures traditionally used in nuclear physics are typically time-consuming, especially for complex multi-detectors characterized by numerous detection units [1-14], demanding for new, faster, methodologies for data analysis.

Although the methodology discussed in this manuscript could be in principle applicable to numerous other experimental methods used for PID, in the present work we focus exclusively on the ΔE-E identification technique with stacks of two detection layers. In similar arrays, if organized in a 2D correlation plot, data recorded by pairs of independent

layers assemble into bi-dimensional non-overlapping clusters, each representing a certain (Z,A) class. Consequently, the problem of PID using the ΔE-E technique is equivalent to the *clusterization* in a bi-dimensional space.

In the literature, numerous algorithms based on Cluster Analysis (CA) or Vector Quantization (VQ) have been developed (see, for example, [15-18]), with optimal performance in *standard* cluserization cases, where one needs to obtain clusters of nearly equal *distortion*. Unfortunately, these algorithms are not directly applicable to the bi-dimensional assemblies typical of the ΔE-E technique because the peculiarities of the stopping power of ions in the matter generates clusters with a large variability of size and dispersion [19]. For this reason, only a few works previously tried to exploit CA/VQ methods in nuclear data classification problems (see, *e.g.*, [20]).

Most of the approaches commonly used for the analysis of ΔE-E data involve human-supervised techniques, where the operator manually extracts information by visually inspecting bi-dimensional distributions of data, which is then used as input for supervised learning procedures. Among the latter, particularly popular are error minimization algorithms based on mathematical models [21, 22], which contain Z and A-values explicitly, even if artificial neural networks have been also proposed [23]. Mathematical models offer the advantage of requiring information only for a reduced number of clusters, allowing extrapolation to the entire dynamic range of the detector and reducing the effort for the operator. The amount of information required for these algorithms can be further reduced, as an example, following the prescriptions of ref. [24], which have been effectively tested for the FAZIA multi-detector.

In this work, we describe a novel algorithm for PID in multi-detector data based on advanced artificial intelligence approaches typically used in data classification tasks. The novel algorithm combines Evolutionary Computing (EC) and Cluster Analysis and is capable to classify data in a physically meaningful way reducing the time required for this task to a few minutes or hours of CPU-time with nearly zero supervision by the operator. The search performed in the algorithm consists in two separate levels: the upper level performs a global search through an EC algorithm that treats each solution as an individual of a given population and applies some suitable evolutionary criteria; the lower level, used as a local *hill-climbing* operator for the EC process, performs a fast local search through a suitable VQ algorithm to speed-up the process.

2. – The algorithm

The algorithm proposed in this work implements a *constrained evolutionary clustering* approach. The goal is that of partitioning the bi-dimensional ΔE-E distribution separating data in the various (Z, A) classes. To obtain a physically meaningful classification, a given partitioning is described by an analytical functional with an explicit dependence on Z and A. For the present work, we used the functional proposed in ref. [21],

$$\begin{aligned}\Delta E =& f_P(E,Z,A)\\ =& \left[(P_0E)^{P_1+P_2+1}+(P_3Z^{P_4}A^{P_5})^{P_1+P_2+1}+P_6Z^2A^{P_1}(P_0E)^{P_2}\right]^{(1/(P_1+P_2+1))},\end{aligned} \tag{1}$$

being $P=\{P_0,P_1,P_2,P_3,P_4,P_5,P_6\}$ a given set of free parameters. For a given choice of the parameters P_i one finds a different partitioning of the ΔE-E distribution. The upper and lower levels of the algorithm, described below, are aimed to find the optimal

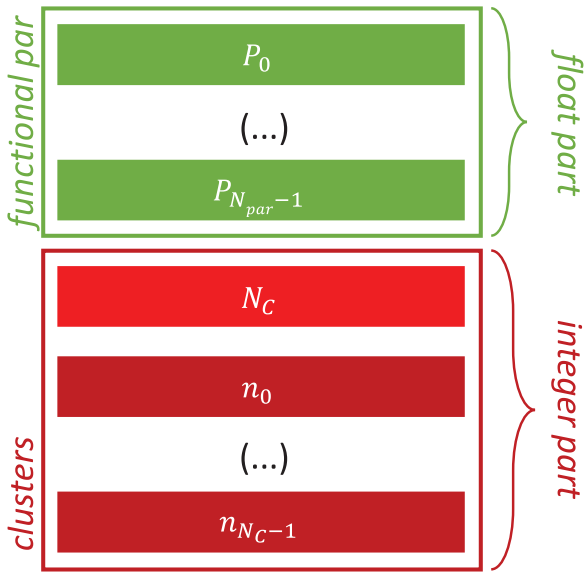

Fig. 1. – Representation of an individual, *e.g.*, a solution to the PID problem. N_{par} is the number of the free parameters $\{P_i\}$ required by the functional and N_C is the number of clusters identified in the ΔE-E distribution.

P set of the parametrization (1). After having found the optimal free parameters, the identification is done through linearization of the ΔE-E distribution in the usual way [22].

2˙1. *Upper Level: the evolutionary computing part.* – EC is a scientific field that concerns the resolution of optimization problems through concepts and ideas derived from natural selection in biological systems [25]. EC is widely applied in numerous domains of science (see, *e.g.*, [26-33] and references therein). In the present work, the EC part foresees the *evolution* of solutions of the PID problem. For a given ΔE-E distribution to analyze, a solution is encoded as schematically described in fig. 1. In such an encoding, $\{P_i\}$ is a set of N_{par} free parameters required by the functional of eq. (1), N_C is the number of identified clusters, and $\{n_i\}$ are the identified species in the given ΔE-E distribution. The parameters $\{P_i\}$, N_C, $\{n_i\}$ are optimized in the upper level according to the following programming scheme, derived from the Darwinian Theory of Evolution:

1) A set of possible solutions according to the encoding of fig. 1 is randomly generated. Each of such solutions is an *individual* of a given *population*.

2) A numerical value, called *fitness*, is associated to each individual. The fitness quantifies how much a given solution is *suitable* to the problem to solve. In the present application, higher fitness corresponds to better solutions. The fitness function chosen in this work accounts for the *distortion* of each cluster, *i.e.*, the average dispersion of points around the cluster center, as described in ref. [34].

3) Until the average fitness in the population is maximized, the following steps are iterated:

a) Two individuals are selected (*parents*) to be used as a starting point for the generation of a new individual (*offspring*). The selection criterion is stochastic and accounts for the fitness of the individuals.

b) The offspring is obtained through a mechanism of parents encoding recombination (*crossover*, see ref. [34] for more details). In this phase, the chromosomes of the parents, *i.e.*, their encoding, are suitably combined to generate new individuals. With a suitably low probability, some portions of the derived encoding are randomly varied (*mutation*). Such a process has a crucial importance as it allows to introduce missing genetic code and to keep genetic diversity in the population. The fitness is calculated for the new individual.

c) The new individual replaces another individual in the original population, which is randomly chosen with a probability larger for low fitness individuals.

2˙2. *Lower level: the clustering analysis part.* – This level is invoked when a promising individual, *i.e.*, an individual with a *good* fitness is produced by the upper level (sect. **2˙1**). This part allows to rapidly optimize the parameters derived in the upper level through a procedure derived from standard CA/VQ algorithms.

The goal of a typical VQ algorithm consists in the representation of a given set of vectors $\mathbf{x} \in X \subseteq \Re^k$ through a set, $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_C}\}$, of N_C reference vectors in $\Re^k$. Each reference vector is called *codeword*, while a set of codewords $Y = \{\mathbf{y}_i\}$ is called *codebook*. In VQ, a codebook is derived to represent the entire initial dataset, while CA, equivalently, deals with identifying clusters of data. In the present, peculiar, application, X contains the bi-dimensional data points of a given ΔE-E distribution, while Y contains hyperbolic loci predicted by the functional of eq. (1).

A given solution of the VQ problem can be represented by a function $q : X \longrightarrow Y$, *i.e.*, a function that associates a given bi-dimensional ΔE-E point to the *nearest* hyperbolic locus. The determination of q allows to obtain a partition $\mathcal{S}$ of the original dataset X constituted by N_C subsets, S_i, called *cells*:

$$\mathcal{S} = \{S_i;\ \ i = 1, \ldots, N_C\}. \tag{2}$$

The Quantization Error (QE) can be then used to quantify the *quality* of a given partition of the ΔE-E distribution. QE is the value deduced by $d(\mathbf{x}, q(\mathbf{x}))$, being d a suitable distance operator between ΔE-E data points belonging to X and hyperbolic loci from eq. (1), as defined in ref. [34]. The performance of a given quantizer q is evaluated through the Mean QE (MQE)

$$\mathrm{MQE} \equiv D(Y, \mathcal{S}) = \frac{1}{N_P} \sum_{i=1}^{N_C} D_i, \tag{3}$$

where N_P is the number of data points in the ΔE-E distribution, and D_i is the total distortion of the i-th cell, being it defined by the following equation:

$$D_i = \sum_{n:\mathbf{x}_n \in \mathcal{S}_i} d(\mathbf{x}_n, q(\mathbf{x}_n)). \tag{4}$$

Given the definitions elucidated above, the lower level of the algorithm proceeds with the following steps:

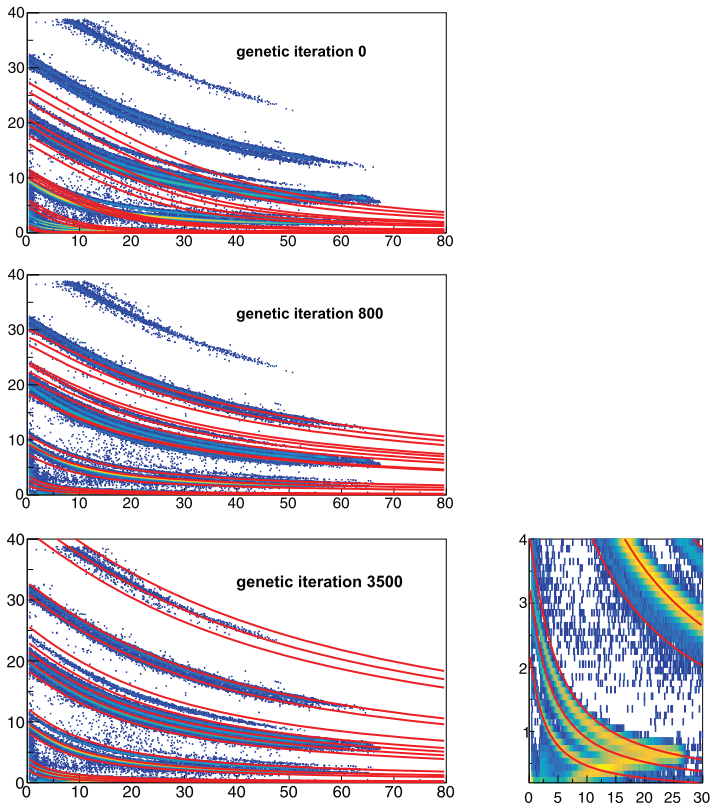

Fig. 2. – Representation of the best individual in the population after 0, 800, and 3500 genetic iterations. The bottom right panel is a zoom to light isotopes after 3500 iterations.

1) *Initialization*: the result of the EC part is chosen as the initial codebook.

2) *Partition calculation*: given the codebook determined in the previous step, the ΔE-E data are grouped into clusters, *i.e.*, each data point is associated with a given isotope following the Z-A-hyperbolic loci.

3) *Termination condition*: the MQE at the current iteration $D_{\rm curr}$ is compared with the one obtained in the previous iteration $D_{\rm prev}$. If the ratio $|D_{\rm prev} - D_{\rm curr}|/D_{\rm prev}$ is less than a prefixed threshold (ε) then the algorithm ends; otherwise, it continues with the next step;

4) *Codebook calculation*: by using the partition calculated in step 2), a new codebook is calculated based on a fit of clustered data with eq. (1).

5) Return to step 2).

3. – Results

The performance of the present algorithm in identifying isotopes via the ΔE-E technique has been tested on experimental data obtained with a detector made of six

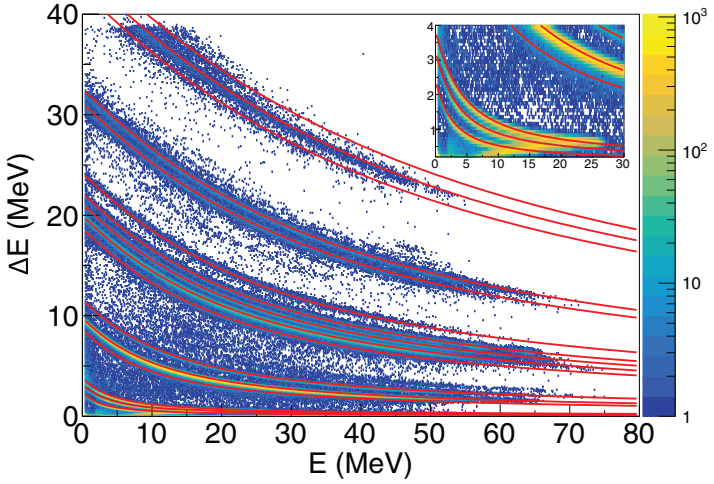

Fig. 3. – Functional used for PID with the parameters obtained after the CA part is invoked.

wedge-shaped ($70\,\mu$m/$1500\,\mu$m) telescopes arranged in a *lampshade*. Each telescope was segmented into 16 annular strips. Consequently, the number of independent bi-dimensional distributions to classify is 96. The experiment was performed at the ISAC-II rare-isotope ion beam facility at the TRIUMF laboratory of Vancouver (Canada). A ^{9}Li accelerated beam was delivered on a LiF target at an energy of 65 MeV. $^{9}\mathrm{Li} + {}^{6}\mathrm{Li}$, $^{9}\mathrm{Li} + {}^{7}\mathrm{Li}$ and $^{9}\mathrm{Li} + {}^{19}\mathrm{F}$ collisions were investigated. They resulted in the production of several ions especially in the range $1 \leq Z \leq 5$. This experiment represents a valid benchmark for advanced PID methods because all the detection units were capable to resolve nearly all Z and A values in the range of interest.

Figure 2 shows the capabilities of the EC level of the algorithm in maximizing the fitness. In the figure, we showed the best individual in the population after 0 (top panel), 800 (middle panel), and 3500 (bottom panel) iterations. The right panel contains a zoom of the low-Z region with the best individual after 3500 iterations. Red lines represent the individual, with each line representing the *center* of the corresponding hyperbolic locus, each corresponding to a given codeword. After 0 iterations, the best individual in the population (which, we remind, is generated randomly) gives an unsatisfactory classification. A better performance for light isotopes is obtained after about 800 iterations. However, the distortion of the heavier isotope is much larger than the rest. The solution obtained after 3500 genetic iterations is very close to a good maximum of the fitness function. In general, a number of 3500 iterations is found to be largely sufficient to obtain a fully satisfactory codebook for all cases explored in this paper.

A further improvement of the identification shown in fig. 2 can be obtained invoking the CA part on the best individual obtained after 3500 iterations. The result of the clustering algorithm is shown in fig. 3. Semi-quantitatively, it can be easily seen that the various codewords significantly better approximate observed clusters with respect to the pure EC individual improvement. This a result of the CA optimization. The result looks satisfactory for all clusters. The improvement is seen especially by comparing the results for hydrogen isotopes before and after the CA is invoked (figs. 2 and 3). In addition, despite the punch-through effect visible for hydrogen isotopes, low Z codewords are in fully satisfactory agreement with clusters corresponding to ^{1}H, ^{2}H and ^{3}H isotope,

testifying the robustness of the approach.

The analysis discussed in this section involved the data from 96 independent ΔE-E distributions. The task was completed in about 440 seconds on a commercial Intel i7-9700K (8 cores) processor at a frequency of 3.4 GHz, without human supervision.

4. – Conclusions and perspectives

This paper describes a novel approach for the analysis of nuclear physics data, capable to automatically, and with minimal human supervision, perform the task of PID. Even if the approach is developed and tested for ΔE-E distributions obtained with silicon telescopes, the idea could be implemented for various other identification techniques, given a specific mathematical parametrization of the observed loci.

The new algorithm exploits evolutionary computing and clustering analysis, and is constrained with existing functionals for PID.

For the data used in the present work, the new method allowed to accurately identify charge and mass of all resolved species for 96 individual detection cells without human supervision.

In the future, the present algorithm could be further developed to be included in the online and offline analysis pipelines of existing multi-detectors [35-40].

* * *

The authors would like to thank the experimental nuclear physics group of the Ruder Bošković Institute for making the data used to probe the performance of the algorithm available. The authors would also like to show their gratitude to Dr. Neven Soić and Dr. Nikola Vukman for useful discussions and to Dr. Dora Tot for reviewing the manuscript.

REFERENCES

[1] Pouthas J. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **357** (1995) 418.
[2] Pagano A., *Nucl. Phys. News*, **22** (2012) 25.
[3] Wuenschel S. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **604** (2009) 578.
[4] Davin B. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **473** (2001) 302.
[5] Wallace M. S. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **583** (2007) 302.
[6] Dell'Aquila D. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **929** (2019) 162.
[7] Bougault R. *et al.*, *Phys. Rev. C*, **97** (2018) 024612.
[8] Borderie B. *et al.*, *Phys. Lett. B*, **782** (2018) 291.
[9] Acosta L. *et al.*, *J. Phys.: Conf. Ser.*, **730** (2016) 012001.
[10] Bishop J. *et al.*, *Phys. Rev. C*, **100** (2019) 034320.
[11] Lukasik J. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **709** (2013) 120.
[12] Marquínez-Duran G. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **755** (2014) 69.
[13] Valdré S. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **930** (2019) 27.
[14] Camaiani A. *et al.*, *Phys. Rev. C*, **103** (2021) 014605.
[15] Patanè G. and Russo M., *Neural Netw.*, **14** (2001) 1219.
[16] Patanè G. and Russo M., *IEEE Trans. Neural Netw.*, **13** (2002) 1285.
[17] Baraldi A. and Alpaydin E., *IEEE Trans. Neural Netw.*, **13** (2002) 662.
[18] Frigui H. and Krishnapuram R., *IEEE Trans. Pattern Anal. Mach. Intell.*, **21** (1999) 450.
[19] Benkirane A. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **355** (1995) 559.
[20] Wirth R. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **717** (2013) 77.
[21] Tassan-Got L., *Nucl. Instrum. Methods Phys. Res. B*, **194** (2002) 503.
[22] Le Neindre N. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **490** (2002) 251.

[23] Iacono-Manno C. M. and Tudisco S., *Nucl. Instrum. Methods Phys. Res. A*, **443** (2000) 503.
[24] Gruyer D. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **847** (2017) 142.
[25] Koza J. R., *Genetic Programming: On the Programming of Computers by Natural Selection* (MIT Press, Cambridge, MA, USA) 1992.
[26] Gotmare A., Bhattacharjee S. S., Patidar R. and George N. V., *Swarm Evol. Comput.*, **32** (2017) 68.
[27] Darwish A., Hassanien A. and Das S., *Artif. Intell. Rev.*, **53** (2020) 1767.
[28] Russo M., *Swarm Evol. Comput.*, **27** (2016) 145.
[29] Campobello G., Dell'Aquila D., Russo M. and Segreto A., *Appl. Soft Comput.*, **94** (2020) 106488.
[30] Buccheri E., Dell'Aquila D. and Russo M., *Diabetes Res. Clin. Pract.*, **174** (2021) 108722.
[31] Buccheri E., Dell'Aquila D. and Russo M., *Obes. Med.*, **31** (2022) 100398.
[32] Buccheri E., Dell'Aquila D., Russo M., Chiaramonte R. and Vecchio M., *J. Geriatr. Phys. Ther.*, **00** (2024) 1.
[33] Barone F. P., Dell'Aquila D. and Russo M., *Mach. Learn.: Sci. Technol.*, **4** (2023) 045054.
[34] Dell'Aquila D. and Russo M., *Comput. Phys. Commun.*, **259** (2021) 107667.
[35] Lopez O. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **884** (2018) 140.
[36] Ciampi C. *et al.*, *Phys. Rev. C*, **106** (2022) 024603.
[37] Dell'Aquila D. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **877** (2018) 227.
[38] Dell'Aquila D. *et al.*, *Nuovo Cimento C*, **39** (2016) 272.
[39] Sweany S. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **1018** (2021) 165798.
[40] Jin Y. *et al.*, *Phys. Rev. Lett.*, **127** (2021) 262502.