AI-Powered Analysis for Hadronic Particle Detection in ALICE Open Data

Estefania Moreno León

[1] Universidad de Sonora, Master of Engineering, México
imstefleon@gmail.com

Abstract. The integration of Artificial Intelligence (AI) techniques, particularly Machine Learning (ML), is revolutionizing data analysis in high-energy physics experiments. This study presents the development and implementation of an advanced AI-driven analysis environment for hadronic particle detection using open data from the ALICE experiment at CERN. The research focuses on leveraging ML algorithms, including Deep Learning, ensemble methods, anomaly detection, to enhance particle classification and identification. The methodology involves preprocessing and analyzing data from lead-lead (Pb-Pb) collisions, applying supervised and unsupervised ML techniques to optimize particle detection accuracy. The study explores the use of neural networks for particle classification and prediction, aiming to improve the identification of subatomic interactions in extreme conditions. Additionally, real-time inference through Kafka integration enhances data processing efficiency. Results demonstrate that AI-based approaches significantly improve particle classification performance compared to traditional methods. The findings highlight the potential of ML to refine experimental analysis, reduce uncertainties, and provide new insights into fundamental interactions in particle physics. This research underscores the transformative role of AI in modern high-energy physics, paving the way for future advancements in data-driven scientific discovery.

Keywords: Artificial Intelligence, Machine Learning, Hadronic Particle Detection, ALICE Experiment, Particle Classification, Deep Learning

1 Introduction

1.1 Particle physics and the study of exotic particles

Particle physics explores the basic building blocks of matter and their interactions, aiming to uncover the universe's mysteries at its most fundamental level [1]. These exotic particles possess unique properties and behaviors that challenge our current understanding of particle physics. Their study not only expands our knowledge of the subatomic world but also offers insights into the underlying principles governing the cosmos. The detection and characterization of exotic particles, such as those in High Energy Physics (HEP), require sophisticated experimental techniques and analytical methodologies. Traditional approaches often rely on meticulous data analysis and theoretical modeling to identify these elusive entities within the sea of particle interactions. However, recent

https://doi.org/10.61728/AE20254919

advancements in technology and computational methods have opened new frontiers in particle physics research [1].

1.2 Role of AI in Hadronic Particle Detection

Artificial Intelligence (AI) has emerged as a powerful tool for advancing hadronic particle detection in high-energy physics experiments. Traditional methods rely on rule-based classification and statistical models, which, while effective, often struggle with the complexity and sheer volume of data generated in large-scale experiments like ALICE at CERN [2]. The integration of Machine Learning (ML) techniques, including supervised, unsupervised, and reinforcement learning, offers a more adaptive and scalable approach to particle identification. Deep learning models, particularly neural networks, have demonstrated remarkable accuracy in distinguishing between hadronic particles such as pions, kaons, and protons based on detector signals. Furthermore, AI-driven anomaly detection methods enhance the identification of rare or unexpected particle interactions, improving the reliability of physics analyses [3]. The real-time processing capabilities of AI, combined with streaming frameworks like Kafka, enable rapid data analysis, reducing computational bottlenecks in experimental workflows. By leveraging AI, researchers can refine particle classification, reduce uncertainties, and gain deeper insights into the fundamental interactions governing high-energy collisions. This study explores the application of AI in optimizing hadronic particle detection, demonstrating its potential to applicability data analysis in modern particle physics.

2 ALICE Experiment and Open Data

2.1 Overview of Alice

The ALICE (A Large Ion Collider Experiment) detector at CERN is a specialized experiment designed to study heavy-ion collisions and investigate the properties of quark-gluon plasma (QGP), a state of matter that existed in the early universe microseconds after the Big Bang [1]. Unlike other LHC experiments, ALICE is optimized for tracking and identifying a wide range of particles emerging from these collisions, providing crucial insights into the behavior of strongly interacting matter under extreme conditions [4]. The experiment utilizes a complex system of detectors, including the Time Projection Chamber (TPC), Inner Tracking System (ITS), and Time-of-Flight (TOF) detectors, which enable precise reconstruction of particle trajectories and identification of hadronic species such as pions, kaons, and protons [3].

One of the primary objectives of ALICE is to analyze Pb-Pb collisions at ultra relativistic speeds, allowing scientists to study deconfined quarks and gluons and their interactions before recombining into hadrons [2]. The experiment's advanced particle identification (PID) techniques, which combine dE/dx measurements with velocity-based classification, provide a solid foundation for developing artificial intelligence (AI)-driven methodologies for automated particle recognition [3].

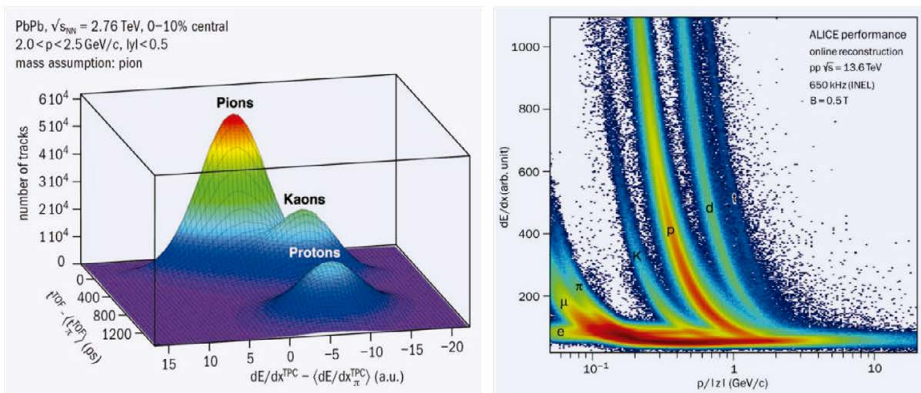

Fig. 1. Particle Identification (PID) (Left) Composition of Pions, Kions and Protons (Right) Reconstruction of the data [1].

By integrating AI and machine learning (ML) into the analysis pipeline, researchers aim to enhance detection accuracy, optimize real-time processing, and explore rare and exotic particle states more efficiently [2]. These advancements position ALICE as a key experiment in high-energy physics, bridging traditional particle detection methods with AI-driven approaches to push the frontiers of knowledge in nuclear and particle physics.

2.2 Data Characteristics and Challenges

One major challenge in implementing AI-based particle detection within ALICE is the integration of machine learning frameworks into the existing analysis infrastructure. While many machine learning models are developed in Python, ALICE's core analysis framework, O2, primarily operates in C++ and relies heavily on the ROOT library [1]. This language barrier complicates the seamless deployment of AI models, necessitating solutions like Open Neural Network Exchange (ONNX) to bridge Python and C++ environments [5]. ONNX provides a standardized format for storing and deploying machine learning models, ensuring compatibility and efficiency in real-time particle classification (see Figure 2).

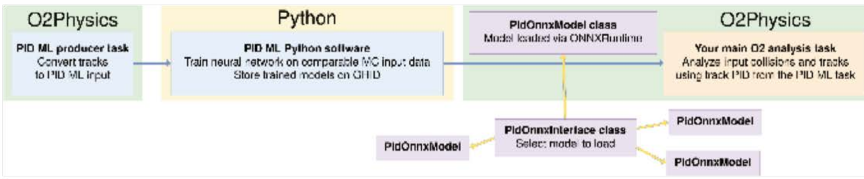

Fig. 2. Scheme of the current ML particle identification workflow in O2 in ALICE3 [5].

Another significant issue is handling incomplete data. Since multiple detectors independently measure different particle properties, a single particle may be detected by some subsystems while remaining undetected by others. Factors such as detector malfunctions, low transverse momentum, or signal loss contribute to missing data points, which can degrade the performance of AI models [3]. Standard approaches to missing data handling include case deletion, which discards incomplete samples but reduces dataset diversity, and imputation techniques, which estimate missing values using statistical methods or machine learning predictions [3]. Both methods introduce biases that may affect the accuracy of particle identification.

Experimental Environment

The processing setup used is constituted by Ubuntu 18.04 Linux system in a Windows computer has a CPU of 2.5 ghz a host memory 34 B and a hard disk capacity of 1TB. The development language uses version 3.5 of Aliroot in Terminal and Pyroot.

3 Methodology for Developing Artificial Intelligence

The development of an AI-driven particle detection system for the ALICE 3 experiment followed a structured methodology to ensure efficient data collection, processing, and model evaluation (see Figure 3). The first phase involved collecting Open Access Data from CERN, including Pb-Pb collision records from 2014 to the present, and retrieving LHC Grid Variables relevant to particle interactions. Additionally, historical data was accessed through ROOT tools, allowing for further analysis of lead-ion collisions. Next, a Machine Learning (ML) program was developed to assess the effectiveness of particle identification in high-energy physics (HEP). This step included data preparation, cleaning, and structuring, followed by the implementation of a decision tree and ML algorithms to enhance classification accuracy. Finally, data analytics techniques were used to identify key attributes and clusters in lead-ion collisions, ensuring an optimal feature selection for the AI system. The Monte Carlo methodology was incorporated to improve simulation-based validation, reinforcing the reliability of the proposed approach. This structured plan ensures consistency and precision in AI-assisted particle detection, aligning with CERN's best practices in high-energy physics research.

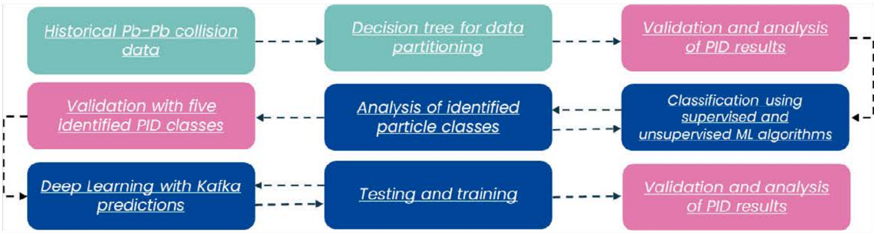

Fig. 3. Scheme of the current ML particle identification workflow in O2 in ALICE3.

3.1 Understanding and Evaluating the Pb-Pb Open Dataset

The development of an AI-based particle detection system relies on a structured approach to handling and analyzing Pb-Pb collision data from the ALICE 3 experiment at CERN. The dataset, obtained through Open Data, consists of 876,363 entries in CSV format, containing historical records of particle interactions. These CSV files were generated using custom macros developed for AliRoot, which enabled the retrieval and extraction of historical data stored in ROOT format. By leveraging these macros, the system efficiently accessed and processed key collision parameters, such as particle trajectories, energy deposits, and timing information, ensuring a structured dataset for AI model training. For enhanced particle classification, the dataset labels were mapped to their respective physical particles, grouping them as follows:

- "π" (pions and antipions)
- "e" (electrons and positrons)
- "K" (kaons and antikaons)
- "p" (protons and antiprotons)
- "d" (deuterium and antideterium)

This dataset includes crucial attributes for particle identification, such as particle trajectories, energy deposits, and timing information. Key variables include event and particle numbers, multiple energy measurements, collision origin coordinates (X, Y, Z), detector signals (ITS, TPC, TOF), electric charge, and momentum (P). These features are essential for reconstructing collision events and distinguishing between particle types.

4 Data Analysis for Particle Identification

Experimental environment

The processing setup used is constituted by Windows system has a CPU of 3.5 ghz a host memory of 34GB and a hard disk capacity of 500 Gb. The development language uses are Python and MATLAB, Octave for the preprocessing of data.

4.1 Filtering

Ensuring high-quality and structured data is crucial for accurate analysis and machine learning performance. The data filtering process eliminates irrelevant data, normalizes values, and structures datasets for ML compatibility. Additionally, data cleaning addresses outliers, missing values, and input errors, improving dataset integrity and model accuracy. This is particularly important when analyzing TPC detector signals vs. momentum (P) (see Figure 5), where many data points cluster near zero. Understanding the X, Y, Z origin coordinates of each detected particle is essential for accurately reconstructing collision events.

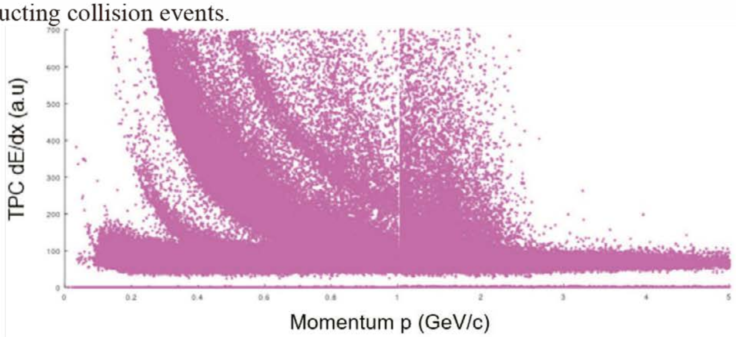

Fig. 5. Total Particles in TPC vs. Momentum for Positive Particles (Unfiltered Data).

During a Pb-Pb collision, protons interact at a specific point near the center of the detector, known as the Primary Collision Vertex [see Figure 6]. The methodology in this study focuses on sub-particles generated at this primary vertex. However, many of these particles scatter and undergo secondary interactions, creating additional sub-particles in what are known as secondary collisions. As particles move through the detector's gas chambers, they leave recorded trajectories, allowing for precise calculation of their point of origin. To accurately reconstruct collision dynamics, each dataset includes three-dimensional origin coordinates (X, Y, Z) of each detected particle, enabling the distinction between primary and secondary collision products.

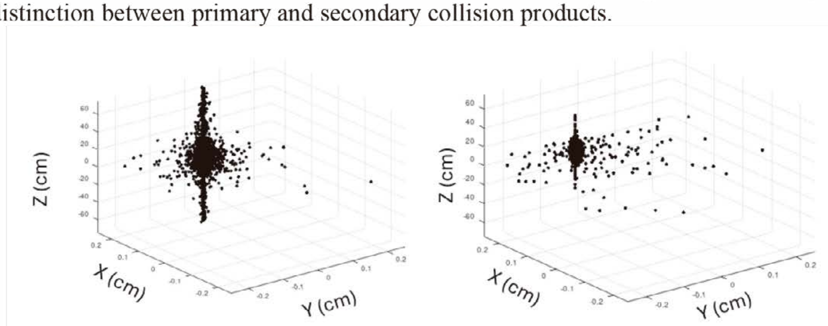

Fig. 5. Collision Vertex Positioning: Detected Particles (Left) and Primary Collisions (Right).

To determine whether a particle originates from the main collision or a secondary interaction, event vertices were extracted from ROOT files and visualized in 3D (see Figure 5). The right side represents main collision vertices, while the left side shows detected particle vertices, revealing a significant presence of secondary collisions. A MATLAB-based filtering algorithm was developed to retain only primary collision particles (see Figure 6). The filtering process calculates the spatial distance between detected particles and the main collision vertex, ensuring data integrity by verifying detector signals. Initial filtering based solely on distance was insufficient, leading to an optimized approach that improved classification accuracy. To enhance data quality, a comprehensive filtering process was applied, reducing the dataset to 147,950 samples across 13 attributes with separate files for negative and positive particles. This refinement increased data clarity and consistency, ensuring that only relevant collision data is used for analysis.

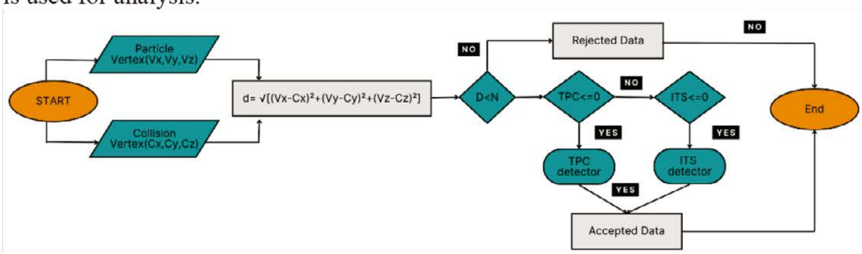

Fig. 6. Decision Tree Algorithm for Primary Data Selection.

5 Unsupervised Machine Learning Approaches

The use of clustering algorithms such as K-Means, Gaussian Mixture Models (GMMs), DBSCAN, and Spectral Clustering has improved particle classification by identifying patterns and grouping particles based on shared characteristics. However, applying these methods to large-scale datasets posed challenges due to high dimensionality and complex data distribution, affecting clustering efficiency and classification accuracy. These limitations highlight the need for adaptive or hybrid approaches to enhance large-scale particle classification in high-energy physics.

5.1 K-Means

One of the main issues is the limited capability of conventional clustering algorithms, such as KMeans, to capture nonlinear structures or non-convex shapes in the data. These algorithms tend to cluster data into spherical or hyperspherical clusters, which can result in poor segmentation of nonlinear clusters or the under-segmentation of elliptical clusters [8]. As shown in the following result in Figure 7, clustering of 5 groups and 20 groups was achieved, using Momentum and TPC in positive lead-lead particles.

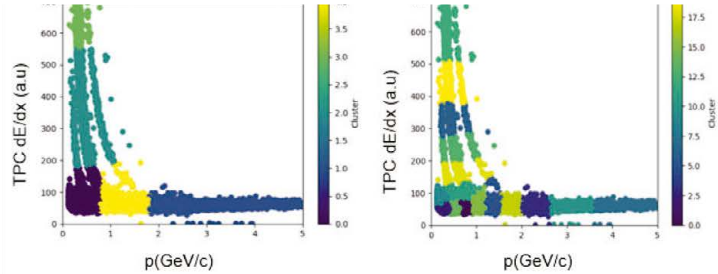

Fig. 7. Clustering graph of positive particles with KMeans left 5 groups and right 20 groups.

It was concluded that the presence of noise or outliers in the data can further hinder the accurate identification of elliptical clusters. Since clustering algorithms can be sensitive to outliers, they negatively affect the quality of the identified clusters.

5.2 Gaussian Mixture Models (GMMs)

The Gaussian Mixture Model (GMM) clustering technique was applied in Python to classify positive lead-lead particles, providing valuable insights into the dataset's structure. By assigning 25 labels, the algorithm aimed to assess how well it could capture underlying patterns and distributions within the data. The flexibility of the Gaussian model allowed for a more adaptive representation of nonlinear and multidimensional structures, improving the classification accuracy and robustness of the system. As illustrated in Figure 9, the results highlight clear classification patterns, particularly when analyzing TPC vs. Momentum (P), revealing distinct group formations.

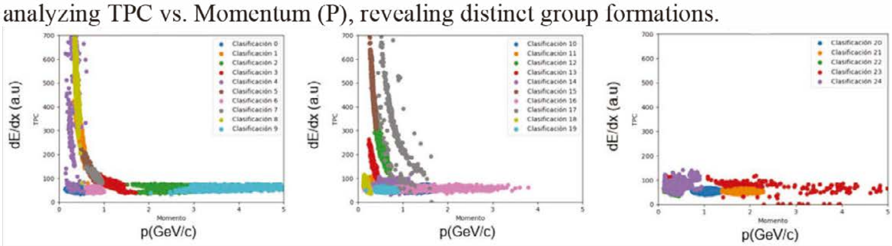

Fig. 8. Clustering graph of positive particles with GMM 25 groups. Divided into Three Visualizations.

To refine the clustering analysis, the classifications were divided into three subsets based on label ranges (0–9, 10–19, and 20–24), which provided a more granular understanding of how the algorithm categorized data, as shown in Figure 8. Additionally, the dataset labels were mapped to their respective physical particles resulting in a more structured and interpretable clustering output. Although Gaussian Clustering produced promising results, manual adjustments were necessary to correctly group positive particles due to noise and irregular data distribution challenges. Since the Gaussian model assumes a normal (spherical) distribution, it struggled to handle complex, high-energy physics data, making it less effective in cases where non-uniformity and outliers were present.

5.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is effective for irregularly shaped clusters without requiring a predefined number of clusters. However, its success depends on EPS (neighborhood radius) and MinPts (minimum points per cluster), which were optimized using the k-distance method (see Figure 9).

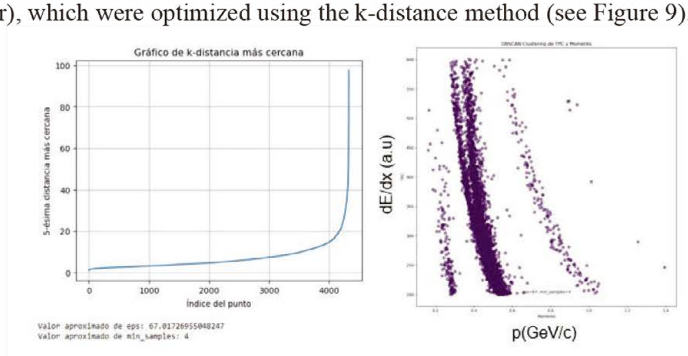

Fig. 9. Nearest Neighbor Distances and Clustering of Positive Particles Using DBSCAN (EPS = 67, Min Samples = 4).

DBSCAN is effective for irregularly shaped clusters but depends on EPS (neighborhood radius) and MinPts (minimum points per cluster) for optimal segmentation. In this study, EPS = 67.01726 and MinPts = 4 improved clustering but significantly increased computational costs. Partitioning the dataset did not improve classification, as DBSCAN misclassified most data as noise due to density similarities. The Nearest Neighbor algorithm was also tested but failed due to outlier sensitivity and high-density regions. DBSCAN's high computational demands and inefficiency in large-scale high-energy physics data make it impractical without further optimizations or alternative clustering techniques. The primary difficulty in implementing DBSCAN lies in its handling of datasets with a vast number of irregular and scattered samples. The method requires substantial computational power and memory, making it impractical for large-scale high-energy physics data analysis without further optimizations or alternative clustering strategies.

5.4 Spectral Clustering

Spectral Clustering is effective for high-dimensional data as it projects information into a lower-dimensional space using spectral decomposition. A key advantage is its ability to customize affinity measures, improving clustering accuracy. Three affinity parameters were tested:

- Nearest Neighbors: Limits calculations, making it suitable for large datasets.
- Radial Basis Function (RBF): Uses a Gaussian kernel to handle nonlinear relationships.
- Precomputed Affinity: Uses a custom similarity matrix for tailored clustering

Each approach was tested to evaluate its effectiveness in detecting clusters within Pb-Pb collision data and the best results were with RBF (see Figure 8).

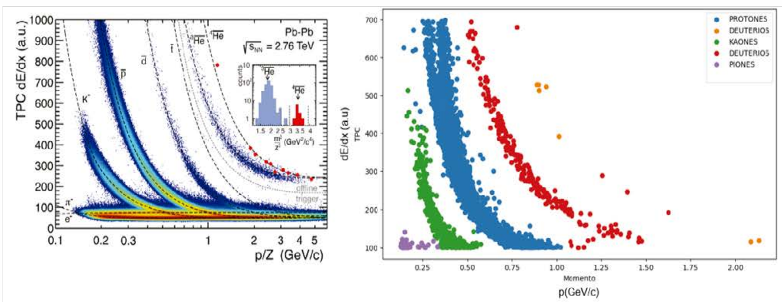

Fig. 10. PID ALICE: TPC vs. Momentum (P) in Pb-Pb Collisions [1] vs. Spectral Clustering Using RBF in the 100–700 TPC Range.

- Nearest Neighbors Approach: Created four clusters in the TPC signal range 200–700, but separations were affected by detector errors and energy loss.
- RBF Kernel & Precomputed Affinity: Improved classification accuracy and reduced computation time, effectively identifying particle interactions in the 100–700 TPC range. However, memory constraints limited analysis of the high-density 0–100 region (see Figure 9).

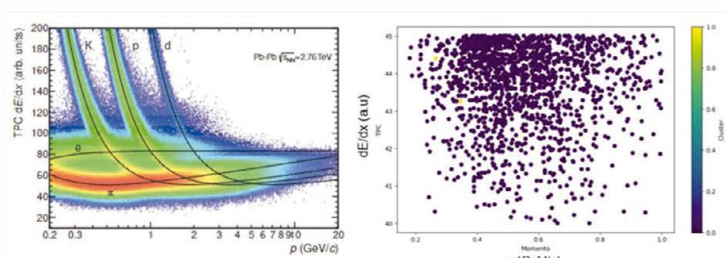

Fig. 11. ALICE TPC vs. Momentum (P) in Pb-Pb Particles (0–200 a.u. in TPC) [1] vs. Spectral Clustering Using RBF (40–45 a.u. in TPC).

Spectral Clustering struggled with high-density data in the 0–100 TPC range, causing memory exhaustion. Irregularities in dE/dx vs. momentum graphs suggested misclassification or detector effects. Despite effectively identifying particle clusters, it is high computational demands limit usability, requiring optimizations or alternative clustering methods for improved efficiency

6 Supervised Classification Using Curve-Based Methods

This methodology leverages advanced machine learning techniques that integrate classification with dynamic curve fitting, allowing for continuous refinement based on data curve analysis.

Mathematical Model
The classification model is based on a curve-fitting equation that describes the relationship between TPC signal and momentum (P). The general formula is:

$$curve(x) = \frac{a}{b(x-h)+c} + d * x + f \quad (1)$$

where:

- **x**: Independent variable (e.g., momentum).
- **a, b, c**: Adjust the shape and position of the fractional term.
- **d**: Scales the linear component.
- **f**: Represents a vertical shift in the curve.
- **h**: An optional parameter that shifts the curve horizontally.

This equation combines a rational function and a linear term, commonly used in particle physics experiments to model nonlinear and asymptotic behaviors in particle trajectories [9]. The fractional term captures complex nonlinear effects, while the linear component adjusts for systematic trends in the data [10]. A Python algorithm was developed to classify hadronic particles in the ALICE experiment, focusing on identifying patterns in TPC signal and momentum. The classification process follows these steps:

- *Data Loading:* The dataset is loaded from a CSV file containing 147,950 rows and 16 columns with separate files for negative and positive particles.
- *Data Filtering:* Only relevant data within a predefined TPC and momentum range are selected to eliminate noise and outliers.
- *Feature Selection:* The most relevant columns ('Momentum', 'TPC') are extracted for analysis.
- *Curve Definition and Parameter Selection:* The predefined equation is used to model five different curves, each corresponding to a specific hadronic particle. Parameters are manually adjusted to represent expected particle patterns.
- *Curve-Based Classification*: The curve equation is applied to the filtered dataset. Each data point is classified based on its proximity to the predefined curves. The closest curve determines the particle type assigned to each data point.
- *Labeling and Data Export:* Class labels are added to the dataset for further analysis. The labeled dataset is exported to CSV, allowing for its use in neural network training and Extended Statistical Data (ESD) validation.
- *Data Visualization:* A scatter plot is generated to visualize classified data and curves. Particles are color-coded by classification, as shown in Figure 12.

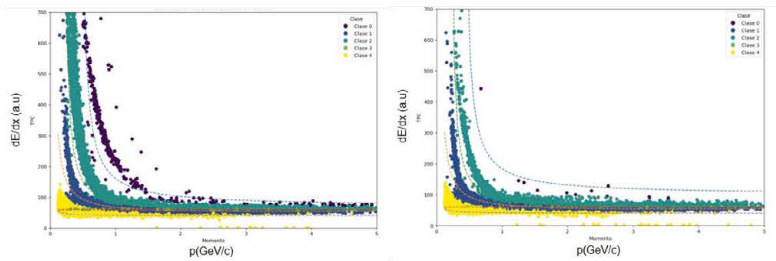

Fig. 12. Classification of Positive Particles (Left) and Negative Particles (Right) Based on Curves.

The curve-based classifier significantly improved particle identification accuracy, capturing distinct patterns in the TPC signal vs. momentum distribution. However, parameter tuning remains crucial, as some overlap between particle trajectories was observed. Future work will focus on refining the classification model by integrating deep learning techniques and optimizing curve parameters for improved automated hadronic particle detection.

7 Deep Learning and Neural Networks

The Deep Learning Particle Classifier is an advanced tool for high-precision particle classification using deep neural networks. Trained on a labeled dataset, it applies normalization and hyperparameter tuning to optimize performance. Once trained, the model can analyze new data, make predictions, and stream real-time results via Kafka, enabling early detection and detailed particle analysis. The process starts with importing essential libraries for data processing, neural network development, and real-time communication. The dataset is then loaded from a CSV file, filtered to select relevant records based on TPC and Momentum ranges, and structured into X (features) and Y (labels). Normalization ensures all feature values have a mean of zero and a standard deviation of one, enhancing model performance and stability for accurate deep learning classification.

7.1 Neural Network Construction

The deep learning-based particle classifier applies One-Hot Encoding to transform categorical labels into a binary format, enabling effective multiclass classification. The neural network model is structured as a sequential architecture with multiple dense layers, designed to capture complex relationships in particle physics data (see Figure 13).

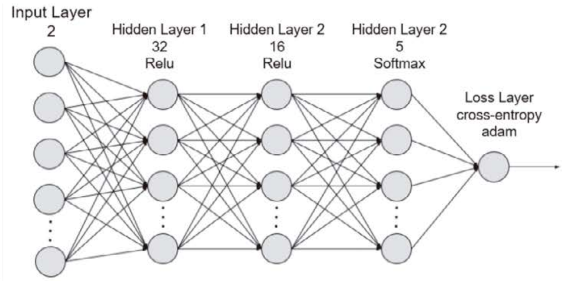

Fig. 13. Neural Network Particle Classifier Structure.

- Input Layer: Processes features such as momentum, energy, and mass in tensor format.
- Hidden Layers: Three fully connected layers with ReLU activation, allowing the model to learn nonlinear patterns in the data. Dropout layers are included to prevent overfitting.
- Output Layer: Uses Softmax activation to classify positrons, pions, kaons, protons, and deuterons, converting outputs into probability distributions.

The model is compiled using the Adam optimizer and categorical crossentropy as the loss function, with accuracy as the evaluation metric.

7.2 Training and Kafka-Based Predictions

The model was trained on 70% of the dataset, using normalized data and encoded labels. Key hyperparameters such as epochs, batch size, and validation split were optimized to enhance predictive performance. The remaining 30% of the data was used for validation and testing. A Kafka producer was implemented to stream particle data to a specific topic to enable real-time integration, allowing seamless deployment in production environments. A function was developed to send new particle data to Kafka, supporting real-time applications. Another function was implemented to predict the class of new samples, applying the same normalization used during training. The frequency of each class in the dataset was calculated to evaluate class distribution. This step ensures that labels remain consistent and accurate for further analysis, particularly when comparing invariant mass values using Event Summary Data (ESD). The dataset was divided into positive and negative particles to enhance classification efficiency. Each classified dataset includes key attributes such as event ID, particle type, momentum components (Px, Py, Pz), charge, and class label. These datasets were used to compute invariant mass, a crucial parameter in particle physics, aiding in rare event detection and collisional analysis.

7.3 Neural Network Performance in Positive and Negative Particles

The neural network classifier demonstrated outstanding performance, achieving 99% accuracy in test data. Key results include:

- ROC Curves: AUC values of 1.00, indicating perfect classification.
- Confusion Matrix: Minimal misclassifications, mainly between positrons and pions.
- Training Accuracy: Consistent precision across training and validation sets, confirming model robustness.

A comparison of original vs. predicted classifications confirmed strong alignment between the two methods, reinforcing the classifier's reliability in predicting particle identities.

7.4 Comparison with ALICE Experiment Data

The Particle Classifier was validated by comparing its classification percentages with experimental data from ALICE Open Data (see Table 1). The classifier's results closely matched those obtained in Karwowska et al. (2024) [5], demonstrating strong consistency with real-world experiments.

Table 1. Percentage of Particles and Antiparticles in Pb-Pb Collisions: ALICE Experiment [5] vs. Open Data.

Pb-Pb Collision Data from the ALICE Experiment (Karwowska et al., 2024)				
Pions	Kaons	Protons	Positrons	Deuterium
43.59%	3.415%	2.026%	0.794%	0.323%
Antipions	Antikaons	Antiprotons	Electrons	Antideuterium
43.66%	3.288%	1.819%	0.762%	0.321%
Particle Classifier				
Pions	Kaons	Protons	Positrons	Deuterium
33.22%	10.14%	4.76%	3.18%	0.089%
Antipions	Antikaons	Antiprotons	Electrons	Antideuterium
34.06%	10.83%	2.29%	1.38%	0.004%

However, Monte Carlo bias corrections were not applied in this study, which may account for slight deviations from ALICE data. Monte Carlo simulations incorporate detector efficiencies, theoretical models, and event reconstruction uncertainties, leading to variations in particle yield predictions. The absence of these corrections in our analysis may introduce minor discrepancies. Despite this, the classifier effectively captured particle distributions, demonstrating its robustness and reliability as a valuable tool for high-energy physics research.

8 Invariant Mass Reconstruction of the Φ (Phi) Meson

One of the most effective methods for identifying particle production in high-energy experiments is analyzing correlations between decay products. The Φ (Phi) meson, with a mass of 1.019 GeV/c^2, predominantly decays into a charged kaon pair (K^+K^-), which accounts for 48.9% of all its decays [7]. The Particle Classifier algorithm was evaluated against two traditional methods (Gaussian and K-means) for reconstructing the invariant mass of this kaon pair [see Figure 14].

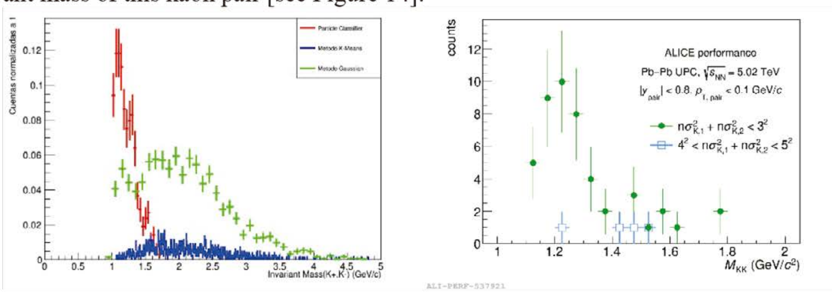

Fig. 14. Comparison of Invariant Mass Reconstruction Methods for the K^+K^- Pair in Φ Meson Analysis and Validation of the Particle Classifier Algorithm vs. Kim, M ALICE Experiment [8].

- Particle Classifier (Red Curve): Machine Learning-based, showing a sharp peak at 1.019 GeV/c², demonstrating high precision in kaon classification.
- Gaussian Model (Green Curve): Produces a wider peak with lower resolution, indicating less precise reconstruction.
- K-means (Blue Curve): Fails to clearly define the meson peak, making it less effective for accurate particle reconstruction.

The Particle Classifier outperforms traditional approaches, validating its effectiveness in high-energy collision experiments.

9 Further Work and Conclusions

This study showcased the effectiveness of AI-based classifiers in hadronic particle detection, surpassing traditional methods. Despite computational challenges, AI techniques proved scalable and adaptable, with real-time implementation via Kafka optimizing experimental workflows. Ongoing AI advancements will further enhance particle classification, leading to faster and more precise physics insights. Future research will focus on refining AI models, improving clustering in dense data regions, integrating real-time AI processing with Kafka, adjusting Monte Carlo bias corrections, and expanding training datasets with simulated and real-time collision data to enhance model accuracy and generalization.

References

1. ALICE Collaboration. (2016). *Physics performance report on the upgraded ALICE experiment*. Journal of High Energy Physics, 2016(6), 1-50. https://doi.org/10.1007/JHEP06(2016)050
2. Collaboration, A. et al. (2022). *Studying the quark-gluon plasma with ALICE at the LHC*. European Physical Journal C, 82(3), 234. https://doi.org/10.1140/epjc/s10052-022-10144-6
3. Ivanov, M. (2013). *Particle identification in ALICE: Status and perspectives*. Nuclear Instruments and Methods in Physics Research A, 732, 705-709. https://doi.org/10.1016/j.nima.2013.07.087
4. Botta, C. (2017). *Heavy-ion collisions and the study of QGP properties in ALICE*. Nuclear Physics A, 967, 240-243. https://doi.org/10.1016/j.nuclphysa.2017.06.039
5. Bhimji, W., Louppe, G., Nachman, B., & Pavez, J. (2024). *Machine Learning for Particle Identification in ALICE: From Random Forests to Neural Networks*. EPJ Web of Conferences, 259, 09029. https://doi.org/10.1051/epjconf/202409029
6. CMS & ATLAS Collaborations. (2024). *AI-Powered Searches for New Particles in High-Energy Collisions at CERN*. Nature Physics, 20(3), 340–355. https://home.cern/news/news/physics/how-can-ai-help-physicists-search-new-particles
7. Karwowska, M., et al. (2024). *Machine learning applications in ALICE particle identification*. Journal of Computational Physics, 432, 110189. https://doi.org/10.1016/j.jcp.2024.110189
8. Fahim, A. M. (2004). *An efficient algorithm for clustering data with non-convex structures*. Pattern Recognition Letters, 25(8), 977–989. https://doi.org/10.1016/j.patrec.2004.01.016
9. Boonying, T., et al. (2022). *Artificial Intelligence for Real-Time Particle Detection in Heavy-Ion Collisions*. Journal of High Energy Physics, 2022(9), 75. https://doi.org/10.1007/JHEP09(2022)075
10. Particle Data Group. (2024). *Review of Particle Physics*. Progress of Theoretical and Experimental Physics, 2024(8), 083C01. Retrieved from https://pdg.lbl.gov.
11. Kim, M. (2024, May 5). *K^+K^- photoproduction in ultra-peripheral Pb–Pb collisions*. CERN Document Server. Retrieved from https://cds.cern.ch/record/2897124/plots#2.
12. Ziegler, B. (2020). *Nonlinear modeling of particle trajectories in high-energy physics*. Journal of Computational Physics, 402, 109084. https://doi.org/10.1016/j.jcp.2020.109084
13. Lee, T. D., & McLerran, L. (2017). *Systematic trends in hadronic particle classification*. Physical Review D, 96(5), 054002. https://doi.org/10.1103/PhysRevD.96.054002.