NATURAL SCIENCE REVIEW 3 100705 (2026)
nsr.jinr.int

Classification of muon tracks from charmonium decays and pion tracks in the model of the SPD detector using neural networks

A. R. Didenko*[1,2], I. V. Yeletskikh†[1], and A. O. Gridin‡[1]

[1] *Joint Institute for Nuclear Research, Dubna, 141980, Russia*
[2] *Lomonosov Moscow State University, Moscow, 119991, Russia*

Abstract

The Machine Learning (ML) approaches were applied to particle identification in the simulation of the Spin Physics Detector (SPD) at the NICA Collider. The results of the identification of muon tracks originating from charmonia decays and pion tracks in the intermediate momenta region (1.5–2.5 GeV/c) are presented. The obtained classifier accuracy is 77% while preserving 99% of muons and rejecting 48% of pions. The efficiency of developed binary classifier is demonstrated through background suppression in $J/\psi \to \mu\mu$ decays.

Keywords: muon identification, machine learning, evolutionary algorithm, DNN, SPD

DOI: 10.54546/NaturalSciRev.100705

1. Introduction

In this paper, the muon–pion classification is performed using the ML approaches for tracks reconstructed in simulation of the SPD detector [1]. Hadron tracks of intermediate momenta (1.5–2.5 GeV/c) may be difficult to distinguish from muon tracks based only on individual parameters of reconstructed track. Muons do not participate in strong interactions, whereas low-energy hadrons often lack sufficient kinetic energy to initiate hadronic showers in the detector material. As a result, hadron trajectories exhibit the same number of hits and track length in tracking detectors as muon tracks. In particular, the reconstruction of $J/\psi \to \mu\mu$ decays may suffer from charged pion background in 1.5–2.5 GeV/c momenta range. Another contamination comes from charged pions that decay into muons in flight, producing secondary tracks that fake prompt muons.

Scheme of the SPD setup is shown in Figure 1 [1]: beam pipe (*1*), vertex detector (*2*), straw-tube tracker (*3*), Time-Of-Flight (TOF) system (*4*), Electromagnetic Calorimeter (ECal) (*5*), magnet (*6*) and Range System (RS) (*7*), straw tracker end-cap (*8*). Event reconstruction

*Corresponding author e-mail address: alisadidenko@jinr.ru
†Corresponding author e-mail address: ivaneleckih@jinr.ru
‡Corresponding author e-mail address: andreigridin@jinr.ru

Received: 11 February 2026; Revised: 9 June 2026; Accepted: 15 June 2026; Published: 26 June 2026
Copyright: © by the author(s), 2026, under license to Joint Institute for Nuclear Research. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International license (CC BY-NC License 4.0).

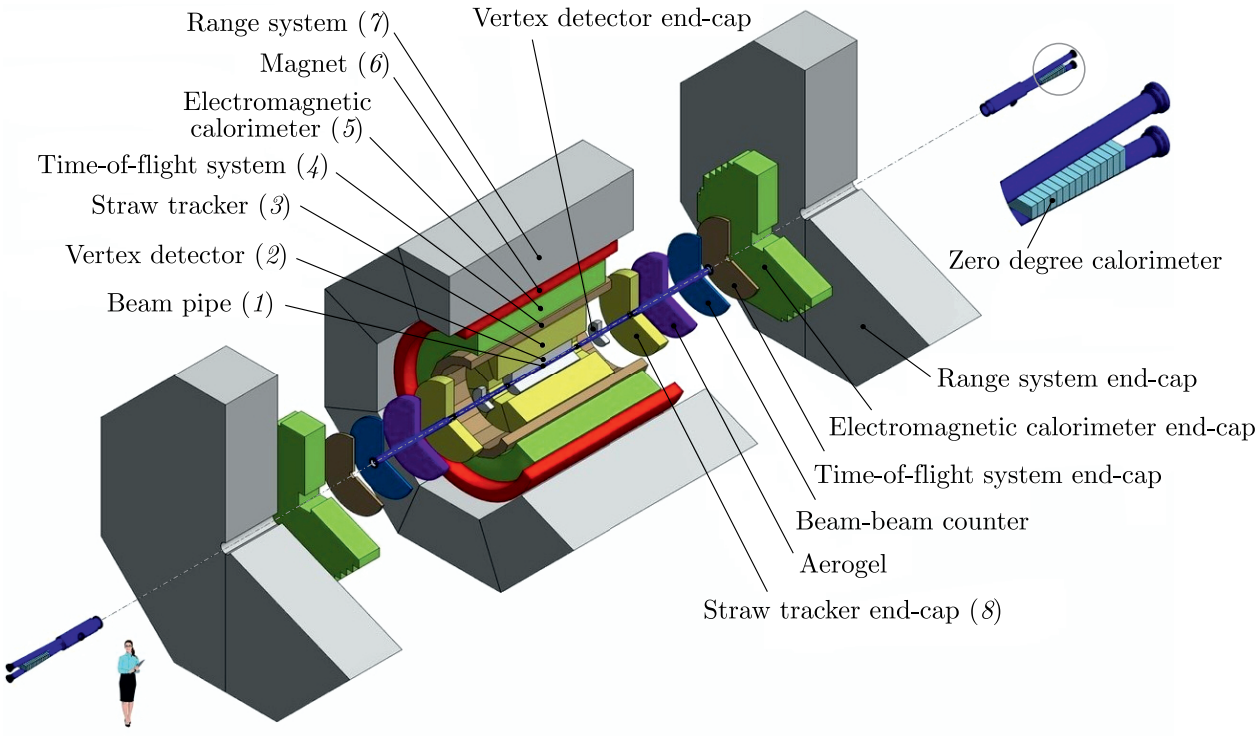

Figure 1. Schematic of the SPD [1].

responses on the combined response of several detector subsystems. The muon–pion separation can make use of track-level observables reconstructed in these subsystems, including track length, angles, hit multiplicities in each subdetector, TOF-based velocity estimation, and calorimeter response. The ML approaches, and in particular Deep Neural Networks (DNNs), can significantly improve the separation power between muons and low-energy hadrons at the SPD by accounting for multi-dimensional correlations of track parameters.

2. Simulated data samples and selections

Around 1.5×10^5 fully reconstructed events (corresponding to approximately 2 million tracks) featuring charmonium production were generated using the Pythia8 event generator within the SPDroot framework [2]. Charmonium states were produced using the `Charmonium:all` process in Pythia8, which includes J/ψ, $\psi(2S)$, χ_{cJ}, and η_c states generated at LO within the NRQCD formalism. Both direct production and feed-down contributions from higher-mass charmonium states are taken into account, with relative yields determined by the production cross sections calculated internally by Pythia8. The J/ψ mesons were forced to decay exclusively via the dimuon channel, $J/\psi \to \mu^+\mu^-$, while the dielectron channel was disabled due to the limited reconstruction capabilities for this mode in the SPD detector. Decay modes of other charmonium states were left unchanged. As a result, the simulated events contain muons predominantly originating from J/ψ decays, as well as pions and other hadrons produced in accompanying processes and in decays of non-J/ψ particles. The χ^2/n_{dof} of the track fit is required to be less than 10 in order to reject poorly reconstructed tracks.

The transverse momentum distribution for μ, π, p, and K is presented in Figure 2. There is a small contribution from non-prompt muons originating from pion decays. These muons are softer compared to prompt muons (from charmonia decays, in particular J/ψ), and most of them reside below 1.5 GeV/c. Transverse momentum spectra are not reliable discriminating variables, since they strongly depend on the specific physical process contributions. A transverse

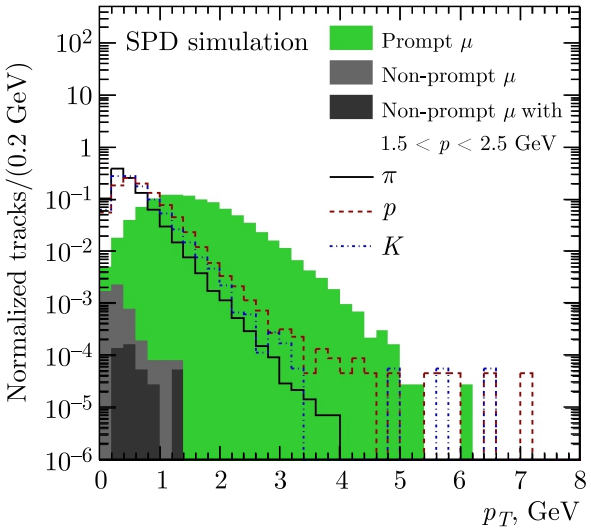

Figure 2. Transverse momentum spectra for muons, pions, protons, and kaons, each normalized to the respective total number of tracks of identified tracks of the corresponding species. All prompt muons and non-prompt muons are shown in green and gray, respectively. The dark gray shaded component corresponds to non-prompt muons within the total momentum range of 1.5–2.5 GeV/c.

momentum selection can be applied in addition to the identification result when muons from a specific process (e.g., J/ψ decay) are selected.

While protons and kaons can be classified using simple track properties, these particle samples are not used for training the ML model. The muon–pion separation is more difficult, especially in the total momentum range of 1.5–2.5 GeV/c. This classification task is performed using the ML approaches. Only prompt muons are used as signal, and charged pions are used as background for training. Particle types are distinguished using kinematic and track-based characteristics. The following variables were used for ML training: pseudo-rapidity (Figure 3, a), number of hits in the vertex detector (Figure 3, b), number of hits in the straw-tube tracker (Figure 3, c), number of hits in the straw tracker end-cap (Figure 3, d), number of hits in inner part of detector (vertex detector, straw-tube tracker, TOF system, and the ECal) (Figure 3, e) and track length in this inner part of detector (Figure 3, f), number of hits in the RS (Figure 3, g), and track length in the RS (Figure 3, h).

The degree of signal–background separation is quantified using the so-called Separation Power (SP):

$$\mathrm{SP} = \frac{1}{2}\sum_{n=1}^{n\mathrm{Bins}} \frac{(s_n - b_n)^2}{s_n + b_n} \times 100\%, \tag{1}$$

where s_n and b_n denote the normalized value in the n-th bin of signal (prompt muons) and background, respectively. Particles on their way to the RS experience multiple scattering and interact with the magnetic field. Compared to muons, fewer hadron tracks reach the RS or have long extrapolated tracks. As a result, the parameters of the tracks reaching the RS provide strong muon–hadron discrimination [1]. The number of hits and track length in the RS (Figure 3, g and h) exhibit the best SP.

For muon–pion separation, the DNN model learns to recognize patterns in correlated tracking variables. The significant complex non-linear correlations observed between certain track parameters represent features that the network can effectively combine for classification. As an example, a two-dimensional dependence of the total number of hits and the absolute value of pseudo-rapidity is given in Figure 4.

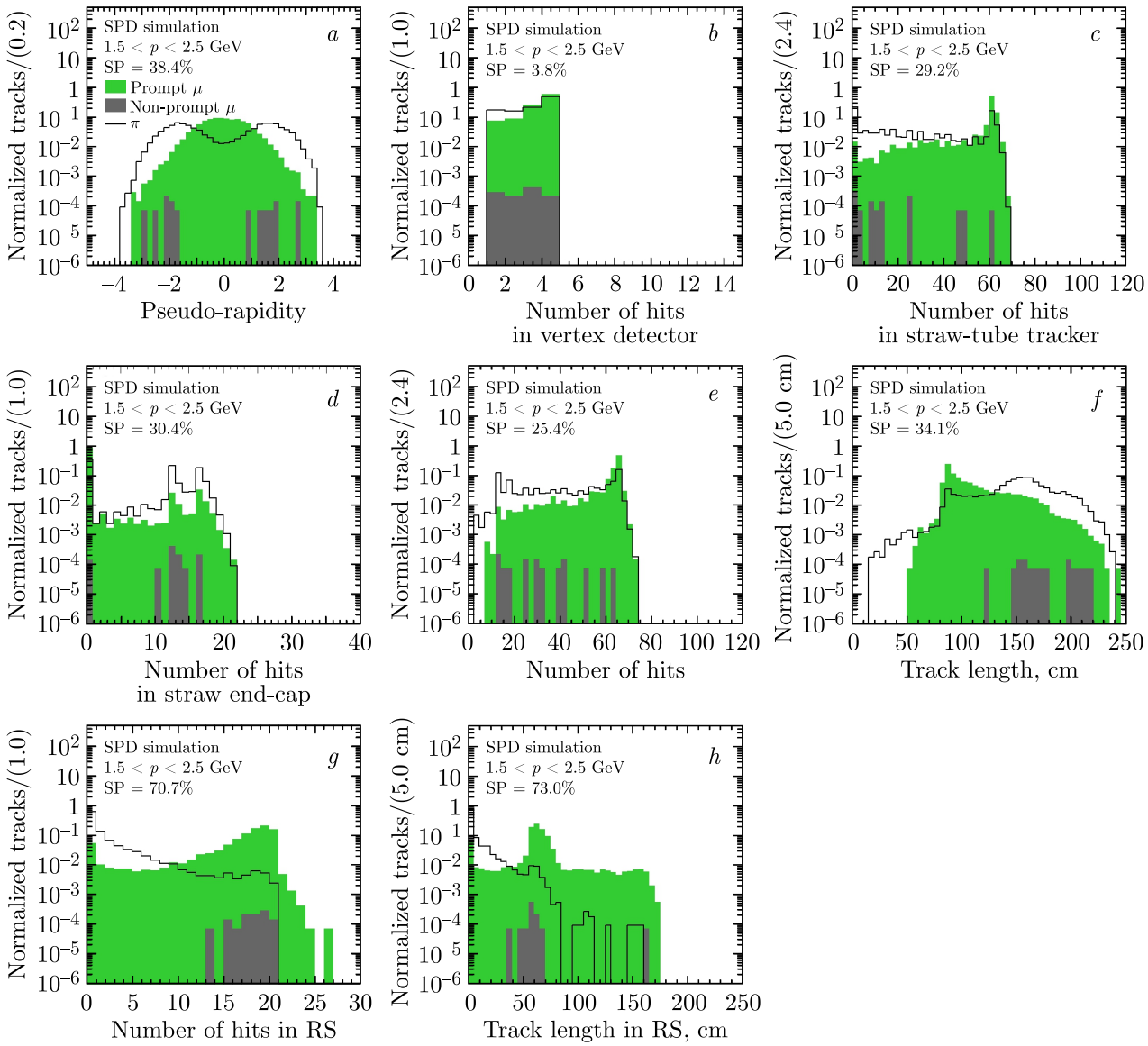

Figure 3. Distributions of the pseudo-rapidity (a), number of hits in different detector subsystems (b–d), total number of hits (e) and track length (f) in inner detector subsystems, number of hits in the Range System (g), and track length in the Range System (h) for μ and π after applying the selection criteria. Distributions are normalized to the number of tracks. The prompt μ and non-prompt μ are shown in green and gray, respectively.

3. Machine learning approach

Numerous software packages implement the ML algorithms (e.g., neural-network-based classifiers) for binary classification problems. Finding optimal parameter set of the ML model is called training. Typical training algorithms attempt to combine the advantages of deterministic gradient descent and stochastic search, ultimately aiming to minimize a loss function that quantifies the discrepancy between the true target value and the ML model prediction. In practical tasks, reaching a sufficiently deep local minimum of this loss function provides an adequate solution. However, even in case of regular loss functions, gradient-based optimizers can be quickly trapped in local minimum of the network parameter space. This problem is further aggravated in high-dimensional, non-convex loss-function landscapes typical of fully connected networks. Evolutionary [3, 4] and hybrid optimization algorithms can be more effec-

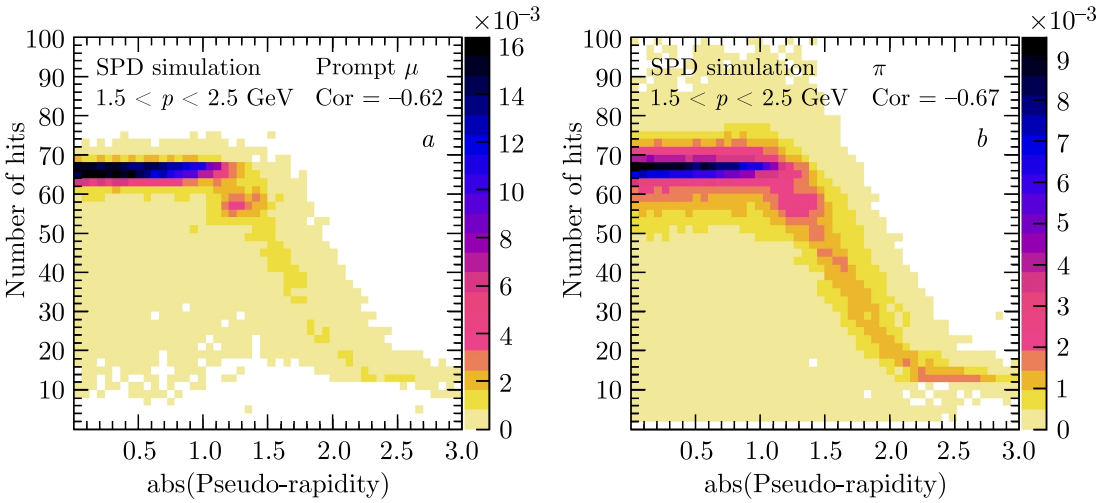

Figure 4. Two-dimensional distribution of the total number of hits and the absolute value of pseudo-rapidity for the prompt muons (a) and pions (b) and the linear correlation coefficients (Cor) for them.

tive (though commonly in cost of extra CPU consumption) in locating global minimum of the complex/irregular loss functions.

We developed a classification method based on a fully connected DNN implemented in C++, which uses an evolutionary training algorithm for parameter optimization. In this approach, the usage of a non-regular loss function (e.g., signal significance or area under selection efficiency curve) is not forbidden and may potentially provide improved classification performance.

Our evolutionary algorithm considers the complete set of trainable weights as an individual genome. A population of 50 networks is initialized with randomly sampled weight configurations. During each subsequent generation, the population of networks is evaluated according to a fitness measure — the total Area Under the Curve (AUC) for the Receiver Operating Characteristic curve (ROC curve). The ROC curve is constructed by plotting the signal efficiency (also referred to as True Positive Rate, TPR) on the y-axis against the background efficiency (also referred to as False Positive Rate, FPR) on the x-axis. Both quantities are thus in the range from 0 to 1. The AUC is defined as

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR}). \tag{2}$$

The best-performing network is selected as the "parent" for the next generation, and its "offsprings" are generated by applying random variations to the parameters ("genome mutations"). Offsprings replace the previous generation of networks. The procedure is repeated until improvements between generations become infinitesimal. Potential overtraining is controlled by applying a penalty to the fitness measure equal to twice the absolute difference between the network AUC values computed on statistically independent training and testing sets of samples.

Hyperparameter values — number of layers, neurons per layer, type of neuron activation profiles, input variable selection, and training algorithm settings — may significantly affect neural network performance. In this study, network architecture consists of two hidden layers with 15 and 9 neurons, respectively. The hyperbolic tangent function is used as neuron activation profile. Subsequent improvement may be achieved in future by using evolutionary strategy for both parameter and hyperparameter optimization.

A comparative analysis shows that the evolutionary optimized DNN achieves performance similar to that of a standard Keras [5] Multi-Layer Perceptron (MLP) trained with the Adam

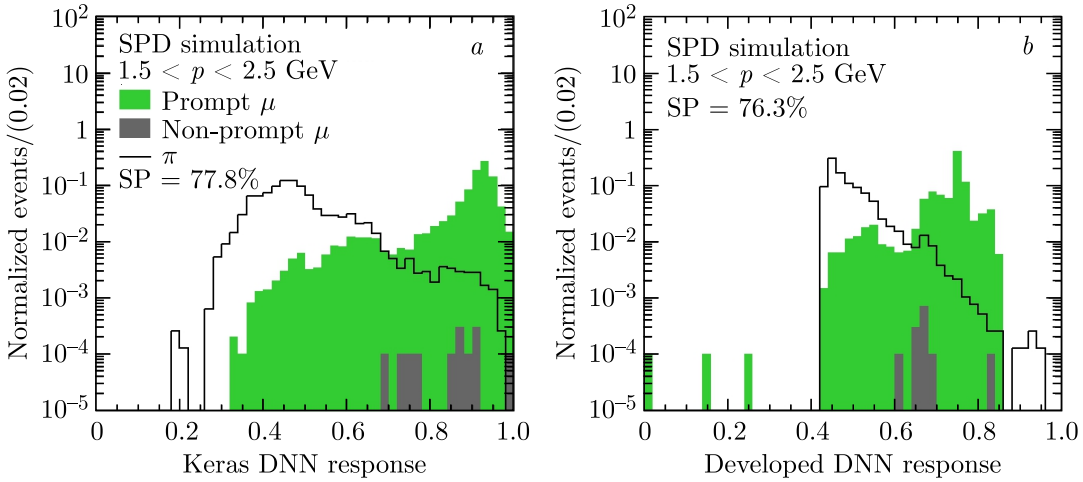

Figure 5. Neural network response of Keras DNN (a) and evolutionary DNN (b). The prompt μ and non-prompt μ are shown in green and gray, respectively.

optimizer and using 32 and 16 neurons in the hidden layers (Figure 5). The evolutionary algorithm demonstrates a strong capability to explore the parameter space while maintaining acceptable computational efficiency.

4. Results

The core result is the demonstrated ability to separate muons and pions effectively using the developed algorithm. The DNN models were trained on the set of track property variables and a kinematic variable pseudo-rapidity. Comparison of neural network response of Keras and developed DNNs for muons and pions is shown in Figure 5, a and b. The Keras model is slightly better by SP and AUC values, the new ML algorithm gives a comparable result in signal significance — the metric most directly related to physics analysis sensitivity. The expected sensitivity is quantified using the Poisson significance:

$$Z_n = \frac{S_n}{\sqrt{S_n + B_n}}, \tag{3}$$

where S_n and B_n are the fractions of signal and background tracks passing the threshold n, each normalized to its own total number of tracks before the DNN output cut (Figure 6, a). The optimal operating point corresponds to the threshold that maximizes Z_n. The significance plot is shown in Figure 6, b. The application of a selection criterion $DNN\,output > 0.633$ to a dataset containing 13,923 signal and 11,088 background tracks results in an increase of the signal significance from 0.707 to 0.925 by a factor of 1.3, with a signal efficiency of 89.69% and a background efficiency of 4.34%, as indicated by the square marker in Figure 6, a and b. For comparison, the Keras-based DNN reaches its maximum significance at a classifier threshold of 0.669, with a signal efficiency of 91.15% and a background efficiency of 5.11% (Figure 6, a and b). The ROC curves in Figure 6, c demonstrate the high performance of both classifiers. The developed DNN achieves a comparable AUC (0.968) to the Keras baseline (0.975), confirming its excellent capability to distinguish muons from pions across all classification thresholds.

As an additional explicit illustration of the identification algorithm, we reconstruct the J/ψ invariant mass and estimate the background under the J/ψ peak arising from pion tracks. By incorporating common vertex reconstruction information from di-muon pairs for J/ψ identification, non-prompt muons are completely removed. The reconstruction of the J/ψ invariant mass

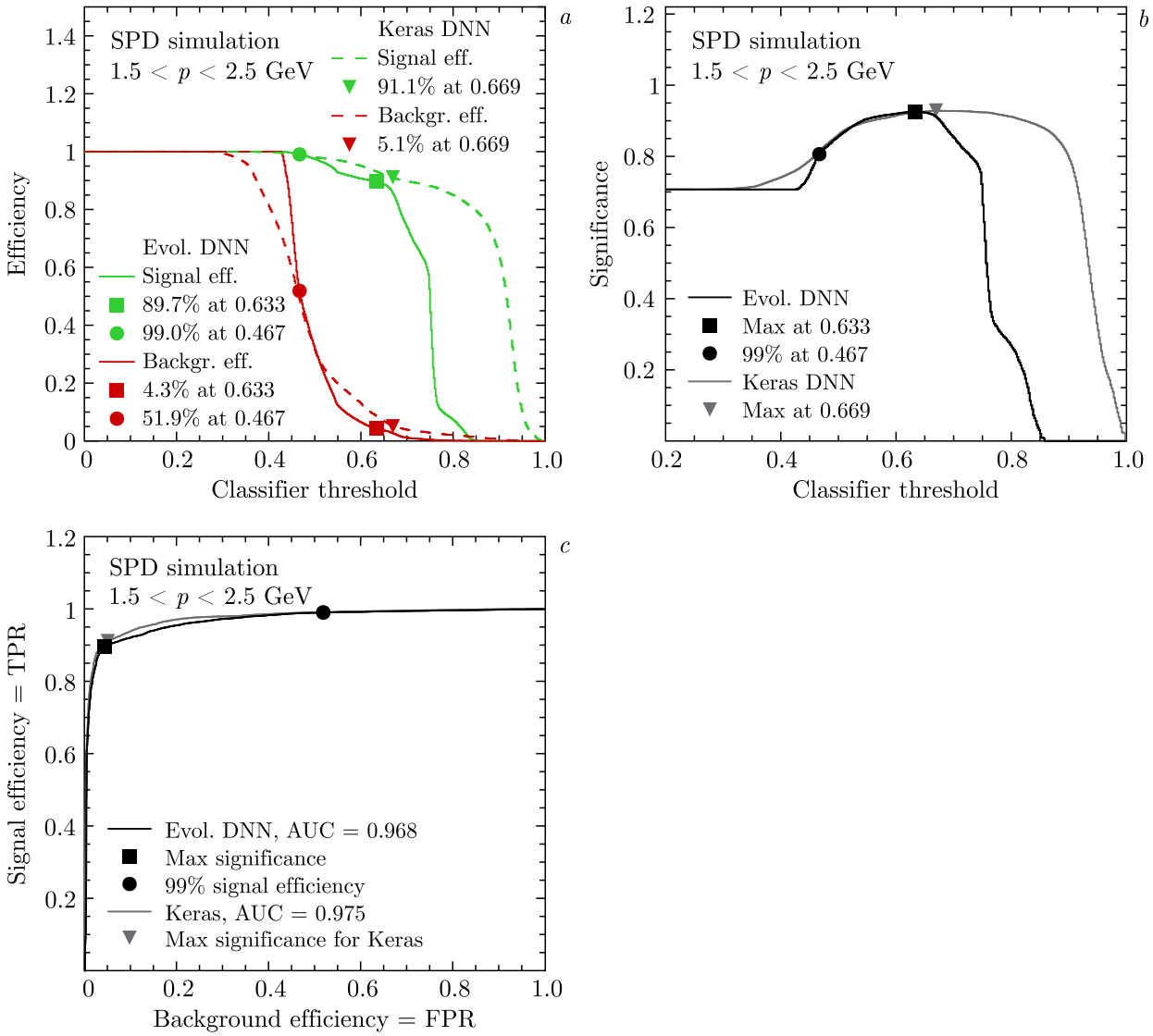

Figure 6. Efficiency of signal and background as a function of the classifier threshold for the evolutionary DNN and Keras DNN (*a*). The comparison of significance (*b*) and ROC (*c*) curves for the signal muons. The maximum discovery significance for the evolutionary DNN is achieved at a network output threshold of 0.633, indicated by the black square. A signal efficiency of 99% corresponds to a DNN output value of 0.467, marked with a black circle. The maximum significance for the Keras DNN is indicated by the gray triangle.

is shown in Figure 7, *a*. The invariant mass spectrum after applying the DNN classifier shows a powerful background suppression (Figure 7, *b*–*e*). One of the optimal cuts can be defined by requiring the maximum of signal significance. A cut on the classifier output more than 0.633 successfully rejects approximately 95.7% of the pion background while retaining 89.7% of the muons (Figure 7, *d*). Possible alternative cut on the classifier output value over 0.467 is keeping 99% of muons (Figure 7, *b*). The classification accuracy is defined through the Confusion Matrix elements (Figure 7, *c* and *e*):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{S + B}, \tag{4}$$

where TP (True Positive) corresponds to muons correctly classified with *DNN output* > *threshold*, and TN (True Negative) corresponds to pions correctly rejected with *DNN output* ⩽ *thresh-*

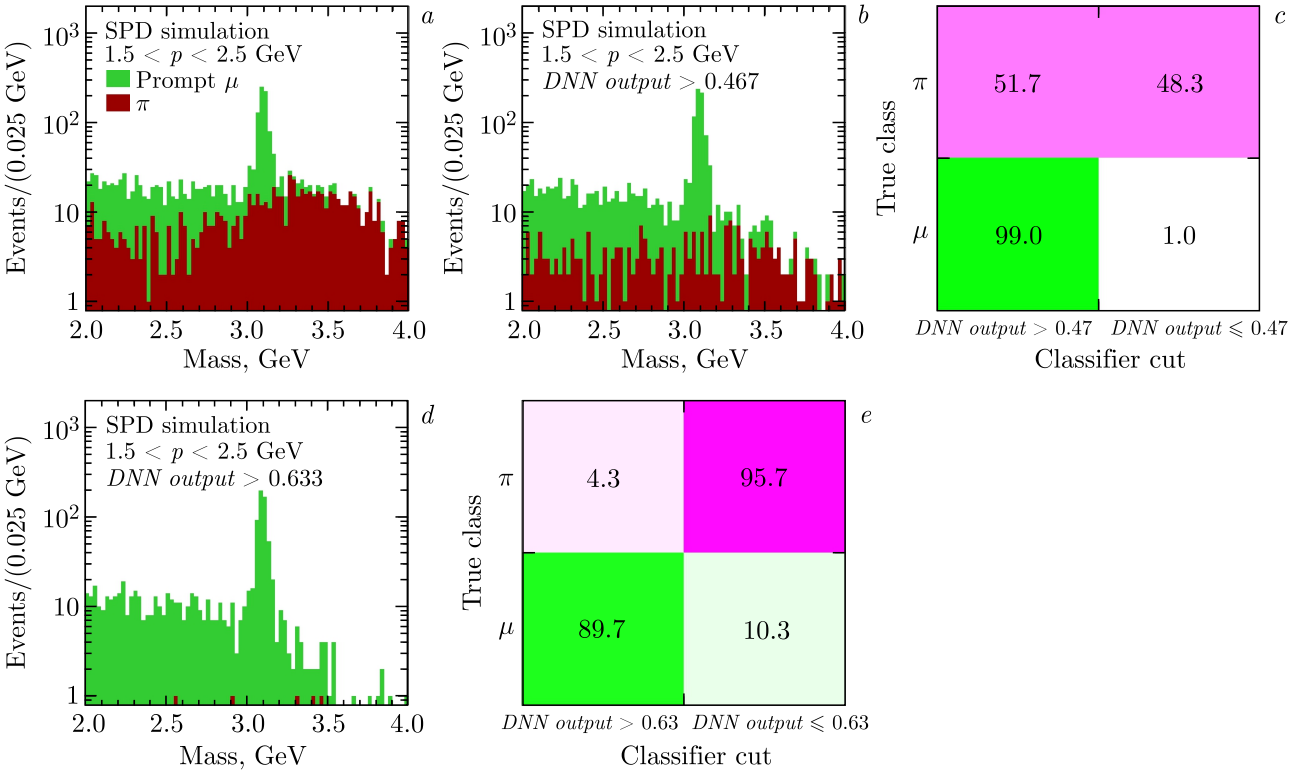

Figure 7. Reconstructed mass spectra before DNN output cuts (*a*), after soft cut (*DNN output* > 0.467) (*b*), and after tight cut with maximum of signal significance (*DNN output* > 0.633) (*d*). The plots *c* and *e* show the Confusion Matrix elements in percents at signal threshold equal to 0.467 and 0.633. White–green (white–pink) scale represents the fraction of preserved muons (pions) in percents.

old, while S and B are total count of muons and pions, correspondingly. The developed DNN gives muon accuracy equal to 92% for 0.633 classifier threshold and 77% for 0.466 classifier threshold.

5. Conclusion

The developed classification algorithm demonstrates good efficiency in separating muons from pions within the momentum range of 1.5–2.5 GeV/c. The developed DNN architecture with evolutionary optimization achieves approximately 95.7% pion suppression while maintaining a high muon identification efficiency of 89.7%. This performance confirms the method's suitability for particle identification tasks in intermediate momentum ranges, providing a valuable tool for experiment.

Future developments could focus on integrating additional discrimination variables to enhance classification performance (such as track dE/dx estimation) as well as developing dedicated algorithms for network hyperparameter optimization.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] SPD Collaboration, Technical Design Report of the Spin Physics Detector at NICA, Natural Sci. Rev. 1 (1) (2024). doi:10.48550/arXiv.2404.08317.

[2] SPD Collaboration, SPDRoot, https://spd.jinr.ru/spd-software/.

[3] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence through Simulated Evolution, New York, 1966.

[4] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, 1975. doi:https://doi.org/10.7551/mitpress/1090.001.0001.

[5] F. Chollet et al., Keras: The Python Deep Learning Library, https://keras.io/, accessed: 2018-06.