Published by IOP Publishing for Sissa Medialab

Received: *November 12, 2023*
Accepted: *December 28, 2023*
Published: *January 25, 2024*

Methods of machine learning for the analysis of cosmic rays mass composition with the KASCADE experiment data

M.Y. Kuznetsov,[a,*] **N.A. Petrov,**[a,b,c] **I.A. Plokhikh**[a,d,e] **and V.V. Sotnikov**[a]

[a] *Institute for Nuclear Research of the Russian Academy of Sciences,*
60th October Anniversary Prospect 7a, Moscow 117312, Russia
[b] *Department of Physics, Novosibirsk State University,*
Pirogova street 1, Novosibirsk 630090, Russia
[c] *Budker Institute of Nuclear Physics SB RAS,*
Akademika Lavrentieva prospect 11, Novosibirsk 630090, Russia
[d] *Kutateladze Institute of Thermophysics SB RAS,*
Akademika Lavrentieva prospect 11, Novosibirsk 630090, Russia
[e] *Department of Mathematics and Mechanics, Novosibirsk State University,*
Pirogova street 1, Novosibirsk 630090, Russia

E-mail: mkuzn@inr.ac.ru

Abstract: We study the problem of reconstruction of high-energy cosmic rays mass composition from the experimental data of extensive air showers. We develop several machine learning methods for the reconstruction of energy spectra of separate primary nuclei at energies 1–100 PeV, using the public data and Monte-Carlo simulations of the KASCADE experiment from the KCDC platform. We estimate the uncertainties of our methods, including the unfolding procedure, and show that the overall accuracy exceeds that of the method used in the original studies of the KASCADE experiment.

Keywords: Analysis and statistical methods; Data analysis; Particle identification methods

[*] Corresponding author.

https://doi.org/10.1088/1748-0221/19/01/P01025

Contents

1 Introduction

Mass composition is one of the main problems in the physics of high-energy cosmic rays (CR) [1]. Despite decades of experimental studies the precise mass composition above the so-called knee of the CR spectrum ($E \gtrsim 10^{15}$ eV) is not known. While the general trend of composition becoming heavier with energy is expected from theory [2, 3], the results of the various cosmic rays experiments are incompatible with each other [4–6]. The knowledge of the spectra of separate mass components in the $10^{15} < E < 10^{18}$ eV energy range is important for the understanding of their origin, in particular, the transition between the galactic and the extragalactic cosmic rays is expected somewhere in this range [1].

The analysis of CR properties in this energy range is complicated by several issues. Unlike CRs of lower energies, these particles can be detected only indirectly, via so-called extensive air showers (EAS) of secondary particles, that they produce in the Earth's atmosphere. The longitudinal and lateral structure of these showers can be recorded by various experimental techniques: Cherenkov and fluorescence light detection [6], detection of charged particles shower on the Earth surface and muon part of the shower underground [4, 5]. Then the properties of the primary particle can be reconstructed using this data. It is relatively simple to reconstruct the direction of the primary CR from the geometry of the shower and the CR energy from both the geometry and lateral distribution function (LDF) of secondary particles. At the same time, the accurate reconstruction of the primary particle type is much more difficult since the showers produced by different nuclei are not that different. One can either look for a longitudinal development of the shower in the atmosphere deriving the atmospheric depth of the maximum light emitted by the shower, the so-called depth of shower maximum (X_{max}), that is a proxy of a primary particle mass. Alternatively, one can analyze the charged particle content of the shower on the ground and its muonic content underground — the heavier the primary nucleus — the larger the total number of muons. Then, comparing these observables with the Monte-Carlo simulations of air showers from various primary nuclei one can hope to reconstruct the mass composition of primary CRs or even the energy spectra of separate mass components of the CR flux. The problem here is that all the mentioned observables are dependent on the hadronic interaction models used in the Monte-Carlo simulations. So the mass component spectra reconstructed from one and the same experimental data using different Monte-Carlo models may differ significantly [7].

In the present study, we aim to develop modern methods to reanalyze the original data of the KASCADE experiment and to re-derive the CR mass components spectra from it. The method we choose for this analysis is machine learning (ML). Its benefits for the field of cosmic-ray experiments data analysis were proven by many recent studies [5, 8–12]. The key idea behind the applicability of these methods is that the data from surface detectors about an EAS event can effectively be interpreted as an image, which opens the way for extensive usage of the machine learning methods developed for image analysis recently. Another idea is that the information viable for mass composition analysis of CRs is likely not accumulated in a few observables of EAS by rather dispersed in the whole "image" of the EAS footprint, so that the ML methods that are working with a huge number of variables is expected to efficiently "convert" this image back to a type of the primary particle.

The paper is organized as follows. First, we introduce the KASCADE experiment, its data, Monte-Carlo, and reconstruction in section 2. Then we present the machine learning methods we develop for this study in section 3. We make an initial estimation of the performance of all these ML methods using KASCADE Monte-Carlo in section 4. In section 5 we perform several tests and estimate the main uncertainties for the convolutional neural network method (CNN) that showed the best performance. In section 6 we discuss and test the unfolding procedure for the reconstruction of the separate mass components spectra. We estimate all uncertainties related to this procedure, as well as the overall accuracy of the mass components spectra reconstruction. We apply the CNN + unfolding to a small part of the KASCADE data set that we call "unblind data", and compare the results with the original KASCADE mass composition analysis. We conclude in section 7.

2 Experiment, data and Monte-Carlo

KASCADE is an extensive air shower experiment that was located in KIT Campus North, Karlsruhe, Germany (49.10°N, 8.44°E), at 110 m a.s.l., corresponding to an average atmospheric depth of 1022 g/cm^2 [13]. It started to operate in 1996, underwent an extension called KASCADE-Grande in 2003, and finished the data acquisition in 2013. The KASCADE experiment (without Grande extension), which data is used in the present study, was comprised of 252 scintillator detectors placed in a rectangular grid with 13 m spacing, covering the area of $200 \times 200\,\mathrm{m}^2$ in total. This array is capable of detecting a secondary particle footprint of extensive air shower initiated by cosmic rays with primary energies in $\sim$ 500 TeV–100 PeV energy range. Detectors in an outer part of the grid contained a metal shielding layer between scintillator layers so that the upper scintillator layer detects a e/γ-dominated signal while the lower one detects a muon-dominated signal. The KASCADE experiment also contained several other parts: a central detector, an underground muon tracking detector, etc., but in the present study, we only use the data from the main detector array. A detailed description of the experiment is given in refs. [13, 14].

In the present study, we use the KASCADE Monte-Carlo sets and preselection data sets provided by the KASCADE Cosmic Ray Data Centre (KCDC) [15]. The cosmic ray events in the datasets are reconstructed from the raw detector readings by the iterative algorithm using the Kascade Reconstruction for ExTensive Airshowers program (KRETA). The reconstruction gives the following parameters (features) for each cosmic ray event: shower core position (x, y), zenith angle (θ), azimuth angle (ϕ), number of electrons (N_e), number of muons (N_μ) and shower age parameter (s). The parameters N_e, N_μ and s are determined from the fit of the modified NKG LDF function [14]:

$$\rho_{e,\mu} = C(s) \cdot N_{e,\mu} \cdot \left(\frac{r}{r_m^{e,\mu}}\right)^{s-\alpha} \cdot \left(1 + \frac{r}{r_m^{e,\mu}}\right)^{s-\beta}; \qquad (2.1)$$
$$C(s) = \frac{\Gamma(\beta - s)}{2\pi (r_m^{e,\mu})^2 \Gamma(s - \alpha + 2)(\alpha + \beta - 2s)};$$

where r is a distance to a shower core, $r_m^e = 89\,\mathrm{m}$ and $r_m^\mu = 420\,\mathrm{m}$ are Moliere radii for electrons and muons respectively, α and β are numerical parameters determined from the Monte-Carlo simulations.

The value of the event energy E is reconstructed from the parameters N_e and N_μ with the following relation [15]:

$$\begin{aligned}\log_{10}(E/\mathrm{eV}) = 1.93499 + 0.25788 \log_{10}(\hat{N}_e) + 0.66704 \log_{10}(\hat{N}_\mu) \\ + 0.07507 \log^2_{10}(\hat{N}_e) + 0.09277 \log^2_{10}(\hat{N}_\mu) - 0.16131 \log_{10}(\hat{N}_e) \log_{10}(\hat{N}_\mu) \qquad (2.2)\end{aligned}$$

where $\hat{N}_e$ and $\hat{N}_\mu$ are obtained from N_e and N_μ by θ-dependent rescaling to account for atmospheric attenuation, see ref. [14] for details. It is important to note that this formula is built using Monte-Carlo events based on high-energy hadronic interaction model QGSJet-II.02 [16] and low-energy hadronic interaction model FLUKA-2002.4 [17]. We computed the energy resolution and bias to be 9.5% and -1% respectively, in terms of the logarithm of the simulated to the reconstructed energies ratio, $\ln(E_{\mathrm{MC}}/E_{\mathrm{rec}})$, for given quality cuts, additional energy cut and composition of the full Monte-Carlo set (see below). The full distribution of the uncertainty of the energy reconstruction is shown in figure 1. In addition to observables attributed to a CR event as a whole, there are values of e/γ and μ energy deposits for separate detectors within the event. There are also arrival timings of the shower front for each detector but we do not use them in the present analysis. A graphical example of a recorded event is given in figure 2.

The efficiency of event trigger and reconstruction depends on the type of primary particle, the full efficiency is reached at $E \gtrsim 10^{15}$ eV [14] for all primaries. We use the quality cuts recommended by KASCADE [7]: $\theta < 18°$, $x^2 + y^2 < 91$ m, $\log_{10} N_e > 4.8$, $\log_{10} N_\mu > 3.6$, and the cut on the shower

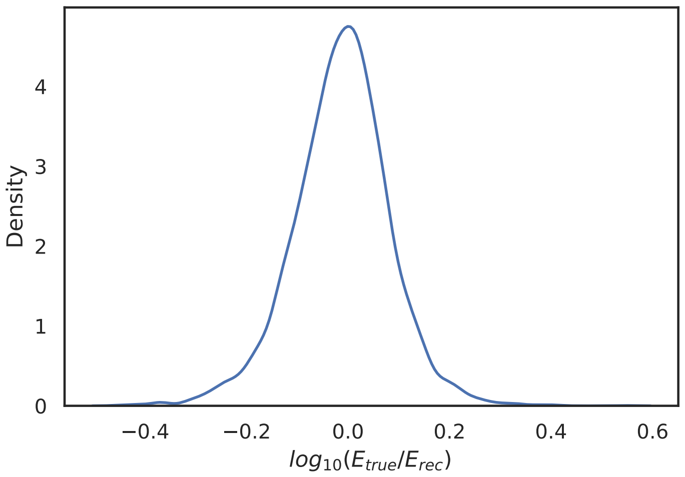


Figure 1. The distribution of the energy reconstruction uncertainty $\log_{10}(E_{\text{true}}/E_{\text{rec}})$ for the KASCADE standard method eq. (2.2).

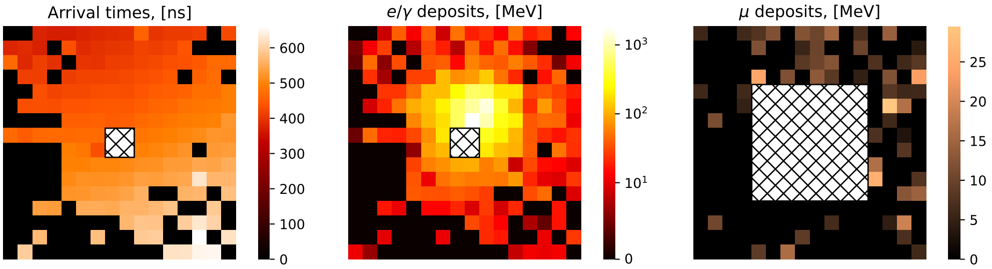


Figure 2. Example of the real event in the dataset. The matrices of arrival times, e/γ, and μ deposits are shown. Reconstructed features of the event are: $\log_{10}(E/\text{eV}) = 15.45$, $\theta = 19.37°$. Note, that KASCADE does not have detector stations in the central 2×2 part for arrival times and e/γ deposits, and in the central 8×8 part for μ deposits. These areas are represented in the figure with a strikethrough area.

age set by KCDC: $0.2 < s < 1.48$ [14], that is tighter than the original KASCADE cut ($0.2 < s < 2.1$). The efficiency of these cuts also depends on a primary type, it reaches 99% at $E > 10^{15.15}$ eV for protons and at $E > 10^{15.33}$ eV for iron nuclei. We also introduce an additional cut on the event energy $\log_{10}(E/\text{eV}) > 15.15$. It is based on our study of confusion matrix and unfolding procedure stability. This additional cut is applied to the real data sets and to Monte-Carlo test sets, but not to Monte-Carlo training sets (see discussion in a section 4). The total number of events in the data set after all cuts is $\sim 3.5 \cdot 10^6$. Also, the set with a looser zenith angle cut $\theta < 30°$ is studied separately, it contains $\sim 8.6 \cdot 10^6$ events in total. In this zenith angle range, the experiment reaches full efficiency of events detection at $E > 10^{15.35}$ eV. We divide the experimental data set into an unblind part containing 20% of randomly selected events and a blind part containing the remaining 80% of events. The unblind part is used for the consistency tests of the mass composition reconstruction methods developed in this study and for the estimation of their realistic uncertainties. The blind part of the data is not analyzed in this paper and is left for the full-scale mass composition study.

The KCDC service provides CORSIKA [18] simulations with events generated for five individual mass groups: p, He, C, Si and Fe in energy range $10^{14} \leq E \leq 3 \cdot 10^{18}$ eV and zenith angle range

$0 \le \theta \le 42°$. Similar sets were produced for several hadronic interaction models, namely: QGSJetII-02 [16] + FLUKA [17] (ver. 2002.4), that were used in the latest original KASCADE mass composition analysis [4] ($\sim 3.7 \cdot 10^4$ events after quality cuts and additional energy cut); QGSJetII-04 [19] + FLUKA (ver. 2012.2.14_32) ($\sim 1.3 \cdot 10^5$ events); EPOS-LHC [20] + FLUKA (ver. 2011.2b.4_32) ($\sim 6.7 \cdot 10^4$ events) and Sibyll 2.3c [21] + FLUKA (ver. 2011.2c.3_64) ($\sim 6.8 \cdot 10^4$ events). As the available amount of Monte-Carlo events is much less than the amount of data events we are taking into account statistical uncertainties of the Monte-Carlo in our results. For each hadronic model, the number of protons and light nuclei events in the MC sets exceeds the number of heavy nuclei but the ratio is smaller than 2. For instance for the set of QGSJetII-04 hadronic model the proportion for p:He:C:Si:Fe is 0.28:0.22:0.19:0.16:0.15. For other hadronic models, the numbers are quite similar. The energy spectrum was set to E^{-2} in CORSIKA simulations and corrected to $E^{-2.7}$ in later stages of analysis. The resulting Monte-Carlo sets were reconstructed using the same code and the actual detector response as for the real data and contained the same observables.

3 Machine learning methods

In the paper, we use a set of different machine learning (ML) methods for event-by-event mass group classification. We are starting with Random Forest (RF) — the classical ML approach, which is our baseline. It takes as input only the reconstructed event features, but not the detectors deposits. Then we switch to using neural networks (NN) to incorporate more data in the analysis. All NN models take as input both event features and deposits from e/γ and μ detectors. In particular, we build a convolutional neural network (CNN), a simple multi-layer perceptron (MLP), and an EfficientNet. All the models were trained with three MC datasets corresponding to three hadronic interaction models: QGSJet-II.04, EPOS-LHC and Sibyll 2.3c separately. The set of the QGSJet-II.02 model is also used separately for direct comparison with the original KASCADE mass composition analysis. These datasets were divided into train, validation, and test subsets. Throughout the study, we use train and validation subsets for training models (the validation set is used for early stopping to avoid overfitting and hyperparameter tuning of the models). All the metrics are evaluated on test subsets of the corresponding MC datasets.

3.1 Random forest

It was chosen to use Random Forest [22] as a baseline solution for the classification of individual events for five mass groups. The following reconstructed features of the air shower were used as input parameters: energy, shower core coordinates, arrival direction, muon and electron numbers, and shower age. We trained two models: classifier and regressor. The classifier predicts the particle type directly while the regressor predicts the mass of the particle, from which we determine the particle type. The hyperparameters of both models were optimized using the Grid Search algorithm, the details are given in appendix A. Both RF models were implemented and trained using scikit-learn package [23].

3.2 Multi-layer perceptron

The simplest neural network we use is a feed-forward multilayer perceptron consisting of two hidden layers, each followed by batch normalization, ELU activation, and dropout (with a rate of 0.15). We use integral signals from the e/γ and μ detector stations and reconstructed zenith and azimuth angles as input for the model. The model was trained with the Adam algorithm [24] using a batch size of

1024 and a variable learning rate (starting from 0.001, then multiplying it by 0.5 when validation loss plateaus) until validation loss stops improving. This quite primitive architecture shows quite good performance, nevertheless (see next section). Detailed architecture of the model is given in appendix A.

3.3 Convolutional neural network

The second NN we use is a convolutional neural network. This approach allows us to include the deposits from the detectors in the analysis. We developed and trained a simple convolutional neural network, inspired by the LeNet-5 [25] model, to classify individual events for five mass groups. The integral signals from the e/γ and μ detector stations for each event are fed into the model. Additionally, we append the high-level features $\log_{10} N_e$, $\log_{10} N_\mu$, θ, s as an input to the first dense layer. We need to note that the CNN model is trained with Monte-Carlo sets with the looser zenith angle cut: $\theta < 30^\circ$. The details of this ML model can be found in appendix A.

3.4 EfficientNet

As an additional benchmark, we trained an EfficientNetV2-S [26] model that belongs to a family of new, high-efficiency convolutional neural networks. EfficientNetV2 uses some specific hardware optimizations, such as the use of Fused-MBConv convolutions instead of depthwise convolutions in early layers, that are combined with the heavy use of neural architecture search. The model takes integral signals from the e/γ and μ detector stations as a main input for convolutional layers, and reconstructed zenith and azimuth angles as an additional input for the bottleneck layer of the model. The model was trained with the Adam algorithm using a batch size of 128 and a variable learning rate (starting from 0.001, then multiplying it by 0.5 when validation loss plateaus) until validation loss stops improving. While EfficientNetV2-S is considerably larger than our CNN, it achieves almost identical performance, which suggests that more complex models may not lead to significant improvements in performance when dealing with this particular problem. Furthermore, the larger model size could lead to increased computational costs and longer training times, making it less practical for our purposes. Therefore, we do not discuss the results of this ML model further.

4 Performance of the methods

In this section, we present the basic results of cosmic ray events primary mass classification using our machine learning methods. We compare their quality and reconstruct Monte-Carlo spectra to check for correctness. All results here and further are computed on the Monte-Carlo test sets or the unblind data set using the quality cuts described in the section 2 plus the additional cut $\log_{10}(E/\text{eV}) > 15.15$ that ensures the full efficiency of detection for all primaries. However, we do not use this cut for the MC training sets: in general for machine learning the total amount of training data is crucial, while their possible non-uniformities are of secondary importance.

4.1 Confusion matrices

The simplest way to estimate the quality of the trained ML models is to compute confusion matrices on the test datasets for each model. The confusion matrix shows the proportion of the particle of each type that the given model classifies as this type and all other types. A more diagonal matrix means more precise classification. Also, the transposed confusion matrix is used as a response matrix for the unfolding procedure described later. The confusion matrices for the ML models used in this paper are shown in figure 3–5. One can see that the matrices are diagonal in most cases. The diagonal elements have much better accuracy than the random guessing (0.2 for 5 components classification). The best performance is shown by CNN.

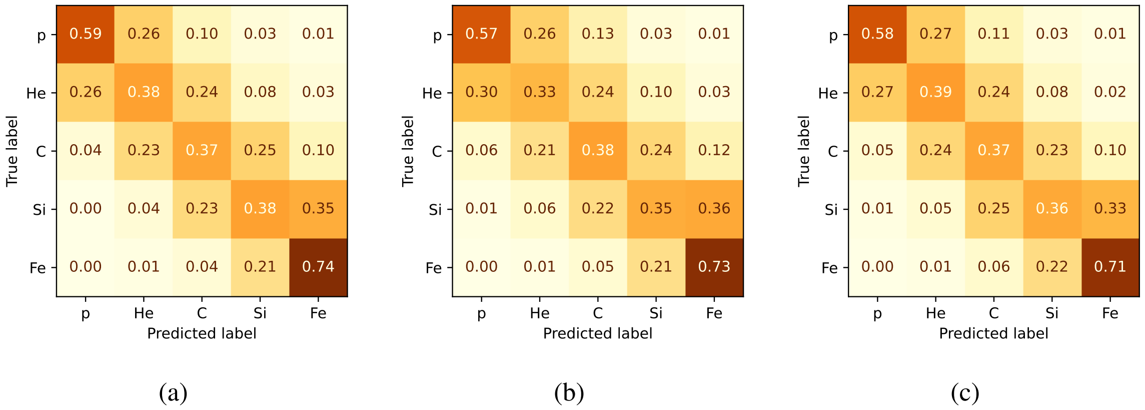


Figure 3. Confusion matrices for RF Classifier models trained and tested with three different hadronic interaction models (a) QGSJet-II.04, (b) EPOS-LHC and (c) Sibyll 2.3c.

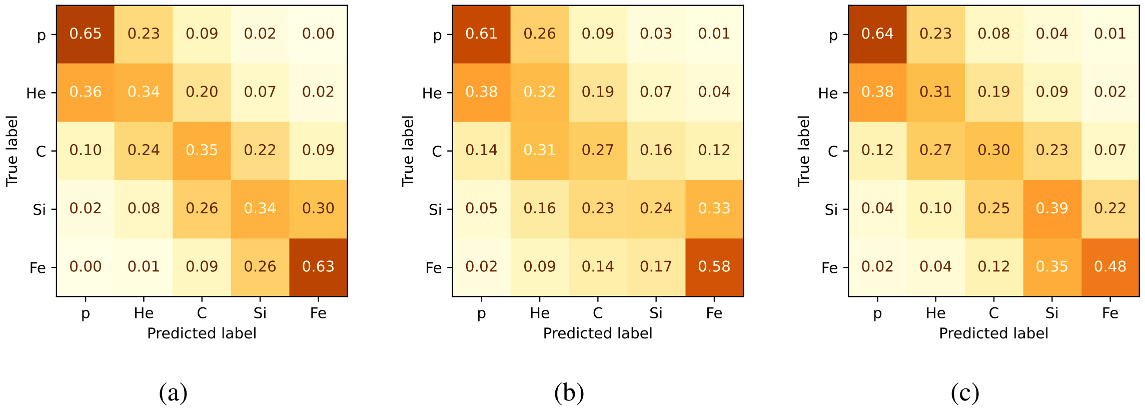


Figure 4. Confusion matrices for MLP models trained and tested with three different hadronic interaction models (a) QGSJet-II.04, (b) EPOS-LHC and (c) Sibyll 2.3c.

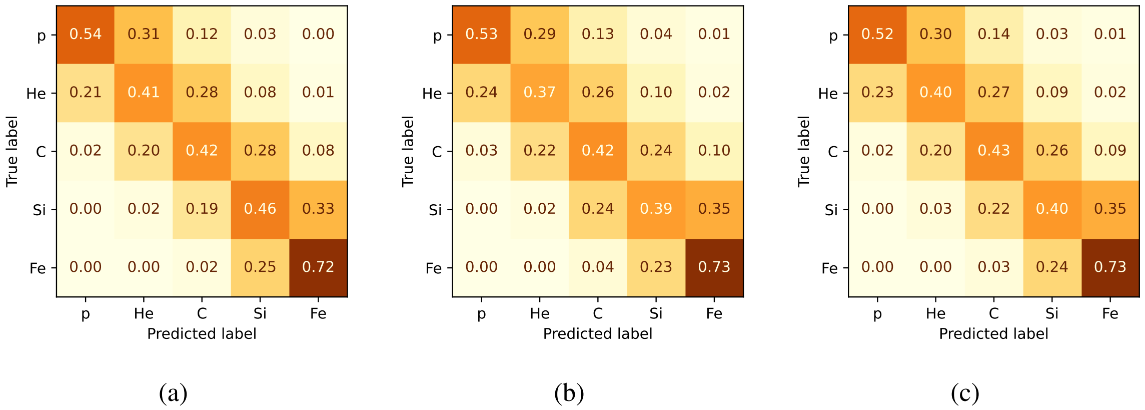


Figure 5. Confusion matrices for CNN models trained and tested with three different hadronic interaction models (a) QGSJet-II.04, (b) EPOS-LHC and (c) Sibyll 2.3c.

4.2 Reconstruction of mass components spectra

In figure 6 we show the discrepancy between the true spectra and the reconstructed spectra of separate mass components for the QGSJet-II.04 model and our reconstructions with RF, CNN, and MLP. One can see that while all ML models have roughly similar accuracy in energy spectra reconstruction in general, the accuracy of the CNN model is the least energy-dependent. The general underestimation of proton flux and overestimation of iron flux are subject to further correction with the unfolding procedure (see section 6).

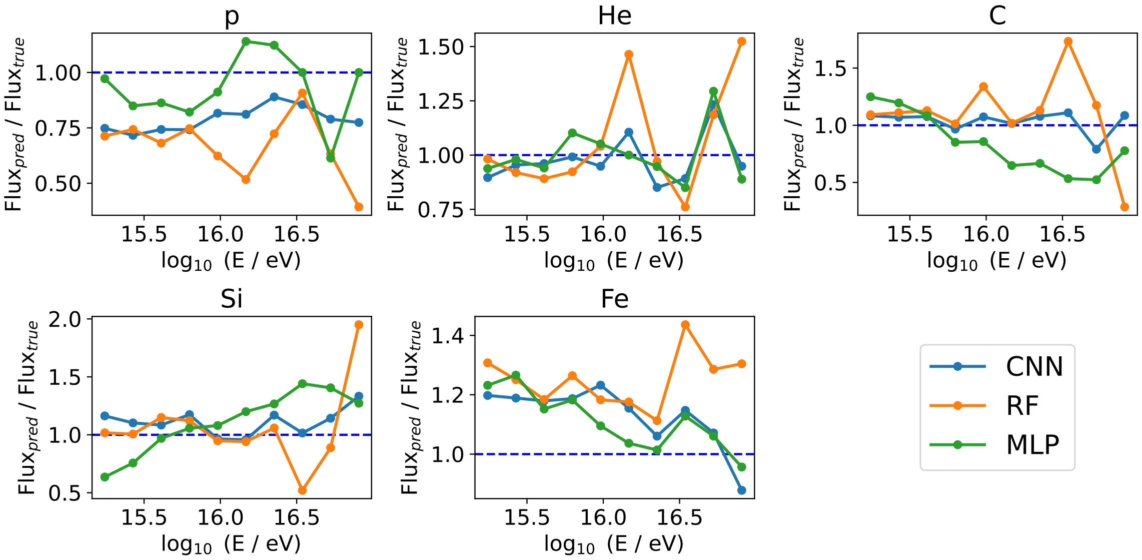


Figure 6. Results of the deviation of predicted spectra from true spectra in Monte-Carlo test sets (QGSJet-II.04) for the CNN, RF and MLP models.

4.3 Mass components reconstruction in mixtures

Another way to estimate the quality of the model predictions is to calculate the mean absolute error (MAE) between true component fractions in ensembles and the predicted fractions [9]. The smaller the MAE for a particular component, the better the ML model does its job. This method allows us to estimate how the accuracy of the reconstruction of particular mass components depends on the type of mixture under analysis.

To perform this estimation we need to create a set of various mass components mixtures. We produce 2 000 random ensembles of 5 000 events each using the test set. There are two obvious methods to make such an ensemble. In the first method, the fractions of components in a given ensemble are distributed uniformly so we call this approach “uniform”. It has the disadvantage that extreme mixtures (for example pure protons) are very rare, while the sets with approximately equal fractions are much more common. Therefore, in addition, we use the second method of creating mixtures, where the proportions of mass components are distributed over a grid with a step of 0.1 from 0 to 1. This approach is called “grid”. The distributions of the absolute errors between true and predicted fractions in these two approaches are shown in figure 7. One can see that the accuracy of the ML reconstruction is much better for the uniform sets than the accuracy averaged over all possible sets generated by the “grid” approach.

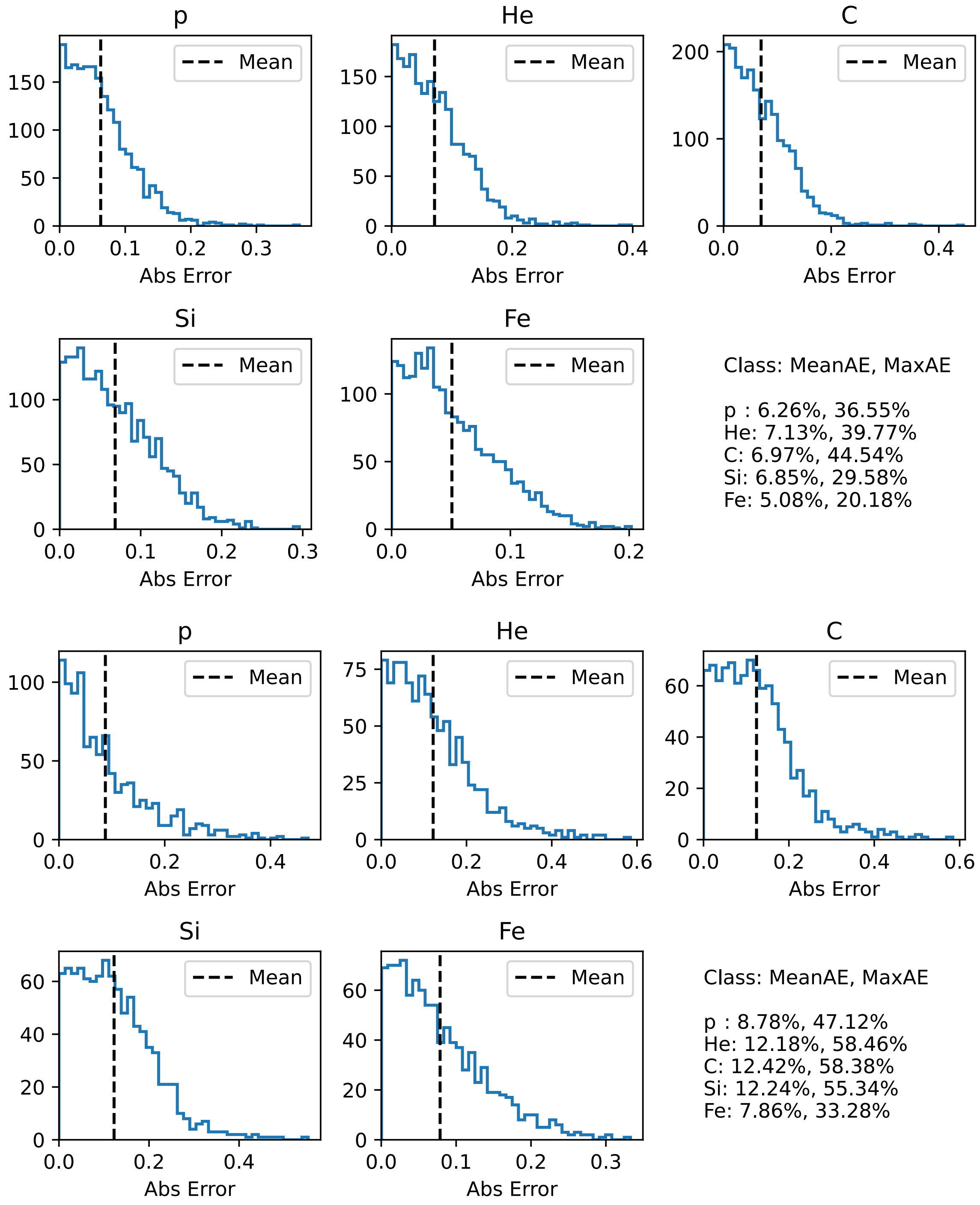


Figure 7. Distributions of the absolute errors between true and predicted fractions for the CNN model trained and tested with QGSJet-II.04 hadronic interaction model (upper figure) in the "uniform" mixtures approach, (lower figure) in the "grid" mixtures approach. Black lines in the figures represent mean absolute error (MeanAE) for each mass component. MaxAE means maximum absolute error for each mass component.

The "grid" approach covers more possible mixtures and gives a more strict estimate of the deviation. Therefore, we use this approach to compare our models. The results for different models in

Table 1. Mean absolute error between true and reconstructed fractions of elements for three different models and mass groups calculated with "grid" approach using QGSJet-II.04 hadronic interaction model. The best result for each mass component is highlighted in bold.

	p	*He*	*C*	*Si*	*Fe*
RF	**0.083**	0.123	0.127	0.123	0.085
CNN	0.088	**0.122**	**0.124**	**0.122**	**0.079**
MLP	0.089	0.128	0.136	0.131	0.089

the "grid" approach are shown in table 1. We see that all the models have a similar quality, but the CNN has slightly better performance. The similarity in the quality of the three ML models, that were built using completely different ML techniques, may reflect the fact that we extracted the maximum available information about mass composition from the given input data.

5 Tests and uncertainties

To ensure the correctness of the developed ML models we perform several consistency tests and estimate several uncertainties. Namely, we study the behavior of the models for data with missed detector stations. We perform an ablation study to estimate the importance of various input parameters. We also estimate the dependence of the ML model performance on the zenith angle and the energy of the primary particle. In this section, we consider the CNN model as our baseline model, while several tests performed for other ML models can be found in appendix B.

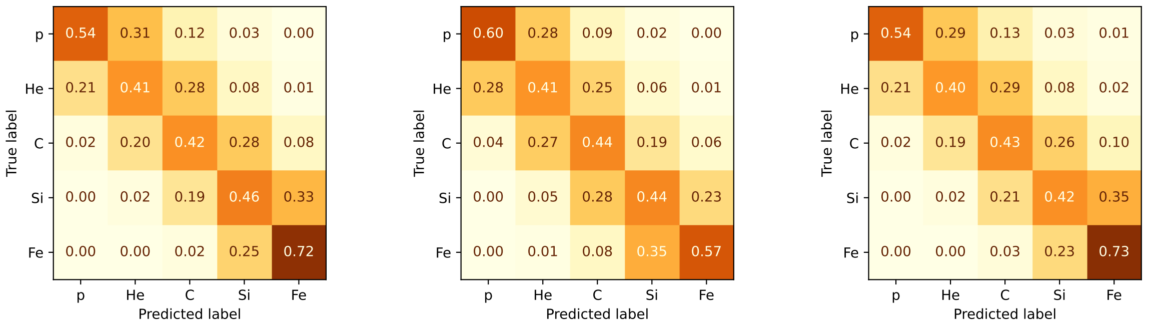
5.1 Ablation study

To estimate the impact of the different input features on the ML method performance, we make an ablation study. We remove input features sequentially, train the model again, and compare the resulting confusion matrices. This procedure helps us to understand the impact of different features on the result. It also tests that the model is focused on physical features and dependencies rather than simulation artifacts.

The results of the ablation study are shown in figure 8. We compare the confusion matrices of the default CNN trained with detectors deposits and reconstructed features, the CNN trained with deposits only and the CNN trained with reconstructed features only. One can see that the matrix for reconstructed features only looks slightly better than the matrix for deposits only. It was found that this difference comes mostly from the zenith angle (θ) In general, the other features complement each other so that the default CNN trained on deposits and reconstructed features shows slightly better performance with the most diagonal confusion matrix.

5.2 Uncertainty of missed detectors in data

The experimental data contains events with non-working detector stations, while in KCDC Monte-Carlo this feature is not simulated. As our ML models are trained with the Monte-Carlo simulations, their performance with the data can be unstable due to this difference. We will consider the possible effect of this difference as a systematic error. To estimate it, we prepared a "corrupted" data set from the test



(a) (b) (c)

Figure 8. Ablation study: confusion matrices (a) of the default CNN, (b) of the CNN trained only with e/γ and μ detectors deposits, (c) of the CNN trained only with reconstructed features ($\log_{10} N_e$, $\log_{10} N_\mu$, s, θ). QGSJet-II.04 hadronic model is used for both training and test datasets.

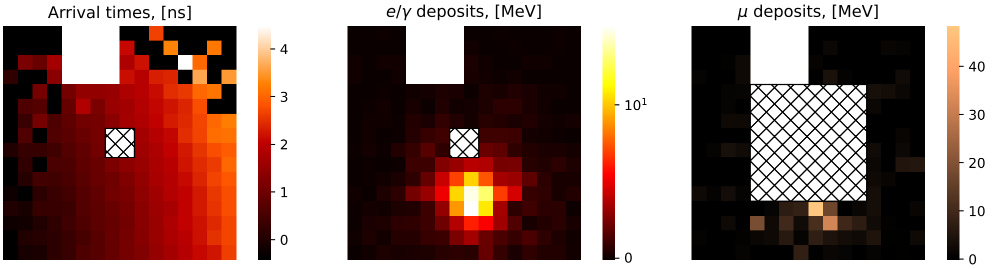


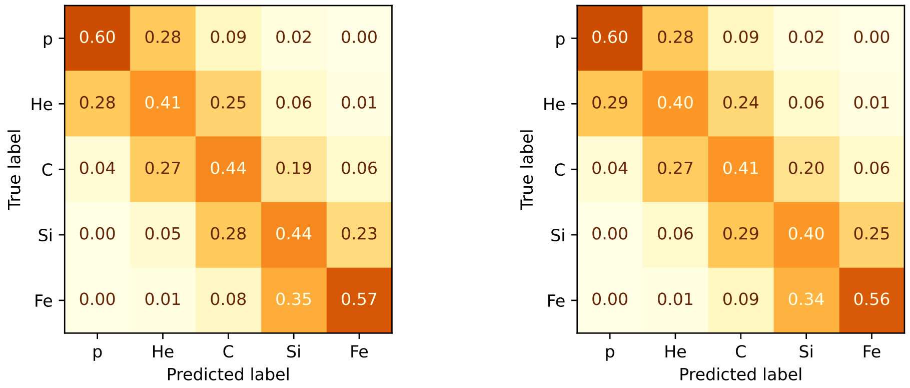
Figure 9. Example of the simulated event in the "corrupted" dataset. The white areas in the event represent missing detectors. The matrices of arrival times, e/γ, and μ deposits are shown. Reconstructed features of the event: $\log_{10}(E/\mathrm{eV}) = 15.61$, $\theta = 15.68°$.

Monte-Carlo set, resembling the experimental data set with non-working stations. Each event in this set has a random region of 4×4 non-working stations. An example of such an event is shown in figure 9.

In figure 10 we compare the confusion matrices of the CNN model obtained with the original test Monte-Carlo set and the "corrupted" Monte-Carlo set. For this test, we use the CNN model trained with detector deposits only to estimate the impact of this uncertainty conservatively. One can see that the general structure of the confusion matrix is conserved and the average difference of the diagonal values is about 2%. In what follows we consider this effect as a separate contribution to the total systematic uncertainty of our analysis method. As ~ 50% of events in the real data set contain non-working detectors, we take into account the estimated uncertainty with a factor of 0.5.

5.3 Zenith angle dependence

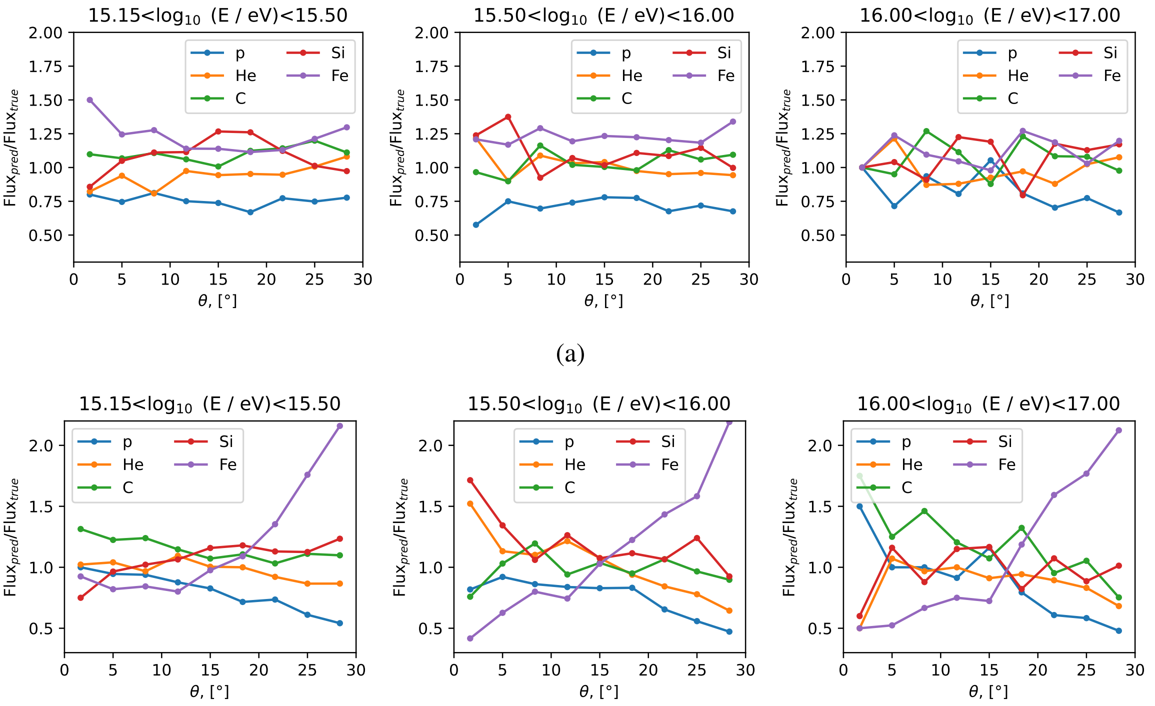
Ideally, the reconstructed mass component spectra should be independent of the zenith angle of the primary particle. In figure 11 we show the dependence of the inaccuracy of the reconstructed flux for separate mass components on the zenith angle in three different energy ranges. We consider two variants of the CNN model: a default one and the one trained without zenith angle input. One can see that for CNN without θ input the error in *Fe* component reconstruction grows dramatically with the angle. While there is no clear θ-dependence in the results of the default CNN. We conclude that the addition of the zenith angle to the inputs ensures the stable behavior of the CNN model results.



(a)

(b)

Figure 10. Confusion matrices of the CNN: (a) trained with e/γ, μ deposits only, (b) the same but with "corrupted" data. Monte-Carlo of QGSJet-II.04 hadronic interaction model is used for training and testing.

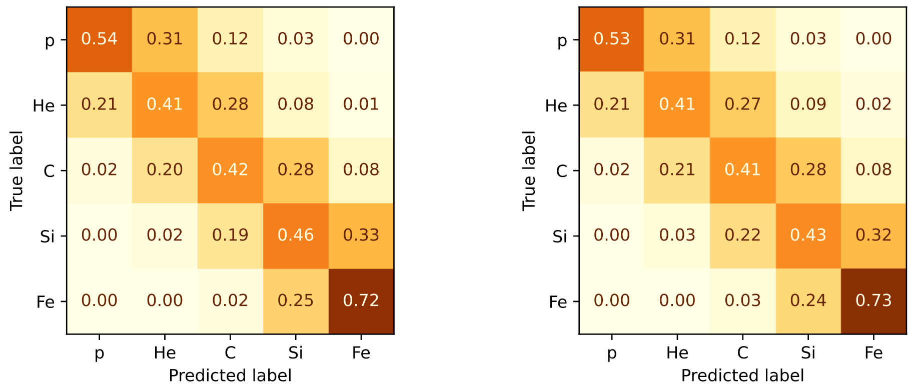


(a)

(b)

Figure 11. The ratio of the predicted flux and the true flux as a function of the zenith angle for five mass groups in three different energy ranges. The figures are (a) for the default CNN (that inputs θ, $\log N_e$, $\log N_\mu$, s and detector deposits), (b) for the modified CNN without θ input parameter.

We also compare the confusion matrices for two zenith angle ranges (below 18° and below 30°). The results are shown in figure 12. One can see that the matrices are almost similar, which shows us the validity of the ML model for a wide θ range. The same conclusion can be derived from figure 13, where we show the dependence of the reconstructed flux inaccuracy for separate mass components on the reconstructed energy in the same two zenith angle ranges.



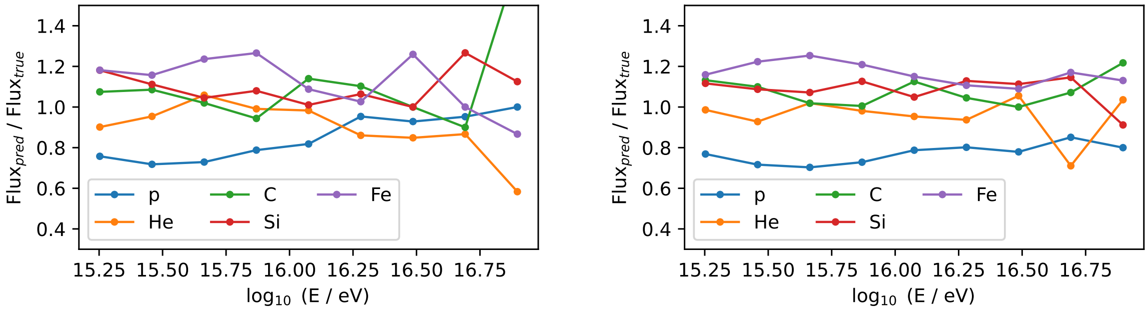
True label \ Predicted label	p	He	C	Si	Fe
p	0.54	0.31	0.12	0.03	0.00
He	0.21	0.41	0.28	0.08	0.01
C	0.02	0.20	0.42	0.28	0.08
Si	0.00	0.02	0.19	0.46	0.33
Fe	0.00	0.00	0.02	0.25	0.72

(a)

True label \ Predicted label	p	He	C	Si	Fe
p	0.53	0.31	0.12	0.03	0.00
He	0.21	0.41	0.27	0.09	0.02
C	0.02	0.21	0.41	0.28	0.08
Si	0.00	0.03	0.22	0.43	0.32
Fe	0.00	0.00	0.03	0.24	0.73

(b)

Figure 12. Confusion matrices of the CNN with the different zenith angle cuts: (a) $\theta < 18°$, (b) $\theta < 30°$. CNN is trained and tested with QGSJet-II.04 hadronic model sets.



(a)

(b)

Figure 13. The ratio of the predicted flux and the true flux as a function of energy for five mass groups for two zenith angle cuts: (a) $\theta < 18°$ and (b) $\theta < 30°$. CNN is trained and tested with QGSJet-II.04 hadronic model sets.

5.4 Energy dependence

In this section, we compare the quality of the CNN predictions in the different energy bins. This is important for the unfolding procedure: if the predictions are stable in a certain energy range the response matrix can be averaged over this range to benefit from the larger statistics. The CNN confusion matrices for different energy bins are shown in figure 14. One can see that the accuracy of the CNN predictions is growing with energy. However we need to note that the behavior of the confusion matrix does not reflect the full picture: because of the steeply falling real spectrum, uncertainties at the higher energies are dominated by statistical fluctuations.

To figure out the evolution of the total accuracy of the mass components reconstruction with energy we look for the behavior of the diagonal elements of the confusion matrix, as they have the largest impact on the result. Their dependence on energy is shown in figure 15. We split the full energy range under consideration into 4 subranges for each of which we separately compute the confusion matrix which is used for the further unfolding procedure (see section 6). These ranges are: $15.15 < \log_{10}(E/\mathrm{eV}) < 15.25$, $15.25 < \log_{10}(E/\mathrm{eV}) < 15.5$, $15.5 < \log_{10}(E/\mathrm{eV}) < 16$,

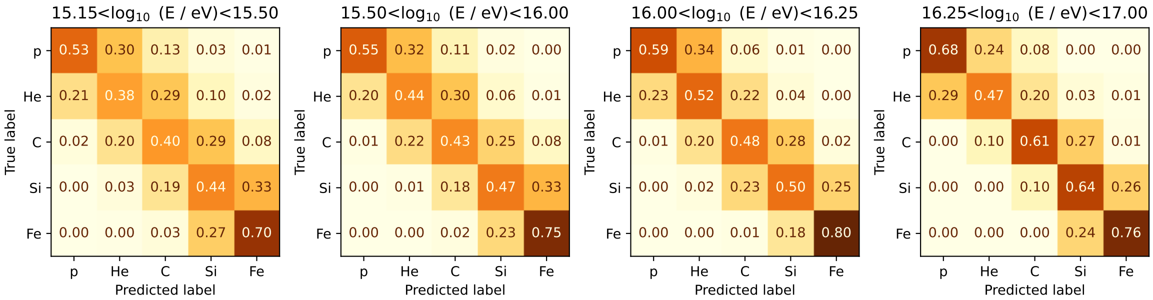


Figure 14. Confusion matrices of the CNN for different reconstructed energy bins. The neural network is trained and tested using the QGSJet-II.04 hadronic interaction model.

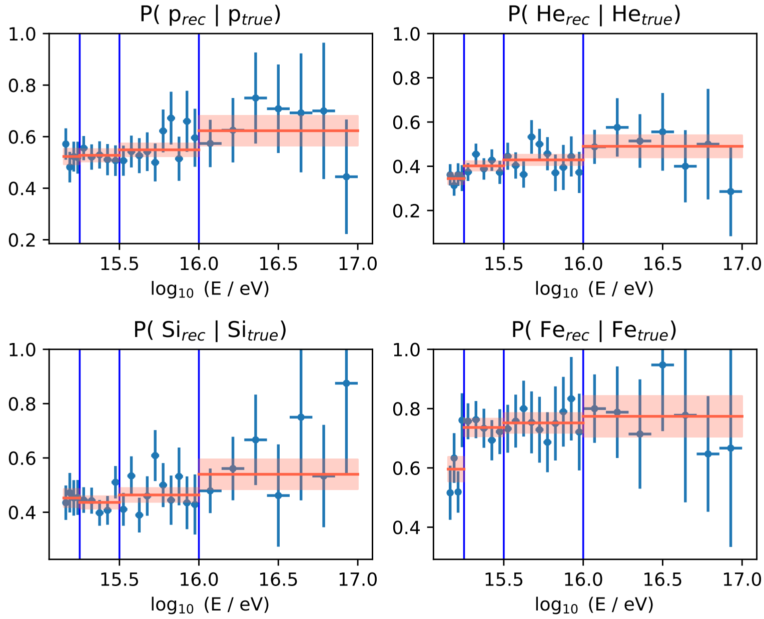
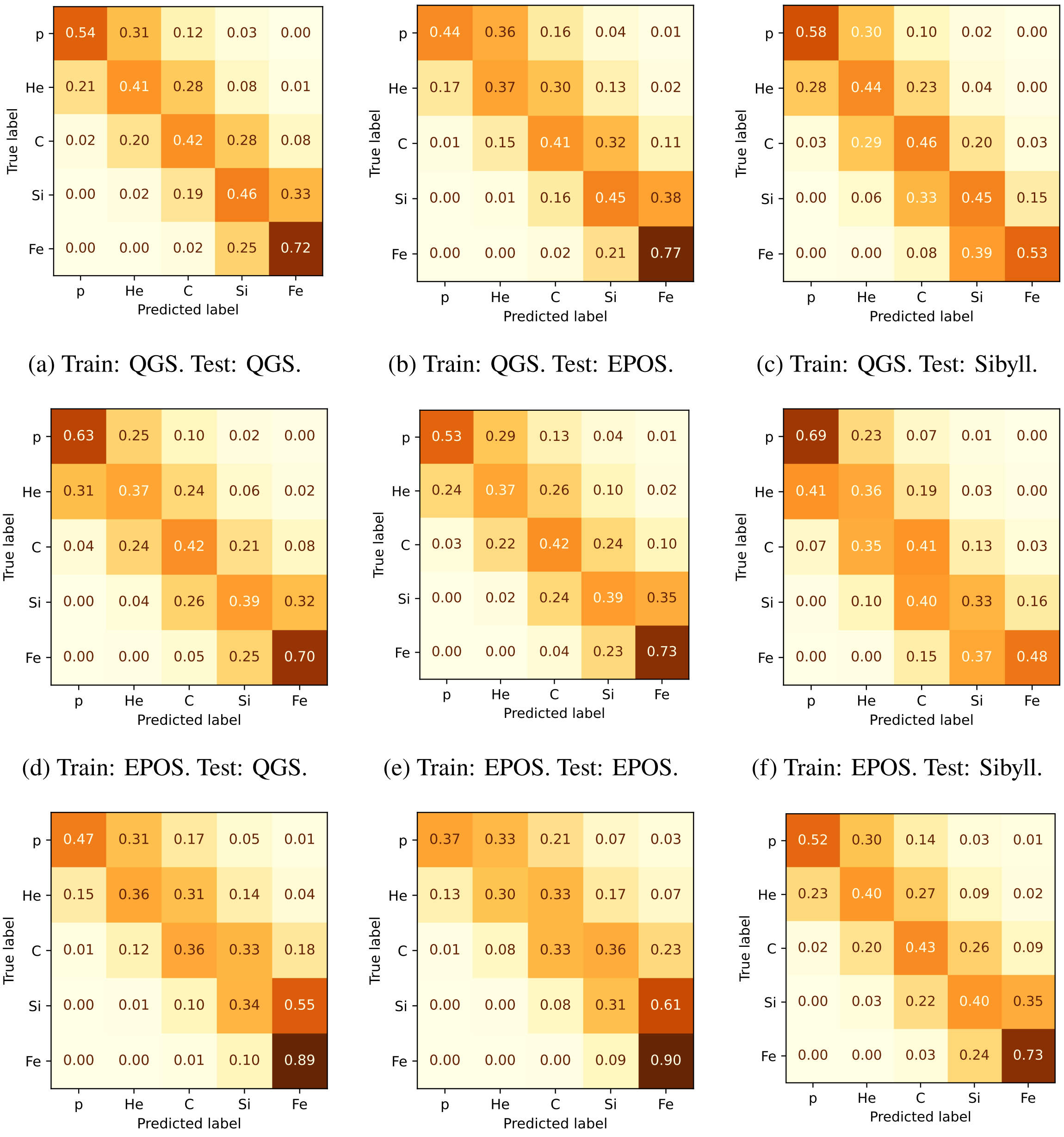


Figure 15. The dependence of the diagonal elements of the confusion matrix on the reconstructed energy for the CNN trained and tested using the QGSJet-II.04 hadronic interaction model. Blue vertical lines depict the ranges where the average confusion matrices are computed for the further unfolding procedure (see text). Red horizontal lines show the average value of a given matrix element in a given energy range together with its uncertainty.

$\log_{10}(E/\mathrm{eV}) > 16$. The chosen splitting has two goals. First, the matrix should be stable in the given energy range to avoid unnecessary uncertainties in the unfolding. Second, the overall accuracy of the mass component reconstruction, including the uncertainty from the limited Monte-Carlo statistics, should grow with energy.

5.5 Cross-hadronic models reconstruction

Here we consider the effect of the different hadronic interaction models. We trained and tested the CNN with the three modern hadronic interaction models: QGSJet-II.04, EPOS-LHC and Sibyll 2.3c. By comparing the predictions of the CNN trained on one interaction model and tested on another one, we can estimate how much our ML model is affected by their difference. The resulting confusion matrices are shown in figure 16. One can see that the predictions between QGSJet-II.04 and EPOS-LHC are generally similar and that the largest discrepancy is between Sibyll 2.3c and other models, where the notable shifts with respect to the main diagonal appear. At the same time, the general structure of the cross-hadronic model confusion matrices is smooth, without any sharp artifacts that give additional credibility to the ML model. Further in this study, we are not discussing the impact of the cross-hadronic model systematic on the resulting reconstruction of mass components spectra, leaving this issue for the full-scale mass composition analysis.



(a) Train: QGS. Test: QGS.

(b) Train: QGS. Test: EPOS.

(c) Train: QGS. Test: Sibyll.

(d) Train: EPOS. Test: QGS.

(e) Train: EPOS. Test: EPOS.

(f) Train: EPOS. Test: Sibyll.

(g) Train: Sibyll. Test: QGS.

(h) Train: Sibyll. Test: EPOS.

(i) Train: Sibyll. Test: Sibyll.

Figure 16. Cross-hadronic model confusion matrices for the CNN models trained and tested separately with three different hadronic interaction models: QGSJet-II.04, EPOS-LHC and Sibyll 2.3c which are marked in the figure as QGS, EPOS, and Sibyll, respectively.

5.6 Test with the unblind set of the real data

To test our ML methods for possible hidden systematics that do not appear in Monte-Carlo, we use a so-called "semi-blind" data analysis in this study. As it was mentioned, we divide the experimental data into blind and unblind parts in a ratio of 80%:20%. Here we show the test of our ML reconstruction procedure with the unblind data set. In figure 17 we show the spectra for the separate mass components of the unblind part of the KASCADE data according to the reconstruction of the CNN model trained with QGSJet-II.04 Monte-Carlo. One can see that the spectra of the separate mass groups are smooth at lower energies, where the statistical errors are small enough. The absence of apparent artifacts in these spectra increases the credibility of our mass composition analysis method. We should note that for a correct comparison between these spectra and the original KASCADE spectra [4] the unfolding effects should be taken into account and the hadronic model should be the same. In the next section, we build the CNN model trained with the QGSJet-II.02 model, perform the unfolding and make such a comparison.

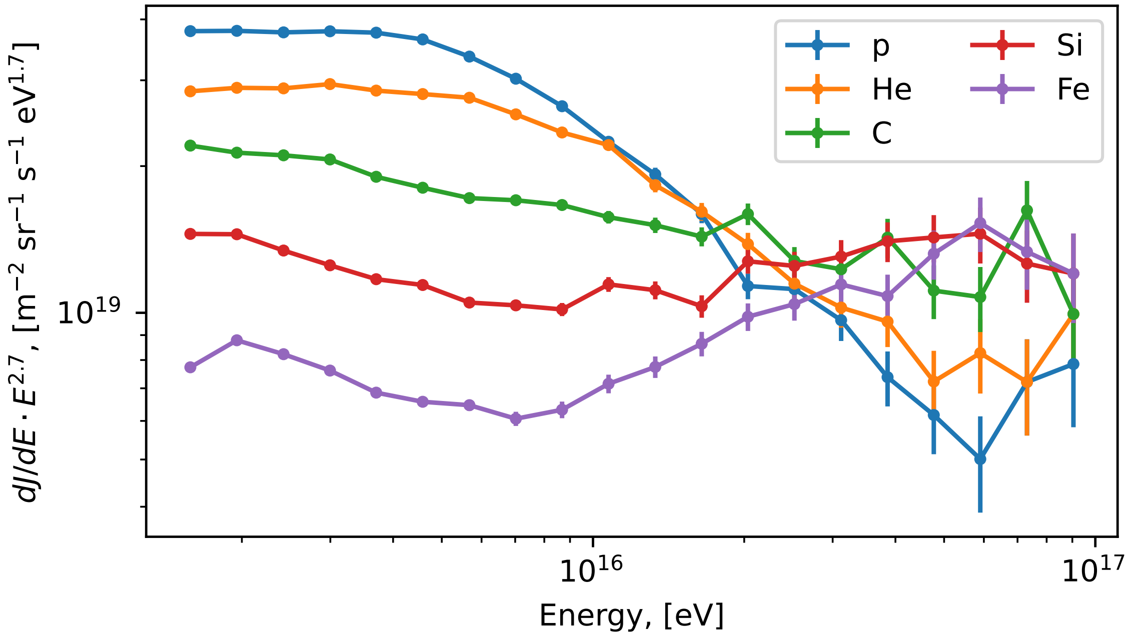


Figure 17. Mass composition (folded) spectra on unblind experimental data for the CNN model trained using QGSJet-II.04 hadronic interaction model.

6 Unfolding and results

The accuracy of our methods of mass component reconstruction is not ideal. The same is true for the standard method of the primary energy reconstruction that we use in this study. At the same time, the uncertainties of the reconstructions are characterized by the confusion matrix of the given method for the given Monte-Carlo set. Using this knowledge we can improve the reconstruction for observables that are averaged over ensembles of events, for instance for spectra of separate mass components. This procedure of the reconstruction improvement is known as unfolding. In this section, we describe the method we use to unfold the reconstructed mass components spectra. We also compute the uncertainties that affect the final result and compare it to the original KASCADE reconstruction.

6.1 Unfolding procedure

We perform the unfolding for primary mass classification and for primary energy reconstruction separately, one after one. We start with the unfolding of a primary particle type. Let's denote the ratio of the number of events in i-th bin of some quantity to all events in a set as a probability P_i. For a single event, we have a standard formula of conditional probability:

$$P(A_n^{\mathrm{rec}}|E_m^{\mathrm{rec}}) = \sum_j P(A_n^{\mathrm{rec}}|A_j^{\mathrm{tr}}, E_m^{\mathrm{rec}}) P(A_j^{\mathrm{tr}}|E_m^{\mathrm{rec}}) \, ; \tag{6.1}$$

where $A_i^{\mathrm{rec/tr}}$ denotes i-th class of the primary particle (reconstructed or true) and E_m^{rec} denotes m-th bin of the reconstructed energy. The same relation can be presented in another notation:

$$Y_n^m = \sum_j R_{nj}^m X_j^m \, ; \tag{6.2}$$

where Y_n^m is a number of reconstructed particles of class n in the reconstructed energy bin m; R_{nj}^m is a so-called response matrix for the reconstructed energy bin m which is equal to the respective transposed confusion matrix and X_j^m is a number of true particles of class j in the reconstructed energy bin m that we want to derive. This is a standard unfolding problem that can be solved in different ways. Here we use a Bayesian iterative approach [27] with python package pyunfold [28]. Stopping criteria $\Delta\chi^2$ are: 0.01 for particle type unfolding and 0.1 for energy unfolding. The uncertainties in response matrices were considered as Poissonian ones.

Then we continue with the energy unfolding. This can be described by the following relation:

$$P(E_m^{\mathrm{rec}}) = \sum_i P(E_m^{\mathrm{rec}}|E_i^{\mathrm{tr}}) P(E_i^{\mathrm{tr}}) \, ; \tag{6.3}$$

where $E_m^{\mathrm{rec/tr}}$ is m-th bin of reconstructed or true energy. We also use the Bayesian iterative approach here with an initial suggestion of $P(E_i^{\mathrm{tr}})$ taken from the Monte-Carlo data.

Finally, we combine these two unfoldings. The probability we need to know, $P(A_i^{\mathrm{tr}}, E_j^{\mathrm{tr}})$, can be written as:

$$P(A_i^{\mathrm{tr}}, E_j^{\mathrm{tr}}) = P(A_i^{\mathrm{tr}}|E_j^{\mathrm{tr}}) P(E_j^{\mathrm{tr}}) \to P(A_i^{\mathrm{tr}}|E_j^{\mathrm{rec}}) P(E_j^{\mathrm{tr}}) \, ; \tag{6.4}$$

here the arrow denotes that we substitute $P(A_i^{\mathrm{tr}}|E_j^{\mathrm{tr}})$ with $P(A_i^{\mathrm{tr}}|E_j^{\mathrm{rec}})$ as this is the quantity we derive from the particle type unfolding. The impact of this substitution is considered as one of the systematics uncertainties of the method, it is estimated in the next section using the Monte-Carlo data. Note, that the "confusion matrices" for the energy unfolding (of the energy reconstruction with eq. (2.2)) are computed separately in each energy bin, while that for the particle type unfolding are averaged over energy intervals defined in the section 6.1.

6.2 Estimation of the uncertainties

Here we estimate the total uncertainty of our reconstruction of mass components spectra, in particular, we take into account statistical uncertainty associated with a limited number of experimental events and a number of systematic uncertainties:

- uncertainty due to missing detectors, as described in section 5.2;
- uncertainty of the response matrix related to a limited amount of Monte-Carlo data;
- uncertainty related to a bias due to the unfolding regularization;

- uncertainty of the energy unfolding due to a different energy resolution for different mass components

- uncertainty of the full unfolding procedure due to the substitution in eq. (6.4).

- uncertainty related to a spectrum index in the Monte-Carlo

- uncertainty related to non-full experiment efficiency at low energies

The total systematic uncertainty is calculated as a sum in quadrature of all separate systematic uncertainties. Since the latest KASCADE results are based on the QGSJet-II.02 hadronic interaction model, we are using the same model to make a valid comparison. We do not include the uncertainties related to the hadronic models in the present analysis, leaving it for a further full-scale KASCADE composition study. All the systematic uncertainties described in this section are calculated in a way that is appropriate for both Monte-Carlo sets and experimental data sets, where the true values of parameters are unknown. The statistical uncertainty of the unfolded mass components spectra is calculated by propagation of the standard statistical uncertainty with the pyunfold package.

The uncertainty due to missing detectors is estimated as follows. First, we evaluate the response matrices for the default Monte-Carlo without missing detectors and the Monte-Carlo imitating missing detectors (see section 5.2) and find their difference. Then, we use this difference as the uncertainty of the response matrix and propagate it into the uncertainty of the resulting mass component spectra. The resulting uncertainty is taken with the weight of $1/2$, the fraction of events with missing detectors in the experimental data. In terms of a flux in a given energy bin, the relative value of this uncertainty is 1–20% on average for all mass components, depending on the primary particle energy. The impact of the missing detectors on the energy unfolding is evaluated using the data from the unblind set. We construct two sets: one with all detectors working in all events, and another one with a number of detectors non-working in all events. The difference in the energy unfolded all-particle spectra of these two sets is interpreted as the uncertainty due to the missed detector. This uncertainty is ~ 4% on average for all energies.

There is a statistical uncertainty in the response matrix coming from a limited number of Monte-Carlo events. We propagate this into a systematic uncertainty of the unfolded mass components spectra using pyunfold package. The relative value of this uncertainty for particle type unfolding in the resulting spectra is 6–20% on average for all mass components, depending on the energy bin. The same relative uncertainty for energy unfolding is 2–15%.

The unfolding method itself is not ideal, it contains an iterative procedure with predefined stopping parameters, that leads to the appearance of a bias in the unfolded results. This bias appears in both the particle type unfolding and the energy unfolding. Theoretically, we need a very large amount of Monte-Carlo events to estimate this bias. In practice, we do the estimations with the following steps. We name the raw ML predictions of component fractions as (vector) ν. Then we unfold these predictions using our procedure and consider the resulting values, μ, as true ones. Then we fold the results back, just by multiplying them with the confusion matrices, and name the result $\tilde{\nu}$. The discrepancy between ν and $\tilde{\nu}$ describes the non-ideality of our unfolding procedure. To translate this discrepancy into the uncertainty of the reconstructed spectra we produce 100 toy Monte-Carlo spectra, randomized using Poisson PDF with mean $\tilde{\nu}$, and unfold them again using our procedure. We name the obtained fractions: $\tilde{\mu}$. Then the resulting uncertainty is estimated as difference $\langle\tilde{\mu}\rangle - \mu$

in each energy bin. The described procedure has the advantage that the uncertainty estimated for the data set has no extra contributions due to possible correlations of the unfolding bias with other uncertainties in Monte-Carlo. For the particle type unfolding the relative value of this uncertainty in the resulting spectra is 1–65% (1–11% if we do not account for bins where we have outliers, see figure 20) on average for all mass components, depending on the energy bin. For the energy unfolding the respective uncertainty is 0.1–13%, depending on the energy bin.

We also estimated the uncertainty in the energy unfolding associated with different energy resolutions for mass groups. For example, the energy response matrix for *Fe* is better than for *p*. Thus, if the given mixture in a Monte-Carlo set used for energy response matrix computation is different from the one we have in a data set, additional uncertainty will appear. To estimate uncertainty we perform two extra energy unfolding separately, using response matrices for *p* only and for *Fe* only. Then we consider the mix obtained after the particle type unfolding and assign the proton energy unfolding to *p* and *He* components but iron energy unfolding for all heavier components, for simplicity. We compare the resulting spectra with those where energy unfolding was performed with a standard response matrix. This uncertainty has an 8–10% value depending on energy.

The uncertainty from eq. (6.4) was estimated with Monte-Carlo sets. First, we perform both particle type unfolding and energy unfolding, fit the resulting points using χ^2 procedure, and consider the resulting spectra as "true" values. Then we fold these spectra using the energy response matrix derived from the Monte-Carlo. This yields us spectra as it would be reconstructed by our ML methods from the given "true" spectra. Therefore, the ratio of these spectra in a given energy bin yields a sought for uncertainty:

$$1 + \delta = \frac{P(A_i^{\mathrm{tr}}|E_j^{\mathrm{tr}})}{P(A_i^{\mathrm{tr}}|E_j^{\mathrm{rec}})} \tag{6.5}$$

The value of this uncertainty is growing from almost zero at low energies up to 13% at high energies, on average for all mass components. This behavior is explained by the fact that the mass group fractions change dramatically around the primary particle energy 10^{16} eV, where the *Fe* fraction rises and the *p* fraction falls. So in the region of the largest derivative of the fractions, the uncertainties reach maximum values.

We also take into account uncertainty due to possible differences in a spectral index γ between the Monte-Carlo set and the real data set. This uncertainty appears because the response matrix used for energy unfolding depends on γ assumed in the MC set. To estimate this effect we perform the energy unfolding with $\gamma = -3.0$ response matrix and calculate the difference from our basic result with $\gamma = -2.7$. This yields an extra uncertainty of 0.5–7% depending on the energy of the primary particle.

There is an extra uncertainty related to our procedure of spectrum reconstruction. Ideally, the spectra for each mass component should be computed taking into account the incomplete efficiency of detection and quality cuts for this component at a given energy. While in our procedure the spectra for all components are computed assuming the full efficiency. This leads to an appearance of extra uncertainty in low-energy bins. We estimate the uncertainty related to this assumption by calculating the difference between the spectrum reconstructed with our method and the one reconstructed assuming the efficiency averaged over all components in a given energy bin. There is an 8% uncertainty for the first bin, a 1% for the second bin, and no extra uncertainty for the higher energy bins. The summary of all calculated uncertainties is given in table 2.

Table 2. Summary of relative values of systematic uncertainties for CNN reconstruction of the mass components spectra. Monte-Carlo set with QGSJet-II.02 hadronic interaction model is used. All the uncertainty values are averaged over all mass components, the range of the values shown is an uncertainty variation within the energy range studied. The values in brackets include the uncertainty of the regularization bias of the particle type unfolding, computed including outlier bins. For missing detectors, the uncertainty of the energy unfolding is given as an average overall energy range studied. Total uncertainty corresponds to a sum of all uncertainties in quadrature.

Unfolding type Uncertainty	Particle type	Energy
Missing detectors	1–20%	4%
Limited MC	6–20%	2–15%
Unfolding regularization	1–11(65)%	0.1–13%
Substitution $P(A_i^{\text{tr}}\|E_j^{\text{tr}}) \to P(A_i^{\text{tr}}\|E_j^{\text{rec}})$	0–13%	—
Energy resolution in mixes	—	8–10%
Energy spectrum index in MC	—	0.5–7%
Detection & cuts efficiency	—	0–8%
Total (13–30(79)%)	6–24(73)%	11–20%

The test of the particle type unfolding with the Monte-Carlo sets for three different ML methods is shown in figure 18. We divided the Monte-Carlo test sets into two equal parts. The first part was used for the response matrix computation, while the second part — for the spectra unfolding. Note, that the results presented in this figure are based on the Monte-Carlo set with QGSJet-II.04 hadronic interaction model, while the general uncertainties calculations presented in this paragraph were based on the Monte-Carlo set with QGSJet-II.02 model. From this figure, one can derive two conclusions. First, the predictions of all three ML methods have a general agreement with the spectra of the true components, within their uncertainties. Second, there is also a good agreement between the unfolded predictions of different ML methods. Both of these findings provide additional credibility to our unfolding method.

6.3 Comparison with the standard KASCADE reconstruction

In this section, we compare our results for the unblind part of experimental data with the original KASCADE results of mass components spectra reconstruction [4, 7, 29]. First, we present the result of our reconstruction of the all-particle spectrum. It includes the energy unfolding and all uncertainties related to the energy reconstruction, as described in section 6.2. Namely, the uncertainties of the finite response matrix, missing detectors impact, impact of mixtures, efficacy, different γ of the MC total flux (-2.7 and -3.0), and the unfolding method are included. The comparison of these results with the original KASCADE all-particle spectrum is shown in figure 19. One can see that both spectra are in agreement within the uncertainties in almost all studied energy range. However, our spectrum has smaller uncertainties, around 11–20%, than $\sim$ 25% uncertainty of the original KASCADE spectrum. Note, that our statistical uncertainties are smaller since we use the larger data set: our unblind set contains $\sim 1.6 \cdot 10^6$ events after quality cuts (without $\lg(E/eV) > 15.15$ cut), while the set used in the latest KASCADE analysis [29] contains only $\sim 6.4 \cdot 10^5$ events. Note also, that our all-particle spectrum reconstruction does not use any ML methods. The minor discrepancy in

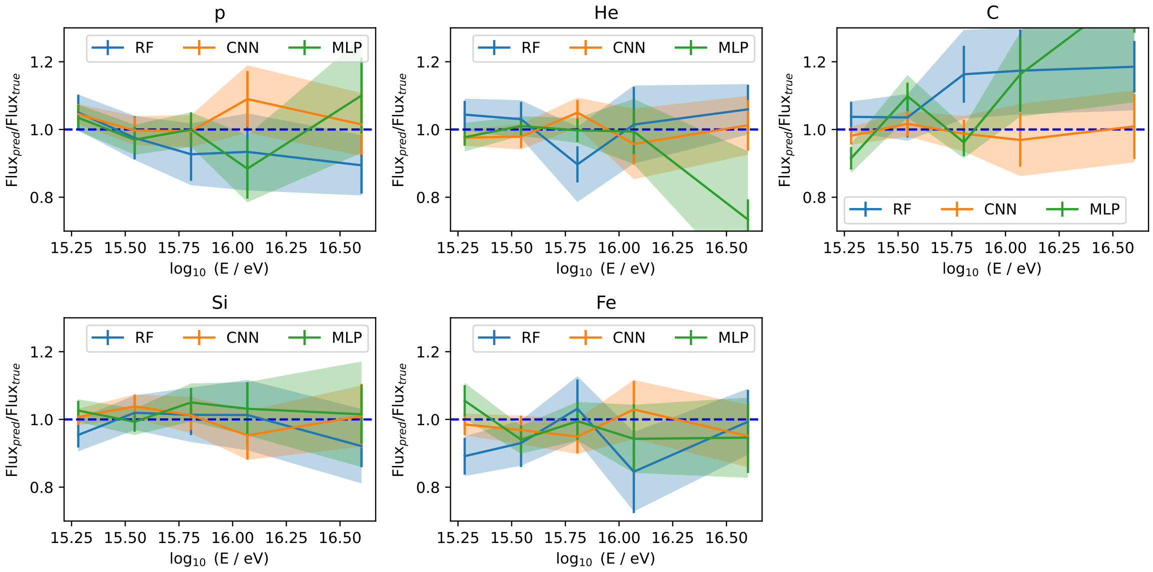


Figure 18. Predicted spectra for each mass component after particle type unfolding performed with the different ML models. Monte-Carlo set with QGSJet-II.04 hadronic interaction model is used. Whiskers represent statistical errors. Bands represent systematic uncertainty that includes the estimated bias of the unfolding procedure and the uncertainty of the response matrix.

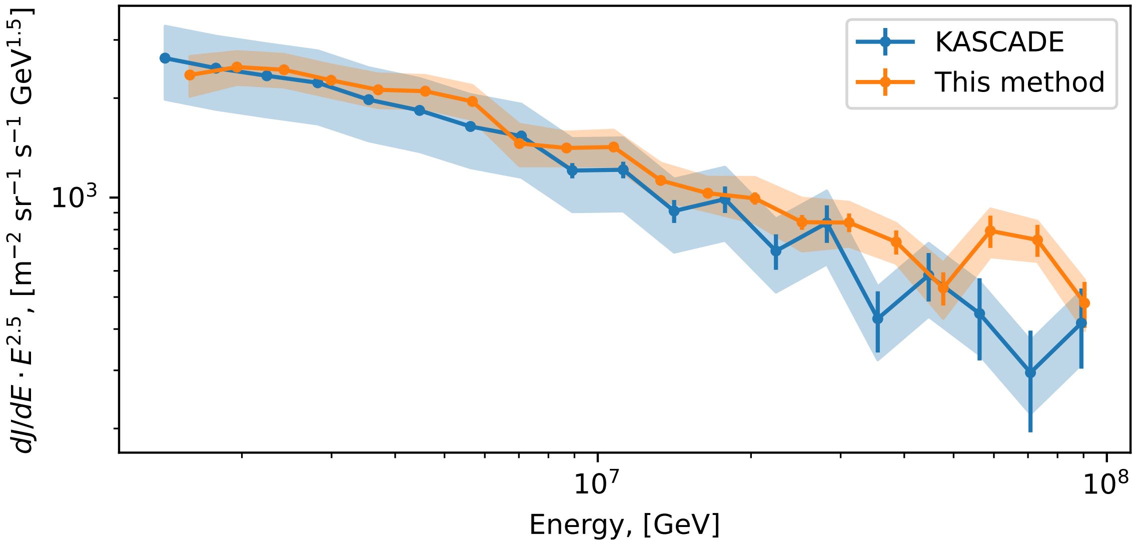


Figure 19. Comparison of all-particle spectra with energy unfolding for our reconstruction and the original KASCADE reconstruction. Energy estimation is based on the QGSJet-II.02 hadronic interaction model in both spectra. Whiskers show the statistical uncertainties. Bands denote systematic uncertainties (see text).

the high energy part of the spectra can happen because of the non-smoothness of the energy unfolding in these bins, that in turn originates from the small amount of Monte-Carlo events in these bins and resulting fluctuations of the confusion matrices from bin to bin: the uncertainty due to unfolding bias could be somewhat underestimated for these bins. This situation is characteristic for both our method and the original KASCADE reconstruction.

Finally, we present the comparison of separate mass component spectra reconstructed using our CNN method with that of the original KASCADE method. Again, the unblind part of the experimental data set is used for our reconstruction. We perform an unfolding and take into account all the uncertainties described in section 6.2. Our method is trained and tested with the QGSJet-II.02 hadronic interaction model, to make the comparison valid. The results are shown in figure 20. One can see that the uncertainties of our method are smaller than those of the original KASCADE method. Nevertheless, the spectra for all components except protons are generally in agreement with the uncertainties.

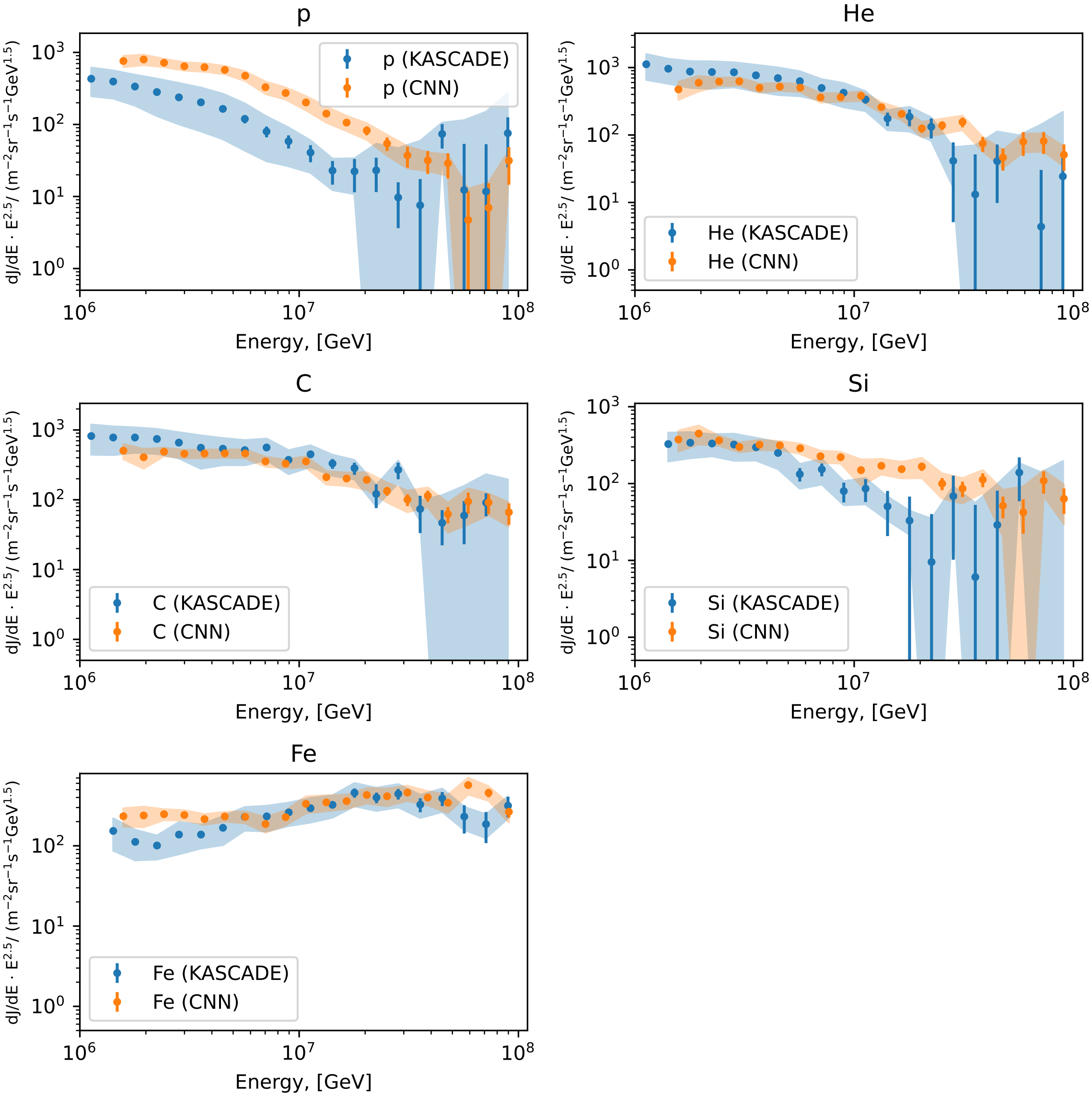


Figure 20. Comparison of the mass components spectra for the experimental data reconstructed using the original KASCADE methods [4] and using our CNN method. QGSJet-II.02 hadronic interaction model is used in both cases.

7 Conclusions

In this study, we have developed several new machine learning methods for the analysis of the public data of the KASCADE experiment. Our goal was the reconstruction of the energy spectra of separate mass components of cosmic ray flux in the energy range from 1 to 100 PeV. We have tested the performance of 4 different ML methods built for mass components classification: random forest, multi-layer perceptron, convolutional neural network, and EfficientNet. All the ML methods were trained and tested with the Monte-Carlo data of the KASCADE experiment. The best accuracy and the most stable performance were demonstrated by the CNN model. We made several cross-checks and stability tests for this network, including tests with a small part of the real data, the unblind set. All the tests were passed successfully proving the stability of the CNN. We also performed the unfolding of the reconstructed mass components spectra and estimated various uncertainties related to this procedure and to Monte-Carlo in general. The resulting all-particle and component-wise spectra built with the unblind data set were compared to that from the original composition analysis of the KASCADE experiment. The overall accuracy of our reconstruction methods was shown to significantly exceed that of the original KASCADE method for both all-particle and component-wise spectra.

Acknowledgments

We are grateful to Dmitry Kostunin for the inspiration of this study and for the assistance at its various stages. We would like to thank Grigory Rubtsov, Vladimir Lenok, and Victoria Tokareva for the fruitful discussions and comments. The work was supported by the Russian Science Foundation grant 22-22-00883.

A Details of the neural networks architectures

This section presents details of the implemented neural network models. In particular, the architecture of the CNN model is shown in figure 21 and that of the MLP model is shown in figure 22. For both CNN and MLP models, we first pre-process the data before feeding it to the input of the classifier. The values of deposits are rescaled to zero-mean and unit-variance normal distribution $\mathcal{N}(0,1)$, and reconstructed features are min-max scaled into the range $[0,1]$. Minor data augmentation, such as event image rotation, is also used to make the training process more stable.

The architecture of the CNN model consists of two main parts. The first part is convolutional filters. We apply a set of these filters to deposits from e/γ and μ detectors. This part is designed to identify the patterns corresponding to a particular primary mass component. The second part is dense layers. It supplies the results of the first part with reconstructed high-level EAS event features and combines them for the classification of the mass components. Therefore, the input of the CNN model consists of deposits from e/γ and μ detectors represented as 16×16 image with 2 channels and high-level features: $\log_{10} N_e$, $\log_{10} N_\mu$, θ, s. The CNN classifier is implemented in PyTorch [30]. The model has ~ 36 000 trainable parameters. We use Adam optimizer [24] for training this model and Weights & Biases package [31] for tracking of the training process.

The MLP model's architecture is extremely simple: it consists of only two dense (feedforward) layers, each followed by batch normalization, ELU activation, and dropout with a rate of 0.15. The MLP model uses a slightly smaller number of inputs compared to the CNN model: it similarly deposits

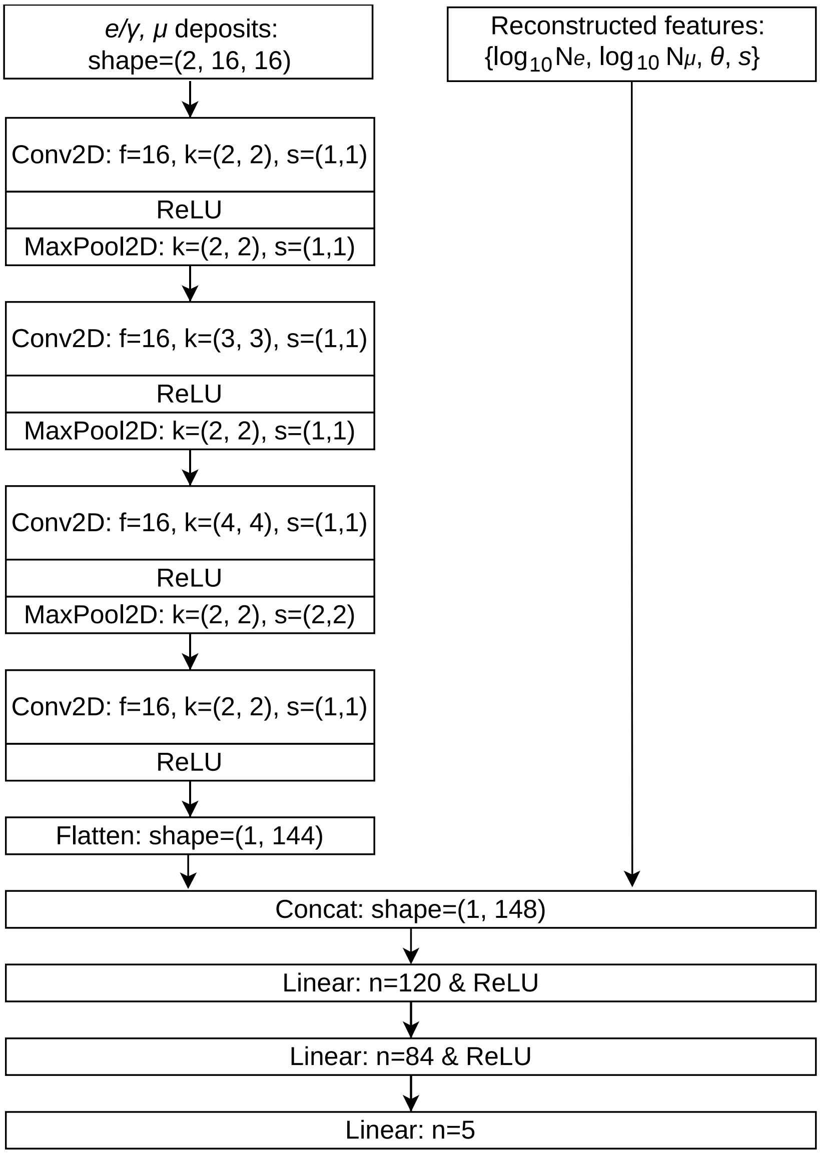

e/γ, μ deposits:
shape=(2, 16, 16)
Reconstructed features:
{log10Ne, log10 Nμ, θ, s}
Conv2D: f=16, k=(2, 2), s=(1,1)
ReLU
MaxPool2D: k=(2, 2), s=(1,1)
Conv2D: f=16, k=(3, 3), s=(1,1)
ReLU
MaxPool2D: k=(2, 2), s=(1,1)
Conv2D: f=16, k=(4, 4), s=(1,1)
ReLU
MaxPool2D: k=(2, 2), s=(2,2)
Conv2D: f=16, k=(2, 2), s=(1,1)
ReLU
Flatten: shape=(1, 144)
Concat: shape=(1, 148)
Linear: n=120 & ReLU
Linear: n=84 & ReLU
Linear: n=5

Figure 21. Architecture of the convolutional neural network. Data flow is represented by arrays. Layer names are given as in PyTorch. Abbreviations: “f” — the number of filters, “k” — kernel size, “s” — strides, “n” — the number of neurons.

from e/γ and μ detectors (although this time they’re converted to 1-dimensional flat array format) and only two high-level features: θ and ϕ. The model has $\sim 200\,000$ trainable parameters and is implemented using TensorFlow [32] and Keras [33].

B Details and tests for random forest model

Here we present the details for the RF model optimization and show some tests performed for this model. The following parameters were optimized for the RF classifier:

- The number of estimators (trees): [1, 10, 100, 500, 1000],
- The maximum depth of the tree: [5, 10, 50, 100, ‘not limited’],

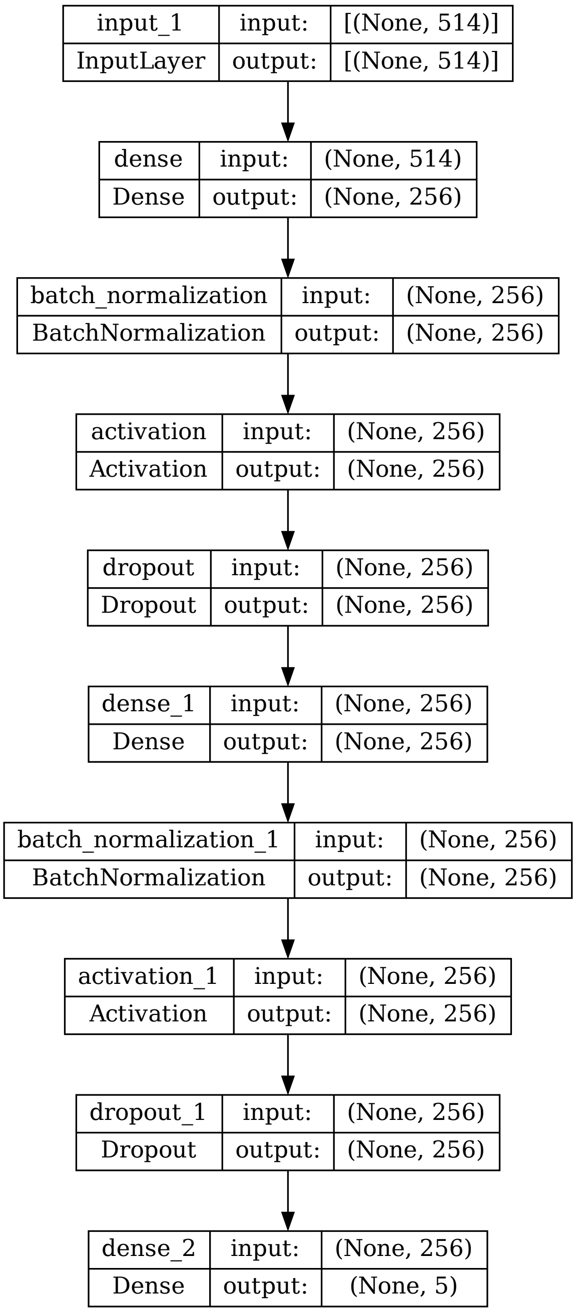
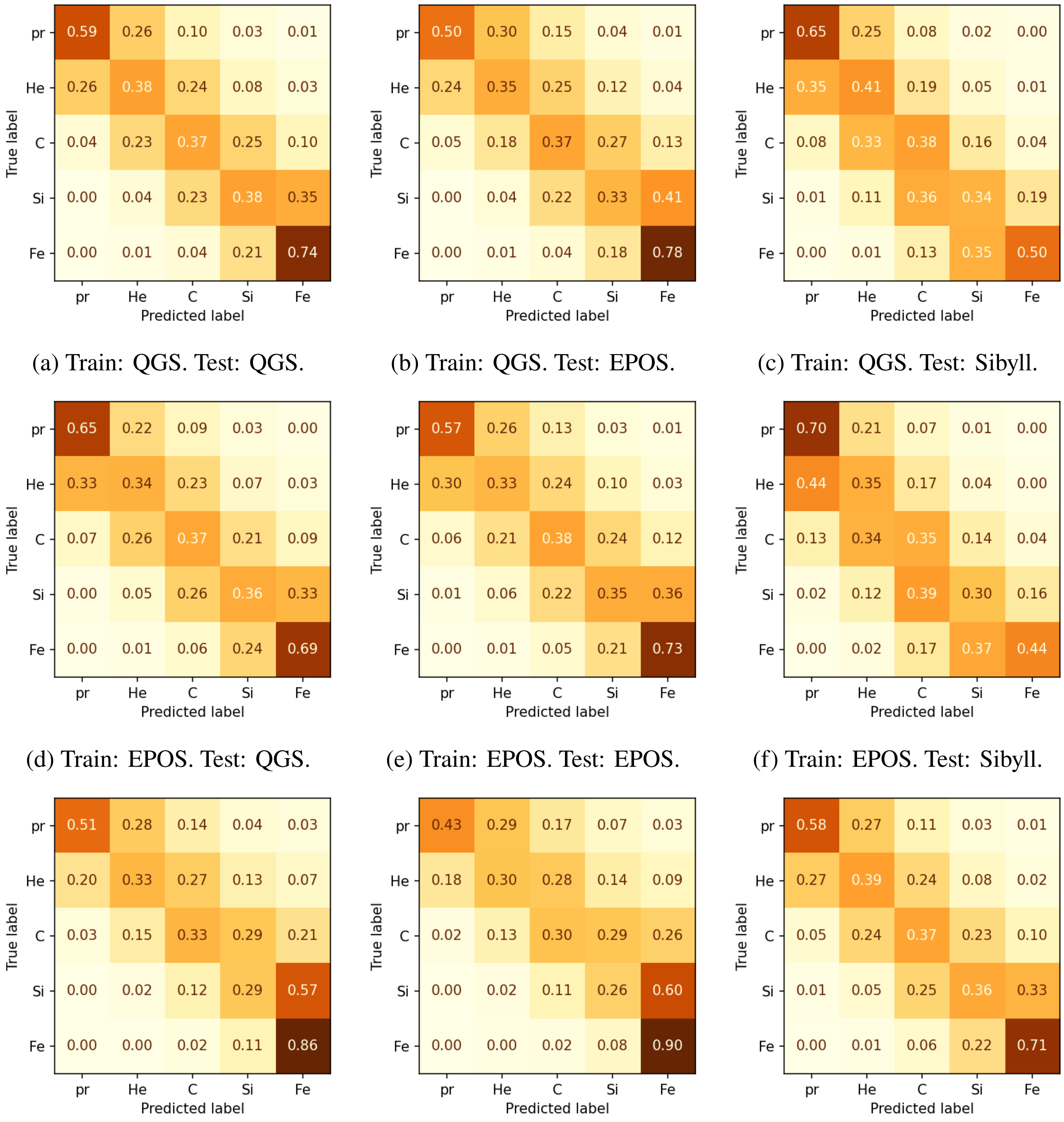

input_1 | input: | [(None, 514)]
InputLayer | output: | [(None, 514)]
dense | input: | (None, 514)
Dense | output: | (None, 256)
batch_normalization | input: | (None, 256)
BatchNormalization | output: | (None, 256)
activation | input: | (None, 256)
Activation | output: | (None, 256)
dropout | input: | (None, 256)
Dropout | output: | (None, 256)
dense_1 | input: | (None, 256)
Dense | output: | (None, 256)
batch_normalization_1 | input: | (None, 256)
BatchNormalization | output: | (None, 256)
activation_1 | input: | (None, 256)
Activation | output: | (None, 256)
dropout_1 | input: | (None, 256)
Dropout | output: | (None, 256)
dense_2 | input: | (None, 256)
Dense | output: | (None, 5)

Figure 22. Architecture of the feed-forward neural network (MLP). The flow of data is shown by arrays. Layer names are given in TensorFlow format. "None" in tensor shapes means variable batch size shape.

- Split quality criteria: ['gini', 'entropy'],
- Maximum number of features to consider when looking for the best split: ['num features', 'sqrt(num features)'],
- Class weights: ['equal', 'balanced'].



(a) Train: QGS. Test: QGS.

(b) Train: QGS. Test: EPOS.

(c) Train: QGS. Test: Sibyll.

(d) Train: EPOS. Test: QGS.

(e) Train: EPOS. Test: EPOS.

(f) Train: EPOS. Test: Sibyll.

(g) Train: Sibyll. Test: QGS.

(h) Train: Sibyll. Test: EPOS.

(i) Train: Sibyll. Test: Sibyll.

Figure 23. Cross-hadronic model confusion matrices for the RF classifier models trained and tested separately with three different hadronic interaction models: QGSJet-II.04, EPOS-LHC and Sibyll 2.3c which are marked in the figure as QGS, EPOS, and Sibyll, respectively.

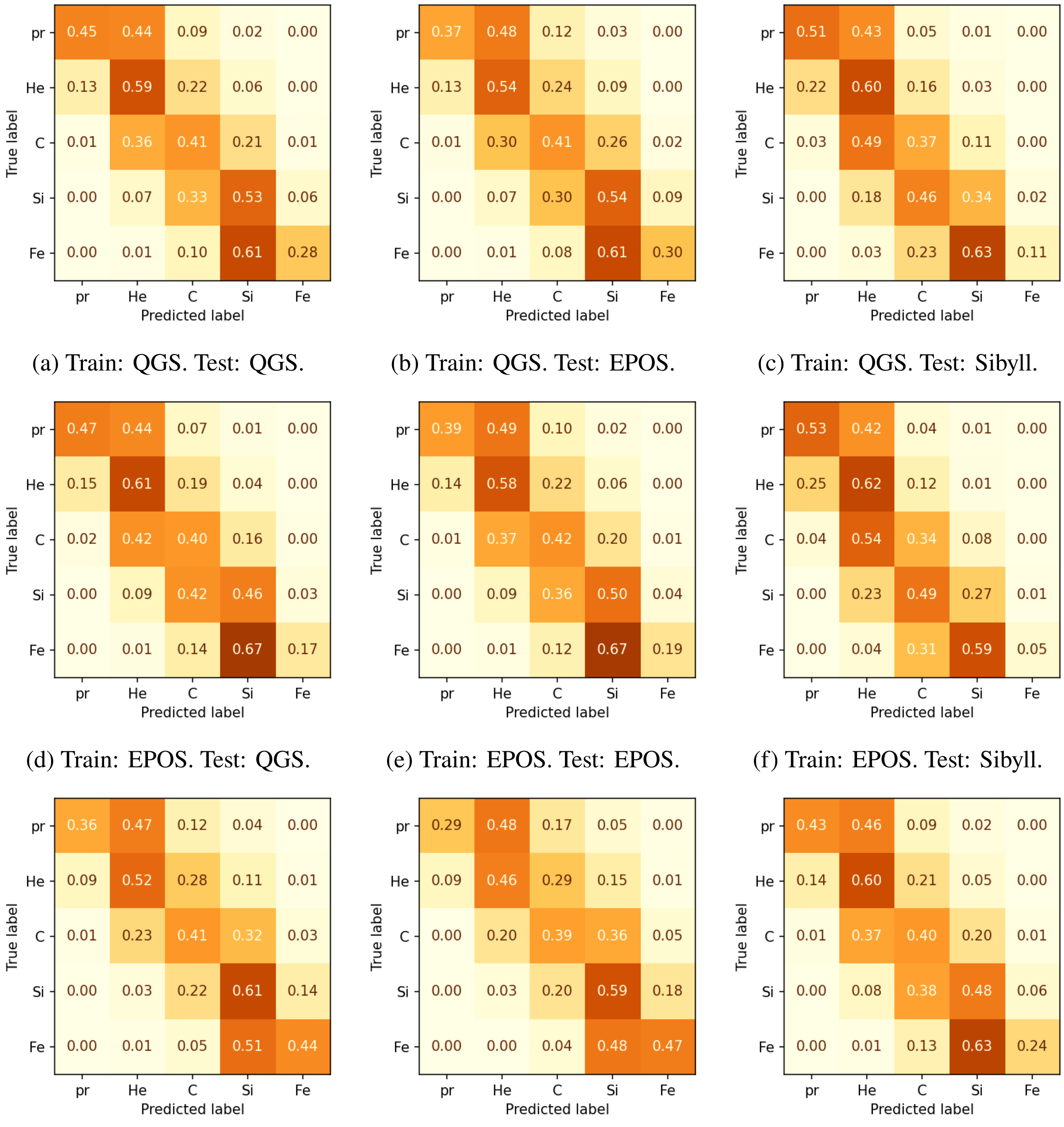
In figure 23 we show the reconstruction of cross-hadronic models for the RF classifier. One can see that the performance of this classifier for cross-hadronic model reconstruction is close to that of CNN, with a little less accuracy for intermediate mass components.

Apart from the RF classifier model the RF regressor model was designed. The following parameters were optimized for this model:

- The number of estimators (trees): [1, 10, 100, 500, 1000],
- The maximum depth of the tree: [5, 10, 50, 100, 'not limited'],

- Split quality criteria: ['squared error', 'absolute error'],

- Maximum number of features to consider when looking for the best split: ['num features', 'sqrt(num features)'].



(a) Train: QGS. Test: QGS.

(b) Train: QGS. Test: EPOS.

(c) Train: QGS. Test: Sibyll.

(d) Train: EPOS. Test: QGS.

(e) Train: EPOS. Test: EPOS.

(f) Train: EPOS. Test: Sibyll.

(g) Train: Sibyll. Test: QGS.

(h) Train: Sibyll. Test: EPOS.

(i) Train: Sibyll. Test: Sibyll.

Figure 24. Cross-hadronic model confusion matrices for the RF regressor models trained and tested separately with three different hadronic interaction models: QGSJet-II.04, EPOS-LHC and Sibyll 2.3c which are marked in the figure as QGS, EPOS, Sibyll, respectively.

In figure 24 we show the reconstruction of cross-hadronic models for the RF regressor. As one can see, this ML model shows a significant off-diagonality of the confusion matrices, especially in the “light” and “heavy” parts of the matrix. Importantly, this problem persists even if the regressor is trained and tested with one and the same hadronic interaction model. This makes the given ML model unsuitable for the mass composition reconstruction in the given setup.

References

[1] S. Gabici et al., *The origin of Galactic cosmic rays: challenges to the standard paradigm*, *Int. J. Mod. Phys. D* **28** (2019) 1930022 [arXiv:1903.11584].

[2] B. Peters, *Primary cosmic radiation and extensive air showers*, *Nuovo Cim.* **22** (1961) 800.

[3] T.K. Gaisser, *Spectrum of cosmic-ray nucleons, kaon production, and the atmospheric muon charge ratio*, *Astropart. Phys.* **35** (2012) 801 [arXiv:1111.6675].

[4] W.D. Apel et al., *KASCADE-Grande measurements of energy spectra for elemental groups of cosmic rays*, *Astropart. Phys.* **47** (2013) 54 [arXiv:1306.6283].

[5] IceCube collaboration, *Cosmic ray spectrum and composition from PeV to EeV using 3 years of data from IceTop and IceCube*, *Phys. Rev. D* **100** (2019) 082002 [arXiv:1906.04317].

[6] Telescope Array collaboration, *The Cosmic-Ray Composition between 2 PeV and 2 EeV Observed with the TALE Detector in Monocular Mode*, *Astrophys. J.* **909** (2021) 178 [arXiv:2012.10372].

[7] KASCADE collaboration, *KASCADE measurements of energy spectra for elemental groups of cosmic rays: Results and open problems*, *Astropart. Phys.* **24** (2005) 1 [astro-ph/0505413].

[8] D. Ivanov et al., *Using deep learning to enhance event geometry reconstruction for the telescope array surface detector*, *Mach. Learn. Sci. Tech.* **2** (2021) 015006 [arXiv:2005.07117].

[9] O. Kalashev et al., *Deep learning method for identifying mass composition of ultra-high-energy cosmic rays*, 2022 *JINST* **17** P05008 [arXiv:2112.02072].

[10] Telescope Array collaboration, *Mass composition of ultrahigh-energy cosmic rays with the Telescope Array Surface Detector data*, *Phys. Rev. D* **99** (2019) 022002 [arXiv:1808.03680].

[11] M. Erdmann, J. Glombitza and D. Walz, *A deep learning-based reconstruction of cosmic ray-induced air showers*, *Astropart. Phys.* **97** (2018) 46 [arXiv:1708.00647].

[12] Pierre Auger collaboration, *Deep-learning based reconstruction of the shower maximum X_{max} using the water-Cherenkov detectors of the Pierre Auger Observatory*, 2021 *JINST* **16** P07019 [arXiv:2101.02946].

[13] KASCADE collaboration, *The Cosmic ray experiment KASCADE*, *Nucl. Instrum. Meth. A* **513** (2003) 490.

[14] J. Wochele, D. Kang, D. Wochele, A. Haungs and S. Schoo, *KCDC User Manual: Open Access Solution for the KASCADE*, version 3.1 (2023), https://kcdc.iap.kit.edu/static/pdf/kcdc_mainpage/kcdc-Manual.pdf [DOI:10.17616/R3TS4P].

[15] A. Haungs et al., *The KASCADE Cosmic-ray Data Centre KCDC: Granting Open Access to Astroparticle Physics Research Data*, *Eur. Phys. J. C* **78** (2018) 741 [arXiv:1806.05493].

[16] S. Ostapchenko, *QGSJET-II: Towards reliable description of very high energy hadronic interactions*, *Nucl. Phys. B Proc. Suppl.* **151** (2006) 143 [hep-ph/0412332].

[17] A. Ferrari, P.R. Sala, A. Fasso and J. Ranft, *FLUKA: A multi-particle transport code (program version 2005)*, CERN-2005-010 [SLAC-R-773] [DOI:10.2172/877507].

[18] D. Heck, J. Knapp, J. Capdevielle, G. Schatz, T. Thouw et al., *CORSIKA: A Monte Carlo code to simulate extensive air showers*, FZKA-6019 (1998).

[19] S. Ostapchenko, *Monte Carlo treatment of hadronic interactions in enhanced Pomeron scheme. Part I. QGSJET-II model*, *Phys. Rev. D* **83** (2011) 014018 [arXiv:1010.1869].

[20] T. Pierog, I. Karpenko, J.M. Katzy, E. Yatsenko and K. Werner, *EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider*, *Phys. Rev. C* **92** (2015) 034906 [arXiv:1306.0121].

[21] F. Riehn, R. Engel, A. Fedynitch, T.K. Gaisser and T. Stanev, *Charm production in SIBYLL*, *EPJ Web Conf.* **99** (2015) 12001 [arXiv:1502.06353].

[22] T.K. Ho, *Random decision forests*, in the proceedings of the *3rd International Conference on Document Analysis and Recognition*, Montreal, QC, Canada, 14–16 August 1995, IEEE (1995), pp. 278–282 [DOI:10.1109/ICDAR.1995.598929].

[23] F. Pedregosa et al., *Scikit-learn: Machine Learning in Python*, *J. Mach. Learn. Res.* **12** (2011) 2825 [arXiv:1201.0490].

[24] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980.

[25] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, *Proc. IEEE* **86** (1998) 2278.

[26] M. Tan and Q.V. Le, *EfficientNetV2: Smaller Models and Faster Training*, arXiv:2104.00298.

[27] G. D'Agostini, *A Multidimensional unfolding method based on Bayes' theorem*, *Nucl. Instrum. Meth. A* **362** (1995) 487.

[28] J. Bourbeau and Z. Hampel-Arias, *Pyunfold: A python package for iterative unfolding*, *J. Open Source Softw.* **3** (2018) 741.

[29] M.R. Finger, *Reconstruction of energy spectra for different mass groups of high-energy cosmic rays*, Ph.D. Thesis, Fakultät für Physik des Karlsruher Instituts für Technologie (KIT), Karlsruher, Germany (2011).

[30] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, in *Advances in Neural Information Processing Systems* **32**, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox and R. Garnett eds., Curran Associates, Inc. (2019), pp. 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[31] L. Biewald, *Experiment tracking with weights and biases*, (2020), https://www.wandb.com/.

[32] M. Abadi et al., *TensorFlow: A system for large-scale machine learning*, in the proceedings of the *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, Savannah, GA, U.S.A., 2–4 November 2016, pp. 265–283 [arXiv:1605.08695].

[33] F. Chollet et al., *Keras*, (2015), https://keras.io.