Dalitz Decay Identification with Machine Learning and Deep Learning Techniques

Yuhao Peng

December 15, 2024

Abstract

This project applies machine learning (ML) and deep learning (DL) techniques to develop a classifier that can distinguish Dalitz decays from background events, which is essential for the dark photon search with Belle II experiment. A non-parametric model, the Random Forest (RF) classifier, and a parametric model, the Artificial Neural Network (ANN), were trained on a balanced dataset of 33,720 entries. By selecting features based on the importance scores from the RF, the performance of the models improved. The ANN model with the selected features, achieved a classification accuracy of 96%. This work not only supports dark matter research but also provides a solid framework for handling complex, high-dimensional data in other similar classification tasks in particle physics.

1 Introduction

The Standard Model of particle physics has been extremely successful in explaining many of the phenomena we observe in the universe. However, it does not provide answers to all the questions, particularly regarding the elusive nature of dark matter. The dark matter is believed to make up a large portion of the universe's mass, but it has not been directly detected. Despite its strong gravitational influence on visible matter, it remains invisible to current observational methods. Understanding dark matter is one of the most important challenges in modern physics, highlighting the need for new theories that go beyond the Standard Model. One of the most promising candidates for dark matter is the dark photon as yet undiscovered particle that could explain interactions between ordinary matter and dark matter.

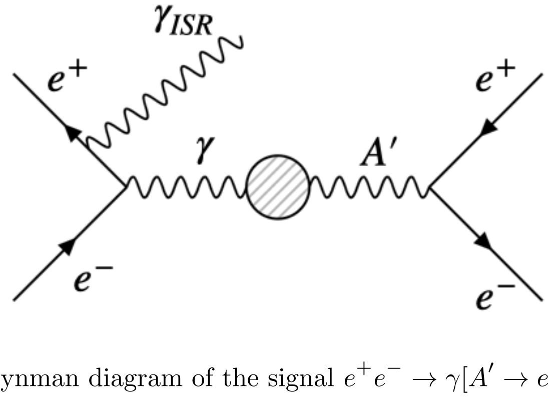

Figure 1: Feynman diagram of the signal $e^+e^- \to \gamma[A' \to e^+e^-]$ decays

ATOMKI, a nuclear research institute in Hungary, discovered an anomaly that may indicate the presence of a 17 MeV dark photon [1]. The Belle II experiment[2], located at the SuperKEKB e^+e^- collider in Japan, is a highly sensitive particle detector designed to measure rare particle decays. Our research group has proposed a project to detect dark photon decays at an energy level of 17 MeV using the Belle II experiment. This will be achieved by identifying a single photon along with an e^+e^- pair event, as illustrated in Figure 1.

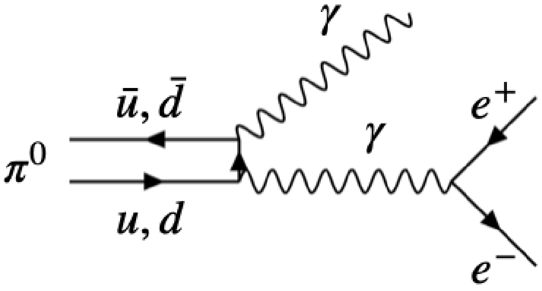

Figure 2: Feynman diagram of the signal $\pi^0 \to \gamma e^+e^-$ decays

Analysis techniques developed to identify the dark photon decays event. To validate and refine our detection methods, we use Dalitz decay[3] events as a benchmark. Dalitz decays are well-understood and produce decay patterns similar to those we expect from dark photons. In a Dalitz decay, as shown in Figure 2, neutral pion (π^0) decays into a photon and an electron-positron pair, and these events can be used to validate our analysis techniques. However, distinguishing Dalitz decays from other background events such as photon conversion, is not a easy task with existing analysis methods.To address this issue, machine learning (ML) and deep learning (DL) techniques are applied to train a classifier on the data generated by Monte Carlo (MC) simulation and then apply it to the experiment data.

2 Theory & Background

2.1 Experiment Events & MC Data

The specific events to be studied are shown in Figure 3, where an electron-positron collision generates a tau pair. These tau pairs decay almost immediately after their creation due to their very short lifetime. τ^+ events are labeled as tags, which are irrelevant to this specific classification task, while τ^- events are labeled as signals because they decay into a neutral pion (π^0) and a charged pion (π^-). The π^0 has a chance of undergoing Dalitz decay at the end of the decay chain with a branching fraction of 1.7%. It is evident that one of the most important features to examine is the vertex of the e^+e^- pair. This vertex is denoted as Ap (A'), as shown in Figure 4, which is an assumption made for the data analysis.

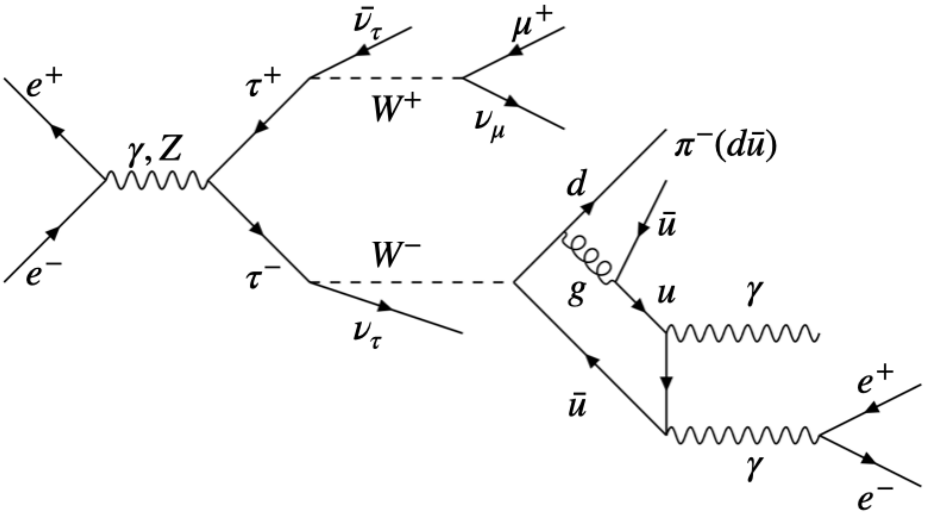

Figure 3: Decay Chain with π^0 made up of $u\bar{u}$

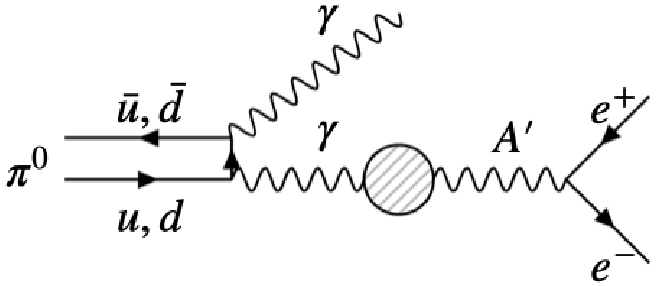

Figure 4: Dalitz Decay with the presence of A'

Before running the actual experiment, Monte Carlo (MC) data is used for validation. The MC simulates the production of e^+e^- collisions along with the corresponding decay chain and models the behavior of the detector. With MC data, the ground truth of each particle's creation is known based on the simulation settings. However, experimental data requires reconstruction to identify and analyze these events.

The genMotherPDG is provided only in the MC data, which gives information about the mother particle of a decay. It labels particles with a unique ID assigned by the Particle Data Group (PDG). For the neutral pion (π^0), the ID is 111. Using this ID, Dalitz decays can be labeled by examining the genMotherPDG of the e^+e^- pair, which can be directly detected by the PXD layer in the detector. If both the genMotherPDG values are 111, then the decay is classified as Dalitz. Otherwise, it is Non-Dalitz.

2.2 Signal & Event Background

Figure 5 shows events of Dalitz decay and photon conversion, which contribute significantly to the background events. Dalitz decay is more likely to occur in the region inside the beam pipe due to the short lifetime of the particles involved. In contrast, photons do not decay in the vacuum space, but instead undergo photon conversion, which decays into an e^+e^- pair. This conversion is more likely to happen when the photon interacts with the material, such as the PXD layer. When the photon hits the PXD layer, it can produce an e^+e^- pair, which is then detected and contributes to the background noise.

Figure 6 shows the spatial displacement of the reconstructed Ap vertex. From this figure, it can be immediately observed that most Dalitz decay events occur inside the beam pipe, with a radius of approximately 1 cm, while most Non-Dalitz events occur at radii of 1.4 cm and 2.2 cm, which correspond to the locations of the PXD layers. This spatial distribution helps distinguish between Dalitz and Non-Dalitz events based on their locations in the detector.

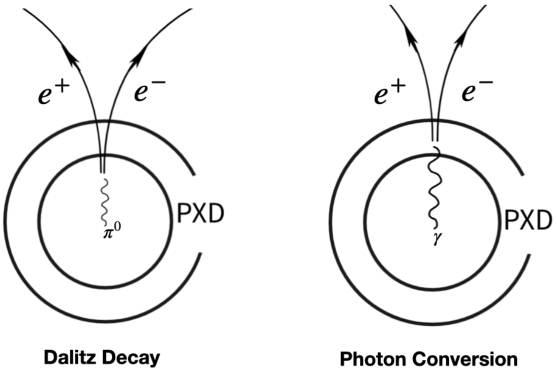

Figure 5: Event location of Photon Conversion vs the Dalitz Decay in Detector

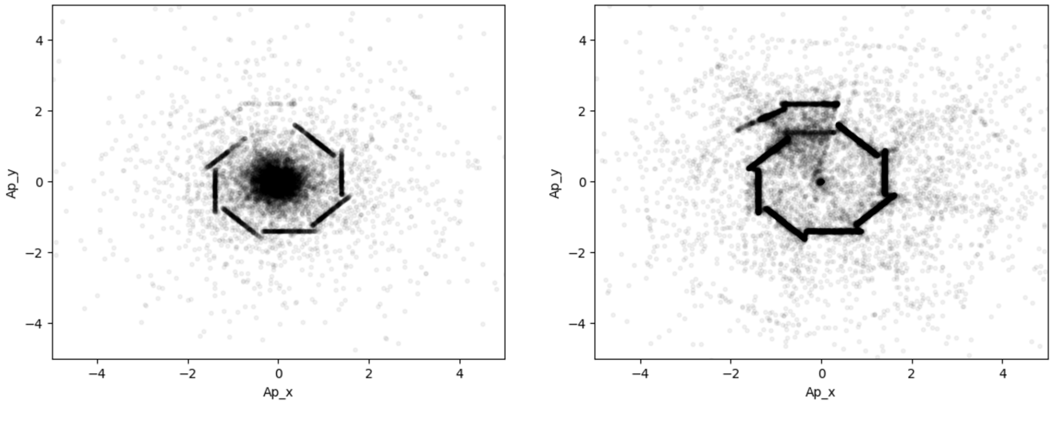

Figure 6: Comparison of the Spatial Displacement of Reconstructed Ap vertex in MC Simulation: Left: Dalitz, Right: Non-Dalitz

3 Implementation

3.1 Data Processing

The raw dataset used in this study is generated by Monte Carlo (MC) simulations. It contains 596 features and 39,248 entries, but a significant portion of these features are redundant. Among them, the first 24 features represent general experimental information, such as collision energy, productions, and other settings. These features are dropped in the first step of data cleaning, as they are irrelevant to the analysis.

Prior knowledge of the data indicates the presence of NaN and infinity values. The infinities are first replaced with NaNs. Any columns completely filled with NaNs are then removed. Since the model is designed to predict experimental data, all columns containing MC-specific information are excluded. The primary objective of this study is to detect Dalitz signal events by analyzing electron-positron tracks. Features irrelevant to this task, such as those unrelated to the direct decay chains (e.g., π^+ and τ^+ decay chains, referred to as tags), along with features such as charge and particle ID, are removed from the data. Additionally, columns with zero variance are removed, as they offer no valuable information for signal classification.

Columns with partial NaN entries are carefully studied to retain as many rows as possible while preserving potentially important features. The most significant features are those associated with the Ap-type vertex of Dalitz decays. For instance, the feature Ap_SigM has the smallest number of non-null entries (37,438) among Ap events. Columns with fewer non-null entries than this threshold are discarded. Features with non-null entries between 37,438 and 39,248 are further examined for their relevance. These features are dropped with NaNs and then classified into Dalitz and Non-Dalitz categories based on the MotherPDG information provided by the MC. However, the distributions of these features show minimal distinction between the two categories, as illustrated in Figure 7. Additionally, prior knowledge suggests that features related to spatial displacements of Ap vertex plays a more significant role than mass in this classification task. A strong evidence can be seen from Figure 8, the Dalitz and Non-Dalitz distributions having distinct and well-separated peaks. Furthermore, the distribution of NaNs across different features can cause significant data reduction. Consequently, all features containing any NaN values are ultimately dropped.

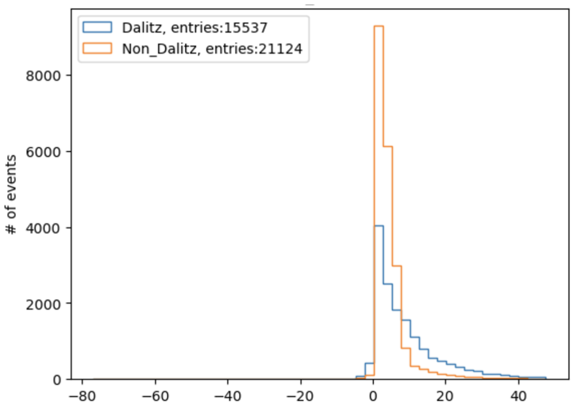

Figure 7: Ap_sigM

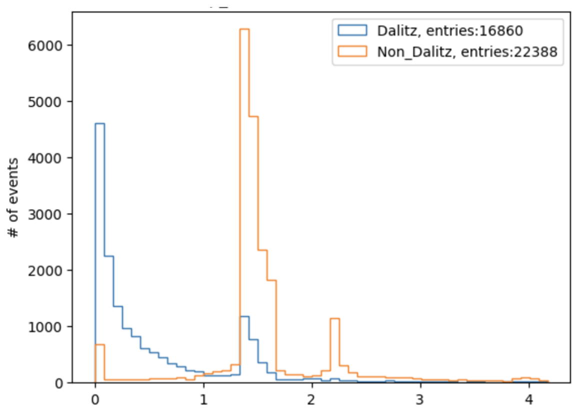

Figure 8: Ap_transverseVertexDistance

The MotherPDG information is used to classify events into Dalitz and Non-Dalitz categories. However, since this information is not available in real experimental data, the genMother features are removed after classification. By the end of this process, the number of features is reduced from 596 to 141. The dataset now contains six types of detected events: tau_sig, pi0, Ap, ep, em, and gamma, corresponding to τ^-, π^0, A', e^+, e^-, and photon, respectively. Additionally, the dataset includes features of vertex reconstruction difference, such as phidiff and dphidiff.

Next, the distributions of all remaining features are reviewed to ensure there are no issues, such as extremely small variances. Two features, gamma_clusterEoP and gamma_EoverP, are found to have variances on the order of 10^{-16}, making them essentially constant. These features are dropped, leaving 139 features in the cleaned dataset.

The dataset is currently imbalanced, consisting of 16,860 Dalitz events and 22,388 Non-Dalitz events. To address this problem, random sampling is applied to create a balanced dataset containing an equal number of signal and background entries. The final processed dataset comprises 33,720 entries and 139 features. Due to the limited number of entries, the dataset is split into 95% for training and 5% for testing. The dataset is normalized using a standard scaler, given the Gaussian nature of the features. Additionally, the standard scaler normalize each column independently, allowing feature selection to be directly performed on the normalized data. With these preparations, the dataset is now ready for model training.

3.2 Model Design

Two types of models, non-parametric and parametric, are selected to achieve signal classification tasks. The non-parametric model used is the Random Forest[4] (RF) classifier. This model is not only effective for classification but also provides insights into feature importance based on impurity reduction. By utilizing feature importance, less relevant features can be identified and excluded, leading to improved performance of the classifier. These less relevant features often contribute as noise, which can negatively impact the model's accuracy.

Feature selection plays a significant role in improving classification accuracy while also reducing the model's size. This makes the model more efficient to train, especially when working with large datasets. Additionally, feature selection helps preventing overfitting, which occurs when a model becomes overly specialized to the training data instead of learning general patterns. By focusing on the most important features, the model relies less on specific data points, improving its ability to perform well on unseen data.

The parametric model chosen for this project is the Artificial Neural Network (ANN) with fully connected layers, as the input dataset is in vector form and does not require feature extraction between columns. It offers greater flexibility and complexity, making it a powerful tool for classification tasks. Furthermore, feature selection determined by the non-parametric RF model will be applied to the ANN model to examine the impact of feature selection on its performance.

3.2.1 Non-Parametric Model

The non-parametric model is first trained on a dataset with 139 features. The Random Forest (RF) classifier is set with 100 estimators, using the Gini criterion, with other settings left at their default values from the scikit-learn package. Training the RF model is quick, taking about one minute, and achieves a testing accuracy of around 95%. The test results, shown in Figure 16, indicate strong performance, with precision and recall of approximately 95% and 96%, respectively. The feature importance of the 30 most important features, as estimated by the RF model, is presented in Figure 9. It shows that the most important features are related to the location of Dalitz decay, the displacement of electron-positron pairs, and their trajectory features, such as ep_cosTheta. This feature corresponds to certain angles where photons travel longer distances in the PXD layer, increasing the likelihood of photon conversion. Additionally, there is a noticeable drop in feature importance at the 27th feature, with the importance of features beyond this point falling below 0.01. This suggests that the first 26 features should be selected as a benchmark for further studies.

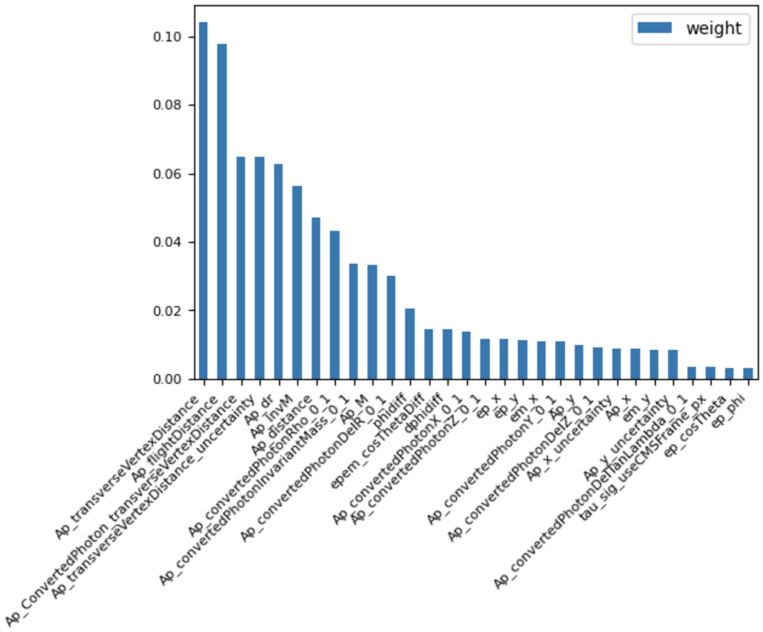

Figure 9: Random Forest importance

To reduce the impact of randomness in the RF training process on feature importance estimation, the column-dropping method is employed. This technique iteratively trains the RF model, removing the most important feature in each iteration until the desired number of features is reached. Based on the RF importance data, 30 features are selected for comparison at the point where the importance drops sharply. This iterative process takes about 30 minutes for 30 iterations, which can be more computationally expensive when applied to larger datasets. The first 26 features are chosen for comparison with the RF importance.

Table 1 presents a comparison of feature selection between two importance estimation methods: RF importance (denoted as R) and RF importance with column dropping (denoted as C). The table shows that the two methods agree well on the first 24 features, though their importance order differs. The primary difference is in the features Ap_y_uncertainty in R and Ap_convertedPhotonDelTanLambda_0_1 in C. Upon examining their distributions in Figures 13 and 14, both features do not have significant differences between Dalitz and Non-Dalitz events. However, Ap_y_uncertainty appears slightly more important, as its distribution shows less overlap between Dalitz and Non-Dalitz.

RF importance (R)	**RF importance with column dropping (C)**
Ap_transverseVertexDistance	Ap_flightDistance
Ap_flightDistance	Ap_dr
Ap_ConvertedPhoton_transverseVertexDistance	Ap_transverseVertexDistance
Ap_transverseVertexDistance_uncertainty	Ap_distance
Ap_dr	Ap_ConvertedPhoton_transverseVertexDistance
Ap_InvM	Ap_convertedPhotonRho_0_1
Ap_distance	Ap_transverseVertexDistance_uncertainty
Ap_convertedPhotonRho_0_1	Ap_InvM
Ap_convertedPhotonInvariantMass_0_1	Ap_convertedPhotonInvariantMass_0_1
Ap_M	Ap_M
Ap_convertedPhotonDelR_0_1	em_x
phidiff	Ap_convertedPhotonDelR_0_1
epem_cosThetaDiff	ep_x
dphidiff	Ap_x
Ap_convertedPhotonX_0_1	epem_cosThetaDiff
Ap_convertedPhotonZ_0_1	Ap_convertedPhotonDelZ_0_1
ep_x	Ap_y
ep_y	em_y
em_x	ep_y
Ap_convertedPhotonY_0_1	dphidiff
Ap_y	Ap_convertedPhotonZ_0_1
Ap_convertedPhotonDelZ_0_1	Ap_convertedPhotonX_0_1
Ap_x_uncertainty	Ap_convertedPhotonY_0_1
Ap_x	phidiff
em_y	Ap_convertedPhotonDelTanLambda_0_1
Ap_y_uncertainty	Ap_x_uncertainty

Table 1: Selected features based on RF importance and RF importance with dropping column (sorted by importance in descending order).

To assess the effect of feature selection, the RF classifier is retrained using the selected features. The model settings remain the same, with 100 estimators and the Gini criterion. An experiment was conducted using various numbers of features, and the results indicate that 26 features provide the optimal performance. The RF models trained with R feature selection (denoted as RFclass_r) and C feature selection (denoted as RFclass_c) both achieve a testing accuracy of around 95%. The precision and recall of these models, shown in Figures 18 and 20, demonstrate better performance compared to the model without feature selection (RFclass).

3.2.2 Parametric Model

The parametric model is based on an Artificial Neural Network (ANN) with fully connected layers, using ReLU activation in the hidden layers and a sigmoid activation in the output layer. Three models are considered, each with different input features: the model with 139 input features is called ANN, the model with feature selection using the R method is called ANN_r, and the model with feature selection using the C method is called ANN_c. These models are trained with early stopping technique and optimized by experimenting on hyperparameters such as the number of hidden layers, the number of neurons in each hidden layer, types of regularization and their strength, mini-batch size and the number of selected features to achieve the best performance. The optimized hyperparameters of architecture are shown in Table 2, and the optimal mini-batch size is 32 for all three models.

Layers	Features of layer	**ANN**	**ANN_r**	**ANN_c**
Input layer	Size	139	26	26
First hidden layer	Number of neurons	128	128	128
	Activation	ReLU	ReLU	ReLU
	Regularization	Dropout(0.4)	Dropout(0.2)	Dropout(0.2)
Second hidden layer	Number of neurons	64	64	64
	Activation	ReLU	ReLU	ReLU
	Regularization	Dropout(0.2)	Dropout(0.1)	Dropout(0.1)
Third hidden layer	Number of neurons	32	32	32
	Activation	ReLU	ReLU	ReLU
	Regularization	None	None	None
Output layer	Number of neurons	1	1	1
	Activation	Sigmoid	Sigmoid	Sigmoid
Total number of trainable parameters		28,289	13,825	13,825

Table 2: Architecture of the ANN models

All three optimized models have a similar architecture, with three hidden layers consisting of 128, 64, and 32 neurons, respectively. The first two hidden layers in each model use a Dropout regularizer [5]. It is important to note that the regularization strength applied to the models with feature selection (ANN_r and ANN_c) is half that of the ANN model. This suggests that feature selection helps preventing overfitting. Furthermore, the ANN_r and ANN_c models have trainable parameters which is less than half of the ANN model, leading to faster training. Specifically, each epoch of the ANN model takes about 3 seconds, while ANN_r and ANN_c take about 1 second per epoch. This difference is expected to become more significant with larger datasets.

The training performances of the artificial neural network (ANN) and its variations, ANN_r and ANN_c, are shown in Figure 10, Figure 11, and Figure 12, respectively. All models demonstrate strong performance.

The validation accuracy for the ANN model reaches approximately 95%. For ANN_r and ANN_c, the validation accuracy is slightly higher, reaching around 96%. This suggests that the modifications in ANN_r and ANN_c improve model performance. The training processes of all three models show a similar trend. Initially, the validation loss is lower, and as training continues, the losses of the models converge and begin to oscillate around each other. This pattern indicates good generalization to unseen data and shows that dropout regularization effectively prevents overfitting.

These figures also show spikes in both training and validation curves. The spikes during training are caused by the mini-batch gradient descent method, where smaller mini-batch sizes generally improve generalization but result in a longer training process. The training curves are smoother compared to the validation curves because training accuracy is calculated directly on the data used for training. In contrast, validation accuracy measures performance on unseen data, which introduces more fluctuations.

Larger models with many parameters are more likely to overfit, which can lead to inconsistent validation performance. Furthermore, dropout regularization can potentially add randomness which may lead to some instability in the validation results. Despite these fluctuations, the overall performance of the models remains strong.

The training and validation results for ANN, ANN_r, and ANN_c show high accuracy and good generalization. The observed fluctuations indicate the need for careful model design and regularization to balance performance and stability.

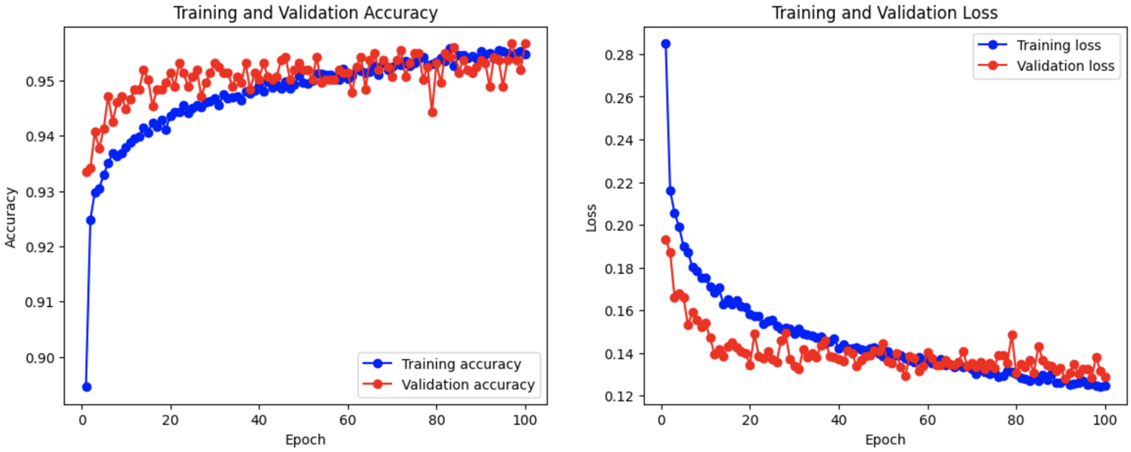

Figure 10: Training performance of ANN

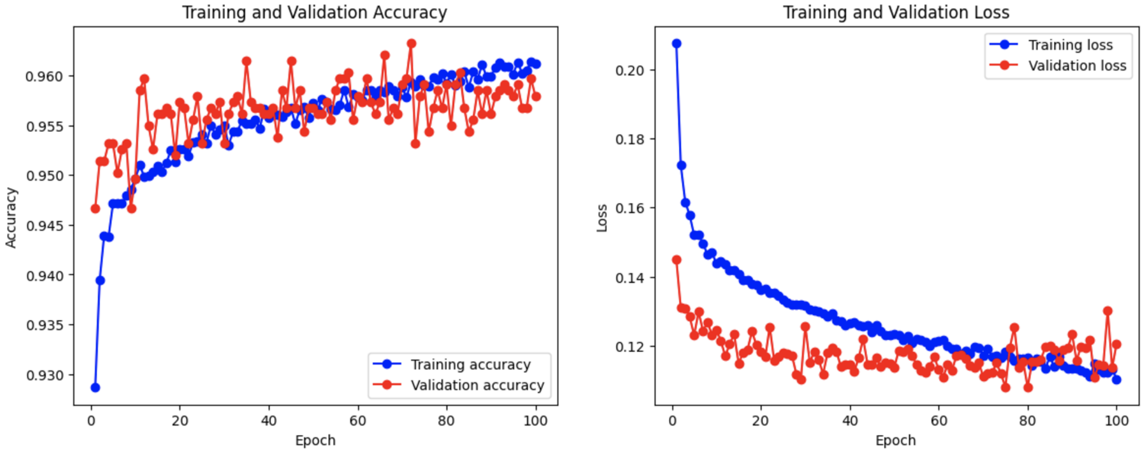

Figure 11: Training performance of ANN_r

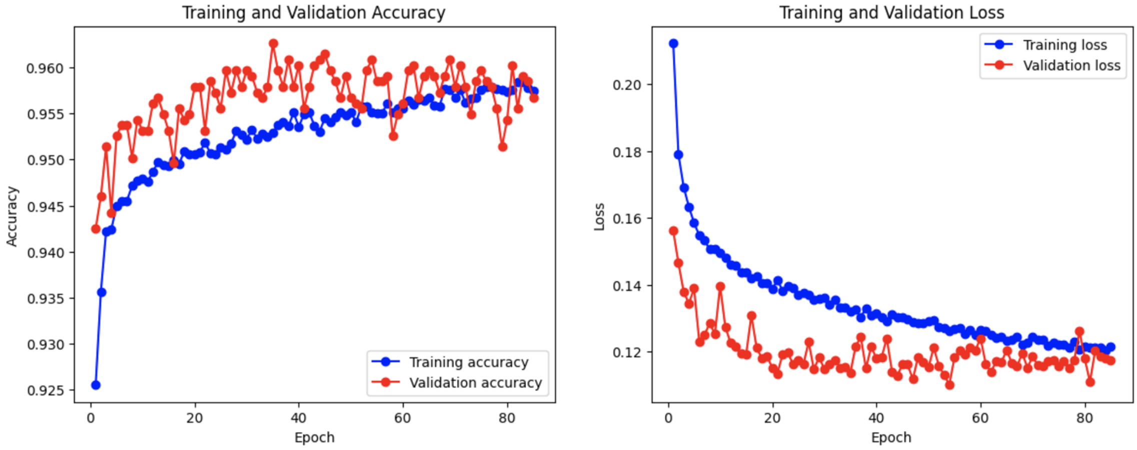

Figure 12: Training performance of ANN_c

3.3 Results & Discussion

Table 3 presents a performance comparison of six models, consisting of three Random Forest (RF) classifiers and three Artificial Neural Network (ANN) models. All six models achieve high performance, with accuracy, precision, and recall values around 95%.

Model	Number of input features	Accuracy	Precision	Recall
RFclass	139	0.9531	0.9495	0.9573
RFclass_r	26	0.9585	0.9574	0.9597
RFclass_c	26	0.9567	0.9541	0.9597
ANN	139	0.9567	0.9562	0.9573
ANN_r	26	0.9632	0.9643	0.9621
ANN_c	26	0.9626	0.9665	0.9585

Table 3: Model performance comparison.

When comparing RF and ANN models, the ANN classifiers achieve better performance across all metrics. This result is expected as ANN models provide more flexibility and complexity. ANN performance can be improved by adjusting hyperparameters, such as the number of neurons, depth of layers, regularization techniques, and learning rates. These factors provide greater control over model optimization. On the other hand, RF classifiers offer fewer flexibilities for parameter adjustment, limiting their potential for further improvement, but with faster training process.

A notable aspect of this analysis is the impact of feature selection. Models utilizing feature selection consistently achieve improved performance metrics. For both parametric (ANN) and non-parametric (RF) models, feature selection improves accuracy, precision, and recall. Specifically, the ANN models with feature selection (ANN_r and ANN_c) reach approximately 96% accuracy, precision, and recall, compared to the 95% range achieved by the ANN model without feature selection. This demonstrates the significant role of feature selection in enhancing model performance.

Among the models with feature selection, those using Random Forest feature importance without column dropping (RFclass_r and ANN_r) perform the best within their respective categories. A possible explanation is that the column-dropping method may negatively impact the selection process. When the most important features are removed in iterations, the model relies on less significant features for training. These features contribute less to the classification task, potentially leading to reduced accuracy. Additionally, the importance scores assigned to these less relevant features by RF may become unreliable and act randomly. This suggests that using the feature importance approach without column dropping is a more effective and consistent method for both RF and ANN models.

Another advantage of RF-based feature selection is its efficiency. Calculating feature importance with RF requires training the model only once, whereas the column-dropping method involves training the model multiple times, approximately equal to the number of features being evaluated. This makes the RF approach faster and more practical, especially for large datasets.

All these results show that feature selection is an effective method for reducing the dimensionality of data, making the training process faster and improving model performance. For future studies with larger Monte Carlo (MC) datasets, this approach can help create a reliable data processing pipeline for signal classification tasks.The suggested method involves using an RF classifier for feature selection, followed by training an ANN model with the selected features. This process not only reduces training time but also increases prediction accuracy and prevents overfitting. Additionally, this feature selection strategy can be applied to similar classification tasks in particle physics and other fields.

4 Conclusion

This project focuses on investigating the application of machine learning (ML) and deep learning (DL) techniques to classify Dalitz decays. The study uses two models: a Random Forest (RF) classifier, representing non-parametric methods, and an Artificial Neural Network (ANN), representing parametric approaches. Both models were trained on a balanced dataset of 33,720 entries, generated from Monte Carlo simulations.

The initial dataset contained 596 features, which is reduced to 139 after extensive data cleaning process. Using feature importance scores generated by the RF model, the most significant features were identified, further narrowing the input dimensions to 26. This selection process improves the model performance while addressing the challenges of overfitting and computational efficiency.

The RF classifier achieved a testing accuracy of approximately 95% while also providing critical insights into feature relevance. On the other hand, the ANN model, optimized with three hidden layers and dropout regularization, demonstrated greater flexibility and improved performance. By incorporating the selected features, the ANN achieved a classification accuracy of 96%, outperforming the RF model in accuracy and generalization capabilities.

This study shows feature selection plays a significant role in improving model performance for both RF and ANN for this specific classification task. It not only enhanced accuracy, precision, and recall but also reduced training time and the risk of overfitting. Moreover, this research establishes a framework for handling complex, high-dimensional data, making it applicable to similar classification tasks in particle physics or in other fields.

References

[1] A. J. Krasznahorkay et al. “Observation of Anomalous Internal Pair Creation in Be8 : A Possible Indication of a Light, Neutral Boson”. In: *Phys. Rev. Lett.* 116.4 (2016), p. 042501. DOI: 10.1103/PhysRevLett.116.042501. arXiv: 1504.01527 [nucl-ex].

[2] J. Kahn. “The Belle II Experiment”. In: *CERN-BINP Workshop for Young Scientists in e+e- Colliders.* 2017, pp. 45–54. DOI: 10.23727/CERN-Proceedings-2017-001.45.

[3] K. Kampf, M. Knecht, and J. Novotný. “The Dalitz decay $\pi^0 \to e^+e^-\gamma$ revisited”. In: *The European Physical Journal C* 46.1 (Feb. 2006), pp. 191–217. ISSN: 1434-6052. DOI: 10.1140/epjc/s2005-02466-7. URL: http://dx.doi.org/10.1140/epjc/s2005-02466-7.

[4] L Breiman. “Random Forests”. In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[5] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Appendix

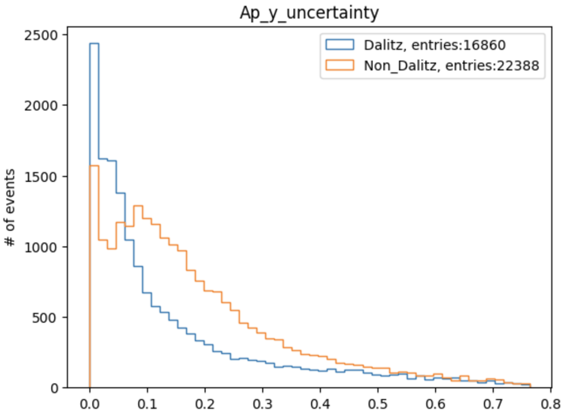

Figure 13: Ap_y_uncertainty

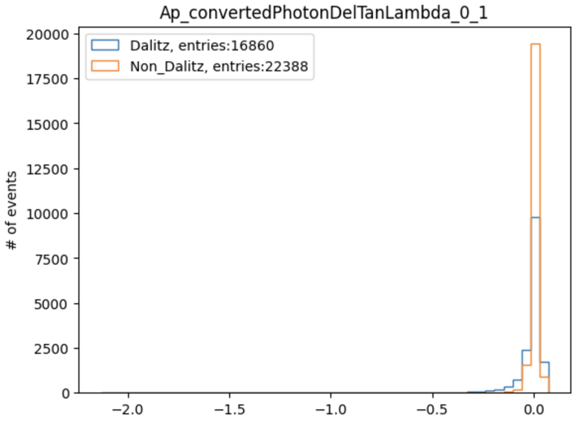

Figure 14: Ap_convertedPhotonDelTanLambda_0_1

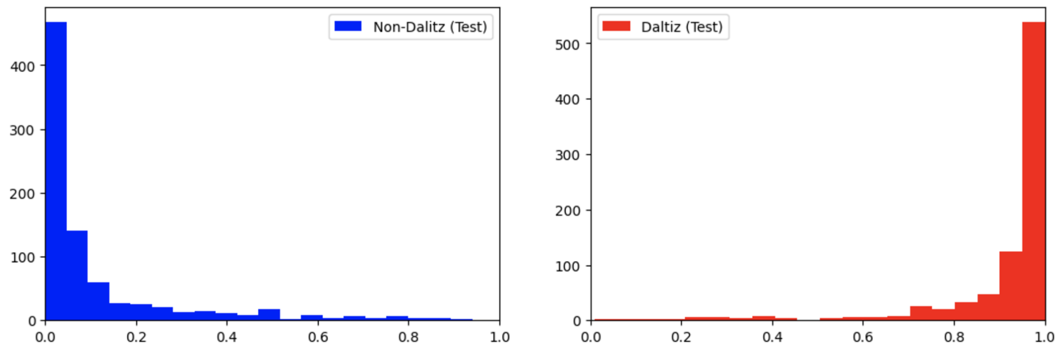

Figure 15: Predicted class probabilities for the test set (RFclass)

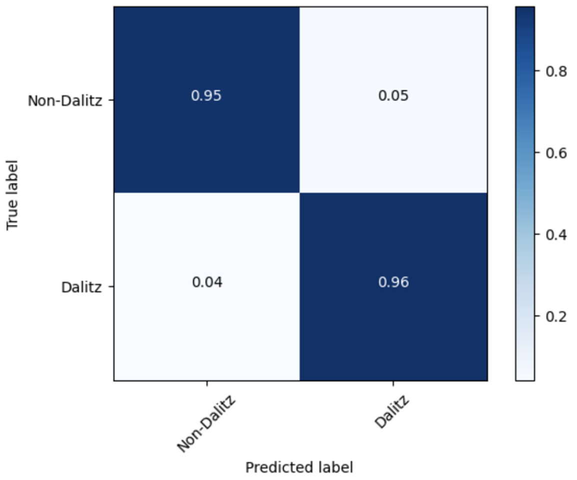

Figure 16: Normalized confusion matrix for the test set (RFclass)

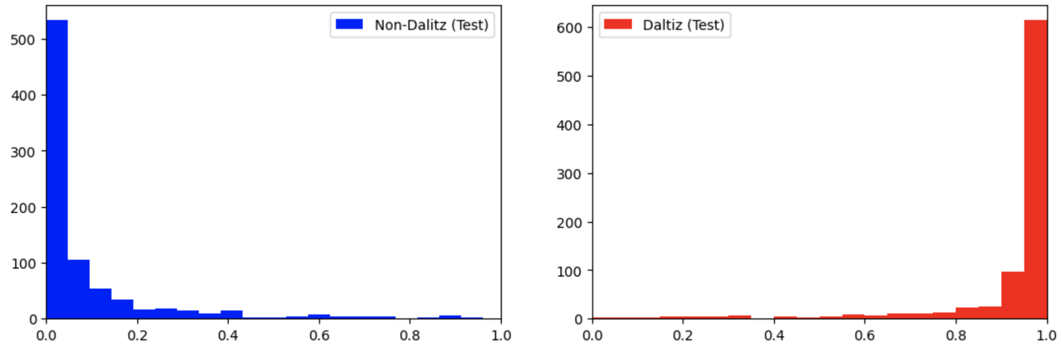

Figure 17: Predicted class probabilities for the test set (RFclass_r)

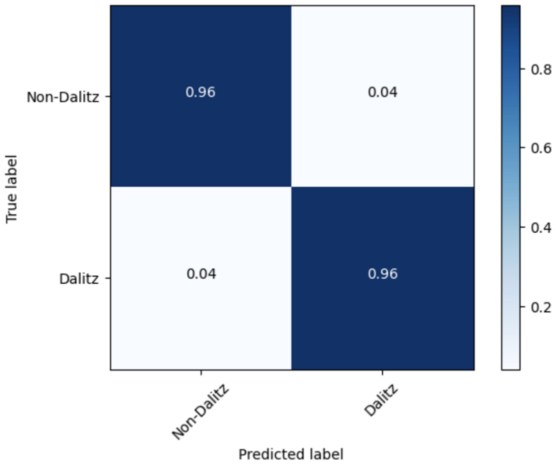

Figure 18: Normalized confusion matrix for the test set (RFclass_r)

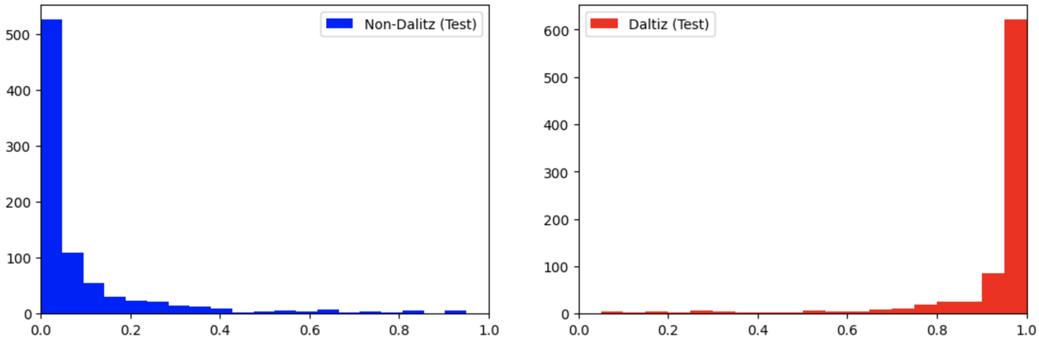

Figure 19: Predicted class probabilities for the test set (RFclass_c)

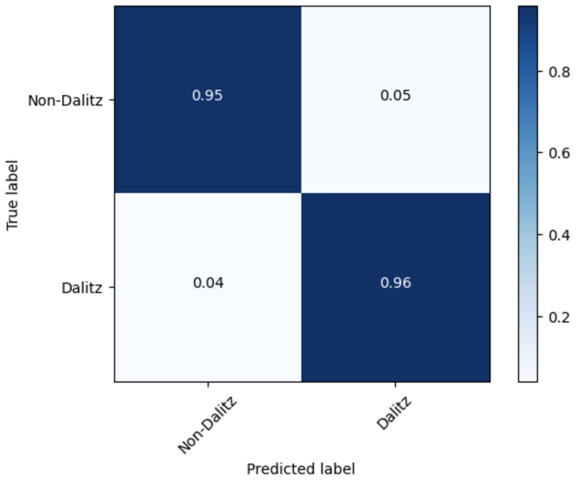

Figure 20: Normalized confusion matrix for the test set (RFclass_c)

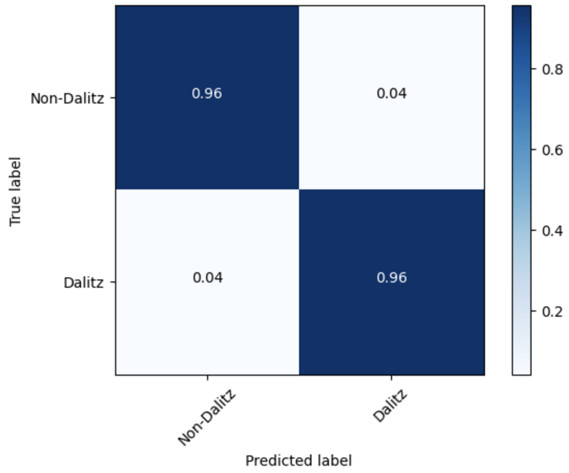

Figure 21: Normalized confusion matrix for the test set (ANN)

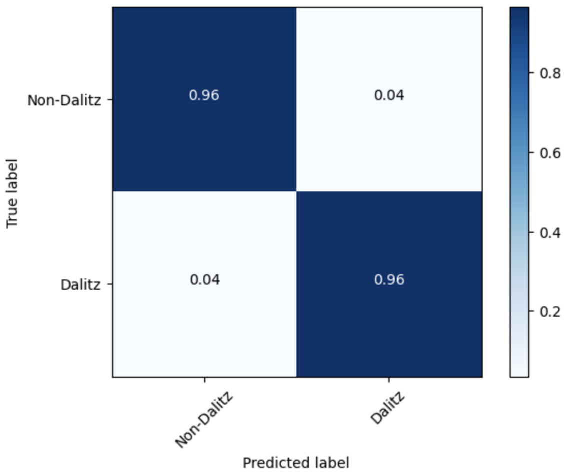

Figure 22: Normalized confusion matrix for the test set (ANN_r)

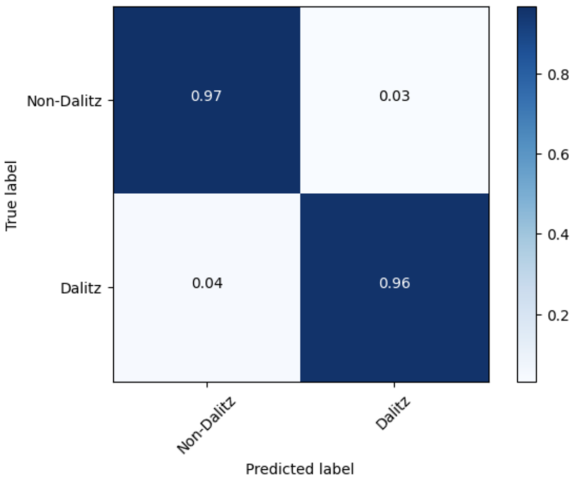

Figure 23: Normalized confusion matrix for the test set (ANN_c)