Machine Learning Techniques for Particle Classification in Microphysics

Zeyu Yang[1], Deshui He[2,*]

{hackonetic@gmail.com[1], OldWater815666@outlook.com[2]}

School of Artificial Intelligence, Xidian University, Xi'an, 710126, China[1]
Dundee International Institute, Central South University, Changsha, 410083, China[2]
*corresponding author

Abstract. This study proposes various machine learning methods for classifying physical particles including protons, pions, kaons, and D mesons in microphysics. The research group evaluated the performance of decision trees, neural networks, traditional methods, *et cetera*. showing that Decision Trees are better than other methods in terms of the mean square error and the accuracy of the final classification. This indicates that the decision tree may provide certain data features and is an outstanding and feasible approach in this field. The last achieved accuracy was 65 % for the global range and 73% for the local range. Thus, it can be a good way to improve particle classification by combining traditional and machine learning methods.

Keywords: Decision trees, neural networks. physics, particle classification, machine learning, regression.

1 Introduction

Particle identification is significant in microphysics. However, classification accuracy is not promising due to the limitation of human-eye identification. Earlier, classification was done using regression techniques that use previously defined features, but this method lacks much precision, particularly when influenced by some outliers. As a result, there are disparities in the accuracy of the classifications; so, more precise classification techniques must be used. As science and technology are developing forward, machine learning techniques have become the most effective approach to classifying objects. While conventional regression models cannot identify non-linear correlations in the data, the machine learning models fixed this problem. Therefore, the method of machine learning can bring breakthroughs in the field of particle classification in terms of procession and efficiency. These models can be generally divided into two parts: dE/dxestimation and particle identification. This study aims to determine whether the machine learning approaches can perform better than conventional methods in a classifying task, then find out the best performing algorithm in classifying particles accurately from a given dataset to aid in the improvement of more accurate methods in microphysics.

1.1 Machine Learning-Based dE/dx Estimation

Predicting dE/dx is one of the paramount challenges. Some of the conventional techniques of calculating dE/dx including calculating the truncated mean can estimate the true value approximately, but it couldn't completely rule out outliers. In the contemporary world, more

CONF-MLA 2024, November 21, Adana, Turkey

DOI 10.4108/eai.21-11-2024.2354579

sophisticated approaches in machine learning can train the models to estimate dE/dx better [1]. For instance, machine learning has been applied to enhance the analyses of the energy deposition of particles in detector, thus refine the determination of dE/dx [2]. In the present study, Lopes has proposed a new deep neural network compression technique that would enhance the execution of the network in FPGAs for improved dE/dxestimation and particle classification. This technique reduces the number of bits needed to represent network parameters, which in turn reduces computation time although the performance is not greatly affected [3]. Shao et al. developed an MLP model that incorporates quantum computing and classical ML algorithms, which in addition to increasing the efficiency of dE/dx estimate enhances classification efficiency [4]. Moreover, the ATLAS organization has tried to use the XGBoost algorithm to better separate the signal and background data, which enhanced the sensitivity of analysis and dE/dx estimation [5]. Hence, as machine learning is applied in particle physics, there has been the development of several fundamental approaches. Compared with the conventional approaches, the machine learning methods give substantial enhancements in the dE/dx estimation, thus on the one hand, enhancing the efficiency and precision of the tools for particle physics study, and on the other hand, cutting down the computation time.

1.2 Particle Classification Using Machine Learning

Traditional methods for particle classification rely on human-eye observation and empirical rules, which perform poorly with complex collision events. Recent studies indicate that deep learning techniques like convolutional neural networks (CNNs), can effectively address these challenges. For example, Tripathi et al. developed a 10-layer CNN model that achieved 65% accuracy with a loss of 0.6 [6]. Komiske et al. introduced the PUMML algorithm, utilizing CNNs to correct pileup effects and reconstruct particle distributions, while also leveraging GANs to generate data, enhancing classification performance and reducing reliance on experimental data [7].

Understanding the decision-making process of machine learning models is crucial in particle physics. Lundberg et al. (2020) improved model transparency using AI techniques like SHAP values, which clarify the relationship between features and classification results [8]. Grojean et al. emphasized the benefits of Interpretable Machine Learning in enhancing model understanding and particle classification accuracy [9]. In conclusion, machine learning has significantly advanced particle classification, replacing conventional methods and yielding impressive results, while also proposing various methods to improve model accuracy and efficiency [10, 11, 12].

2 Model Review

The main goal of the study is to use data to complement regression and classification tasks, divided into two parts. Step One focuses on parameter extraction, applying various methods to transform 50-dimensional data into a crucial one-dimensional parameter that impacts the experiment's outcome. Mean Square Error (MSE) is used to evaluate method accuracy. Step Two aims at classification, combining the generated parameter with existing one-dimensional data to classify using different approaches and compare their accuracy to determine the most effective method. In the following procedures, divided steps on this topic mainly take advantage of three methods: the traditional method, Decision Tree model, and the Neural Network model.

The outcome of the corresponding indicator will be exploited to indicate which is the optimal strategy for these specific tasks.

2.1 Model Review for Step One

In terms of step one, the traditional method employed is called the truncated mean, also known as the trimmed mean, to predict the essential parameter $\frac{dE}{dx}$ signal, which formula is given below:

$$\frac{dE}{dx} = \frac{1}{pr \times N} \sum_{i=1}^{pr \times N} x_i \tag{1}$$

where pr denotes an arbitrary percentage ranging from 0 to 1, N refers to the sample size of the given data, and x_i is sorted in ascending order within each set of the 50-dimensional data, which means the truncated mean only counts the mean of data from the smallest to a specific percentage, excluding the data at proportion pr of the largest values. This way, the value is not affected by outlying points that might distort the mean and give a better estimate. In the experiment, comparing the least MSE value at which the optimal percentage is achieved for predicting the signal from the given value of pr.

The decision tree model used is the Classification and Regression Trees (CART), a machine learning algorithm for classification and regression tasks introduced in 1984. CART constructs a binary tree structure from input features, starting with the entire dataset at the root node. The algorithm evaluates possible splits for all features, selecting the one that maximizes impurity reduction. This process continues until a stopping criterion is met. In Step One, data is sorted and split based on a specific proportion in decreasing order.

The selected portion is then applied to the constructed CART decision tree to predict the dE/dx signal.

The Neural Network model comprises six layers, each utilizing the ReLU activation function, which is shown in Fig 1.

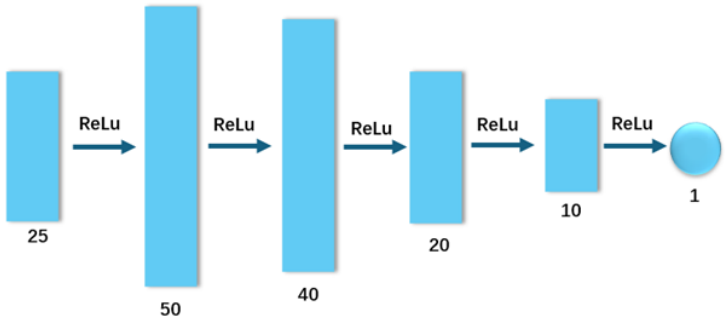

Fig.1 Framework of neural network to predict $\frac{dE}{dx}$

The dataset is first trimmed at the optimal percentage determined by the traditional method, and the neural network transforms this truncated data into a 50-dimensional representation. The network then reduces the dimensionality from 50 to 40, 20, and 10, ultimately outputting a one-dimensional vector.

This final output represents the dE/dx signal we aim to predict.

The dataset is first trimmed at the optimal percentage calculated by the traditional method. Subsequently, the trimmed data is used to train the neural network for the dE/dx signal. This approach ensures that each method—traditional truncated mean and neural network—leverages the most effective data preprocessing techniques to enhance predictive accuracy.

2.2 Model Review for Step Two

In Step Two, the traditional method employed is the cut-based approach. It first estimates the widths and the trajectories from the given data. Then, for an unknown particle, use its momentum p to locate the possible area its true dE/dx may fall. it will be classified to the corresponding category if the predicted dE/dx falls on a specific area. CART decision tree can also be deployed in this step, it will be utilized with another neural network as shown in Fig 2.

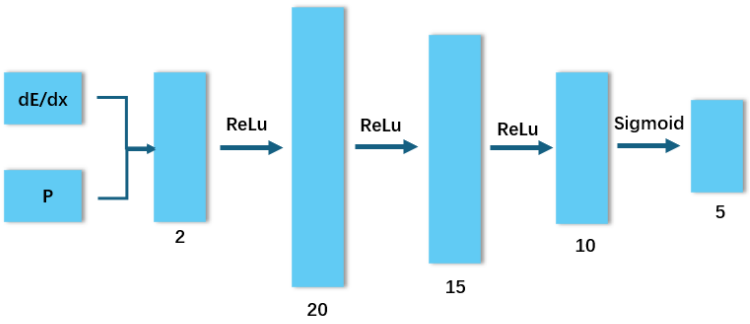

Fig. 2. Framework of neural network to classify

Predicted dE/dx and momentum p is the input data, and the activation function of the hidden layer is ReLU. However, we choose the Sigmoid function to do the normalization of output in the output layer. The unknown particle will be classified to the class with a max value.

3 Experiment

The dataset consists of 100,000 particles, each characterized by 55 values. The first value in each sample is consistently 50, indicating that the subsequent 50 values are features used to predict the $\frac{dE}{dx}$ signal. The second value corresponds to the momentum p, the third to the true signal, and the fourth to the mass, while the fifth value is irrelevant to the project and can be disregarded.

Given that the physical particles are categorized into five distinct types based on their mass, the initial objective of the project is to accurately predict the mass of each particle using the available data, thereby enabling precise classification. Since the exact momentum is known, where $= m \times v$ (with m representing mass and v representing velocity), it is essential to determine the velocity. Velocity has a complex relationship with the $\mathrm{dE/dx}$ signal, a parameter representing the rate of energy loss per unit distance travelled by a charged particle through a medium, as described by the Bethe-Bloch formula.

$$\frac{dE}{dx} = -\frac{4\pi}{m_e c^2} \cdot \frac{z^2 e^4}{\beta^2} \cdot \frac{Z}{A} \cdot \rho \cdot \left[\ln\left(\frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} \right) - \beta^2 \right] \tag{2}$$

The comprehension of the Bethe-Bloch formula is not a concern in this context. In Step One, the primary task is to identify a traditional method or a machine learning model capable of accurately computing the $\mathrm{dE/dx}$ signal from the 50 features associated with each particle, ensuring minimal loss. In Step Two, the objective shifts to developing a model that can predict the velocity using the generated $\frac{dE}{dx}$ signal, thereby replacing the need for the formula. By combining the predicted velocity with the known momentum, the mass can be calculated, enabling accurate classification of the particles.

3.1 Experiment for Step One

By varying the percentage from 0 to 1 and collecting the corresponding Mean Squared Error (MSE), figure 3 is generated, with the x-axis representing the selected proportion and the y-axis representing the loss. The plot reveals that when the percentage is approximately 0.5, the MSE reaches its minimum on both the training and test sets, with values of 0.007554 and 0.007538, respectively.

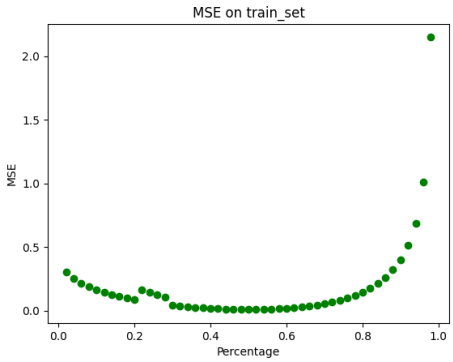

Fig. 3. Scatter plot of percentage vs MSE in truncated mean

For the machine learning regression task, we used a 50% truncation percentage, retaining the top samples and dividing the data into training (30%) and testing sets. To find the best model, we varied the decision tree's maximum depth from 1 to 30, recording the Mean Squared Error (MSE). The optimal depth is 8, yielding the lowest MSE of about 0.004113, while deeper trees showed overfitting. For the neural network, we applied similar preprocessing, set the learning

rate to 0.0001, and trained for 10,000 epochs. The minimum MSE of approximately 0.004519 was achieved at epoch 9998.

To draw a conclusion regarding model performance, we compared the smallest mean squared error (MSE) on the test set between the CART decision tree regressor and the neural network regressor. The results shown in figures 4 and 5 suggest that the decision tree model outperforms the neural network model when predicting a specific parameter using a dataset truncated by half.

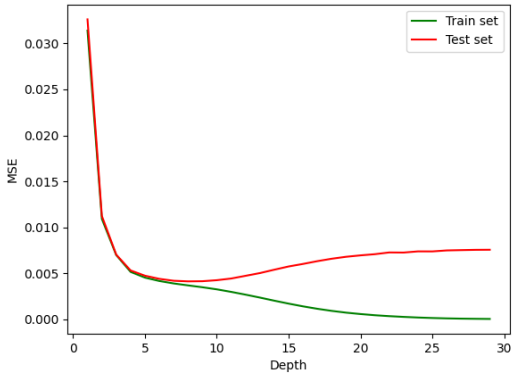

Fig. 4. Plot of depth vs MSE in CART decision tree

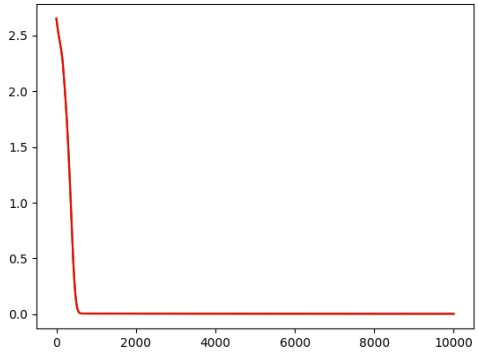

Fig. 5. Plot of epochs vs MSE in neural network

To further assess whether traditional methods surpass computer algorithms, we plotted a histogram that displays the distribution of the ratio $\frac{predict\,\frac{dE}{dx}}{true\,\frac{dE}{dx}}$ for both the truncated mean approach and the decision tree model. This visual comparison highlights the differences between the two methods. In the Gaussian-like distributions, performance is influenced by two factors: the mean's proximity to one, indicating better alignment with true values, and the distribution's width, where a narrower width suggests a smaller standard deviation and greater concentration around the mean.

The overall histogram of the ratio on the test set, shown in Fig 6, reveals that the signal from the decision tree regressor exhibits a smaller width and a mean closer to one. It is obvious that the signal generated by the decision tree regressor has a smaller width and a closer mean to one.

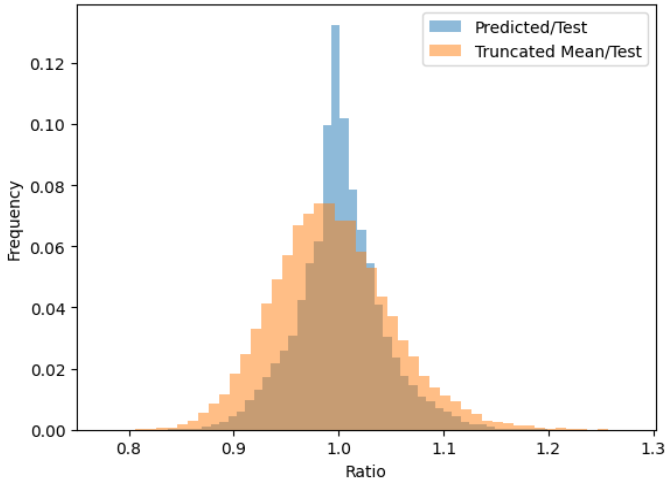

Fig. 6. The overall histogram of the ratio

To conduct a more detailed analysis, we divided the momentum p into five intervals: [0, 2], (2, 4], (4, 6], (6, 8], and (8, 10]. For each momentum interval, we generated histograms for each particle type. Each row represents a different particle type, while columns correspond to increasing momentum intervals. In the histograms, the orange distribution shows the truncated mean ratio (truncated at 50%), and the blue distribution represents the decision tree regressor ratio.

By comparing the Gaussian distributions within each figure, we reaffirm the conclusion drawn from the overall histogram: the decision tree regressor consistently predicts dE/dx more accurately than the truncated mean method.

3.2 Experiment for Step Two

Since there are different ways to predict dE/dx, it is proper to do the classification by using dE/dx predicted by different ways, these are the procedures:

Truncated mean + Cut-based approach (A)

1. Decision tree dE/dx + Cut-based approach (B)
2. Truncated mean + Decision tree classification (C)
3. Decision tree dE/dx + Decision tree classification (D)
4. Truncated mean + Neural network classification (E)
5. Decision tree dE/dx + Neural network classification (F)

Since the machine learning algorithm may be used twice in a single experiment, the data on the testing set will have to be split again, so there are about 9000 objects to measure the performance of the experiment eventually.

The result of experiment A is shown in Fig 11:

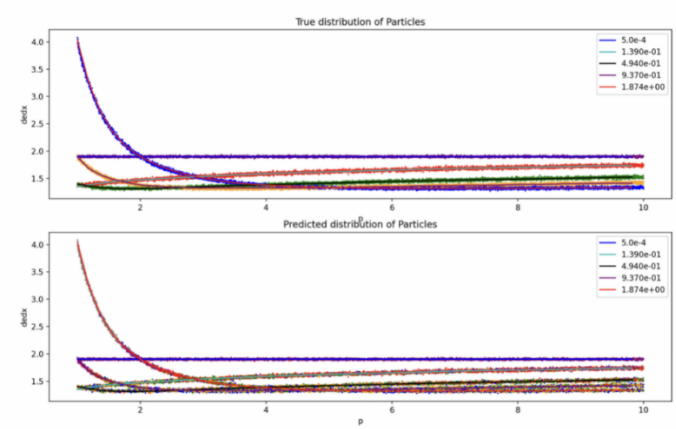

Fig. 11. The result of Experiment A

By changing the width, the best accuracy will be reached when the width is 0.3, which is 0.59625.

The result of experiment B is shown in Fig 12:

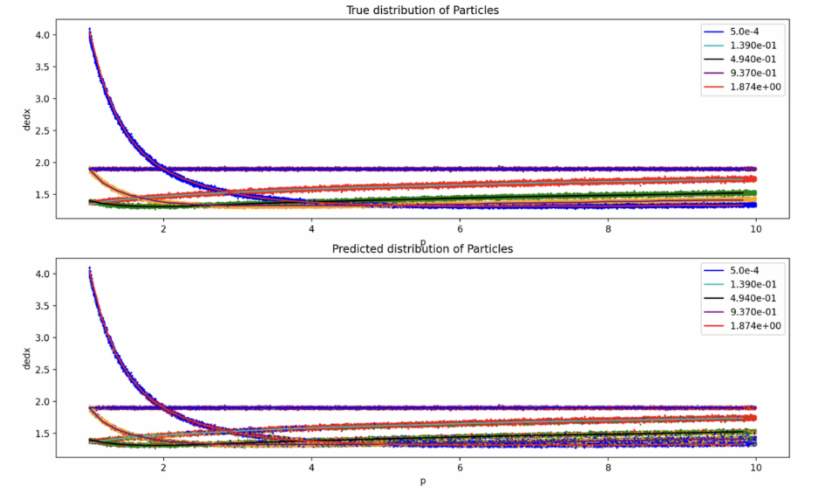

Fig. 12. The result of Experiment B

By changing the width, the best accuracy will be reached when the width is 0.28, which is 047514.

The result of experiment C is shown in Fig 13:

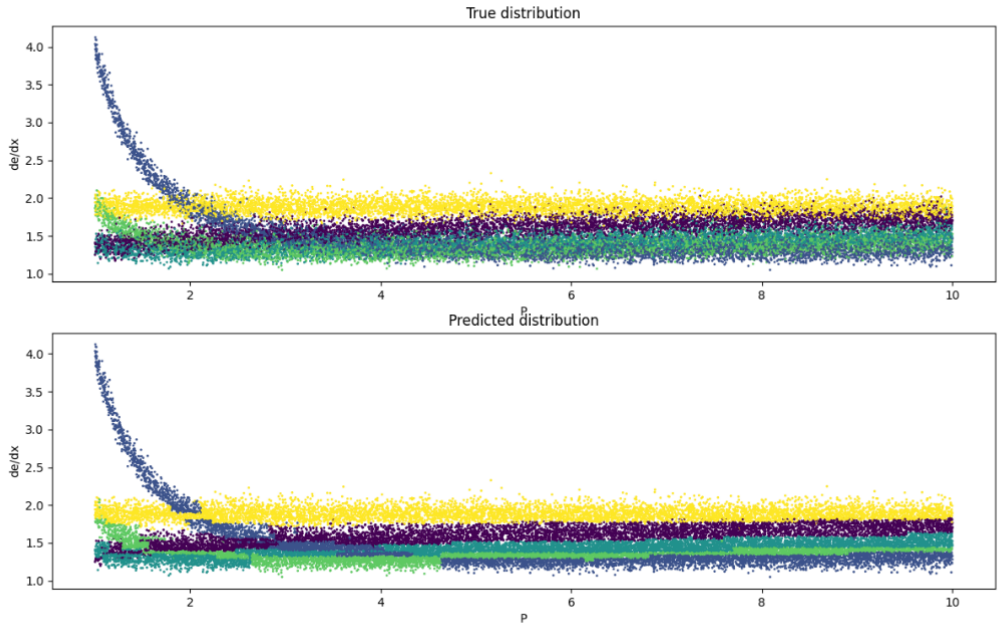

Fig. 13. The result of Experiment C

By changing the max depth, the model performs best when the max depth is 9, where the accuracy is 0.63153.

The result of experiment D is shown in Fig 14:

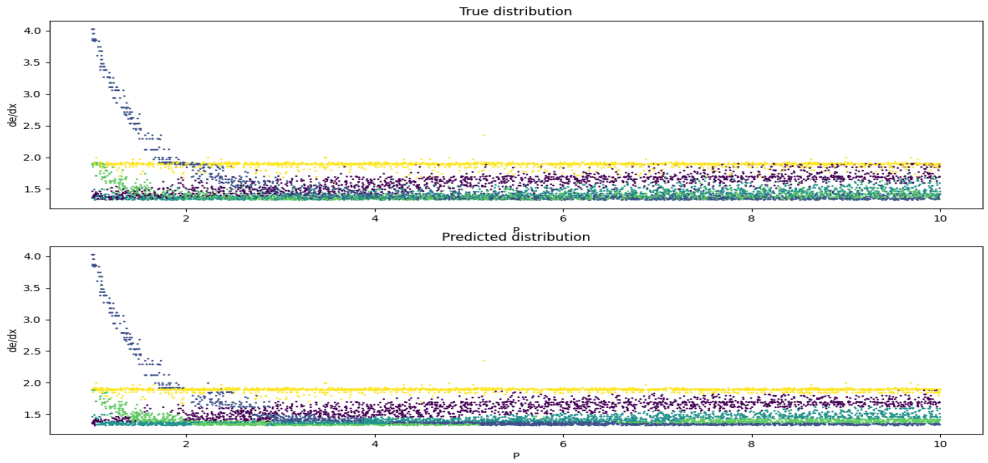

Fig. 14. The result of Experiment D

By changing the max depth, the model performs best when max depth is 8, where the accuracy is 0.654.

The result of experiment E is shown in Fig 15:

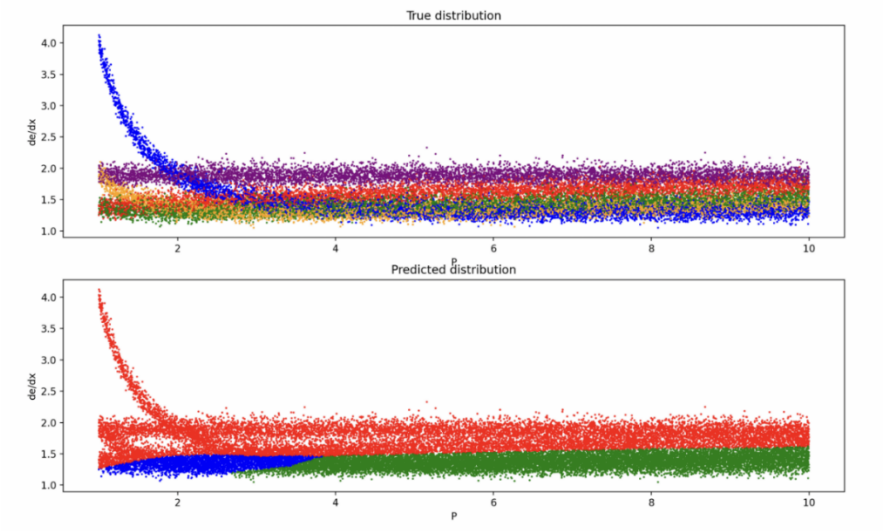

Fig. 15. The result of Experiment E

Set epochs = 20000, learning_rate = 0.001, the eventual accuracy is 0.31.

The result of experiment F is shown in Fig 16:

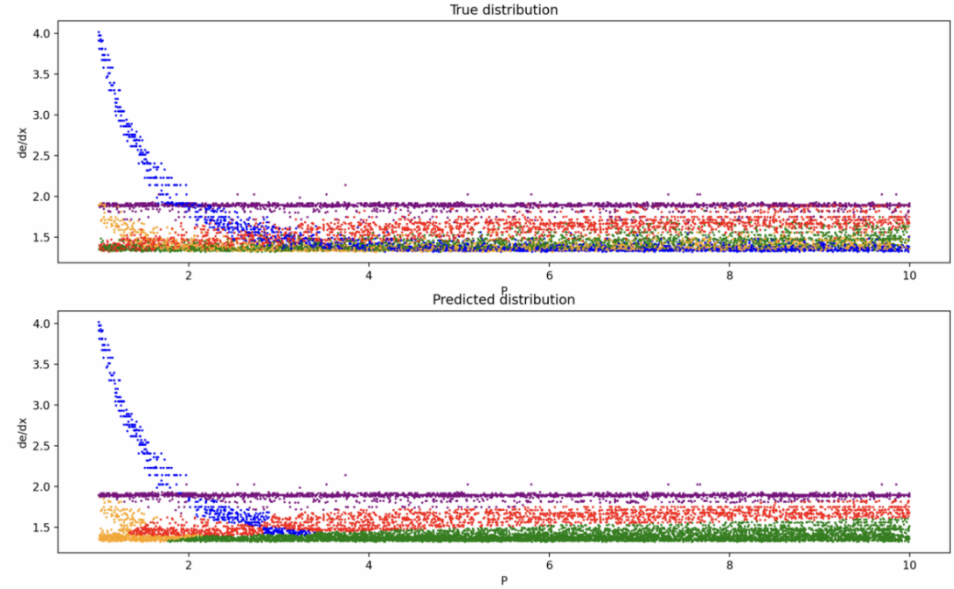

Fig. 16. The result of Experiment F

Set epochs = 20000, learning_rate = 0.001, the eventual accuracy is 0.55.

All the results are depicted in table 1:

Table 1. Accuracy of each method in Step 2

Methods	Accuracy
Truncated mean + cut-based approach	0.59
Decision tree dE/dx + cut-based approach	0.48
Truncated mean + decision tree classification	0.63
Decision tree dE/dx + decision tree classification	0.65
Truncated mean + neural network classification	0.31
Decision tree dE/dx + neural network classification	0.55

Although decision trees have a lower mean squared error than truncated means, accuracy does not improve with a cut-based classification approach. In contrast, using machine learning algorithms generally yields higher accuracy with lower MSE. Decision trees outperform neural networks due to their ability to automatically adapt to data structures without needing to set hyperparameters. To enhance accuracy, it's recommended to exclude electrons and classify particles with momentum between 0 and 2, achieving a best accuracy of 0.7272 with decision trees.

4 Conclusion

In this study, the research group used machine learning techniques to improve estimation and particle classification in particle physics. Traditional methods often struggle with complex collision events and outliers. By leveraging algorithms like MLP, CNN, and decision trees, the group found that decision trees performed best. Results indicated that both classification methods and estimation significantly impact particle identification accuracy. However, when using the decision tree, improvements in accuracy from optimized estimation were minimal. Future studies should enhance algorithms for noise reduction and integrate conventional methods with data-mining approaches to improve model performance and discovery in particle physics. Maintaining the robustness of these models is crucial for broader applications in particle identification.

Table 2. Accuracy of all combined methods in Step 1 and Step 2

method	accuracy	
	truncated mean	predicted $\frac{dE}{dx}$
decision tree	0.6315	0.654
neural networks	0.31	0.55
cut-based	0.596	0.485

The study demonstrated how refined machine learning algorithms enhance particle estimation and classification. These techniques improve identification efficiency and can be applied to various areas in particle physics. For instance, machine learning can analyze specific particle occurrence rates, upgrade event reconstruction processes, or search for new particles in collisions by identifying patterns that traditional methods might miss.

References

[1] Schwartz, M. D. (2021). Modern Machine Learning and Particle Physics. Harvard Data Science Review, 3(2). https://doi.org/10.1162/99608f92.beeb1183

[2] Tripathi, J., Bhatnagar, V. (2022). Convolutional Neural Networks in Particle Classification. In: Dua, M., Jain, A.K., Yadav, A., Kumar, N., Siarry, P. (eds) Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-5747-4_65

[3] Lopes, A. (2021) *Speeding up machine learning for particle physics.* Available at: Speeding up machine learning for particle physics | CERN (home. cern) (accessed: 2024/8/6)

[4] Shao, C.: A quantum model for multilayer perceptron (2018). https://arxiv.org/pdf/1808.10561.pdf

[5] ATLAS Collab., *ATLAS-CONF-2019-028*, CERN, July 23, 2019.

[6] Tripathi, J., Bhatnagar, V. (2022). Convolutional Neural Networks in Particle Classification. In: Dua, M., Jain, A.K., Yadav, A., Kumar, N., Siarry, P. (eds) Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-5747-4_65

[7] Komiske, P. T., Metodiev, E. M., Nachman, B., & Schwartz, M. D. (2018). Learning to classify from impure samples with high-dimensional data. *Physical Review D, 98*(1), Article 011502. https://doi.org/10.1103/PhysRevD.98.011502

[8] Lundberg, S. M. et al. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* 2, 56–67 (2020).

[9] Grojean, C., Paul, A. & Qian, Z. Resurrecting bb¯h with kinematic shapes. *J. High Energy Phys.* 4, 139 (2021).

[10] Grojean, C., Paul, A., Qian, Z. *et al.* Lessons on interpretable machine learning from particle physics. *Nat Rev Phys* 4, 284–286 (2022). https://doi.org/10.1038/s42254-022-00456-0

[11] Purohit, M.V. (2024). Machine Learning in Particle Physics. In: Sachdeva, S., Watanobe, Y. (eds) Big Data Analytics in Astronomy, Science, and Engineering. BDA 2023. Lecture Notes in Computer Science, vol 14516. Springer, Cham. https://doi.org/10.1007/978-3-031-58502-9_9

[12] Butter, A., et al. "Machine Learning in Particle Physics." SpringerLink, 2022.