Machine Learning Techniques For Particle Identification

06.March.2018 I **Waleed Esmail**, Tobias Stockmanns, Michael Kunkel, James Ritman

Institut für Kernphysik (IKP), Forschungszentrum Jülich

Outlines:

- Generation and Preparation.
- Decision Trees.
  - *Evaluation metrics.*
  - *Efficiency calculations.*
- Neural Networks.
  - *Evaluation metrics.*
- Comparison to the Classical PID Algorithms.
- Conclusion.

Generation & Preparation:

Uniform Generator (Box Generator):

- **momentum range**: (0.2 - 5) GeV.
- **phi range**: 0 - 360^{o}.
- **theta range**: 0 - 180^{o}.
- **particle species**: [$e^{\mp}$, $\pi^{\mp}$, $\mu^{\mp}$, $k^{\mp}$, $p^{\mp}$]. One particle per event.
- 750k events (150k for each type).
- 750k for (particles), and 750k for (antiparticles).
- Particles are matched to their **MC truth** information.

Machine Learning:

- Machine learning is about modeling your data.
- Ask the correct questions.
- ML algorithms fall into three broad categories:
  1. *Supervised learning.*
  2. *Unsupervised learning.*
  3. *Reinforcement learning.*

  Focus: *supervised learning (multi-class classification).*

- Inputs for the algorithm: ($\mathbf{A}\boldsymbol{x} = \mathbf{b}$)
  1. **Feature matrix** (`DataFrame`).
  2. **Target column vector**.

energy	momentum	charge	position	MvdHits	GemHits	SttHits	TofStopTime	TofTrackLength	EmcCalEnergy
1.45992	2.1119	-1	1.53556E-4	3	0	26	0.0	0.0	0.250083
4.14557	17.1663	-1	6.65813E-6	5	0	17	3.95565	117.048	0.991413
3.51102	12.3078	-1	0.00648225	1	0	24	0.0	0.0	2.21215
3.45948	11.9486	-1	5.4671E-6	4	0	22	2.89786	86.4262	0.29421
4.78585	22.8849	-1	6.36045E-5	3	0	26	0.0	0.0	2.62603

only showing top 5 rows

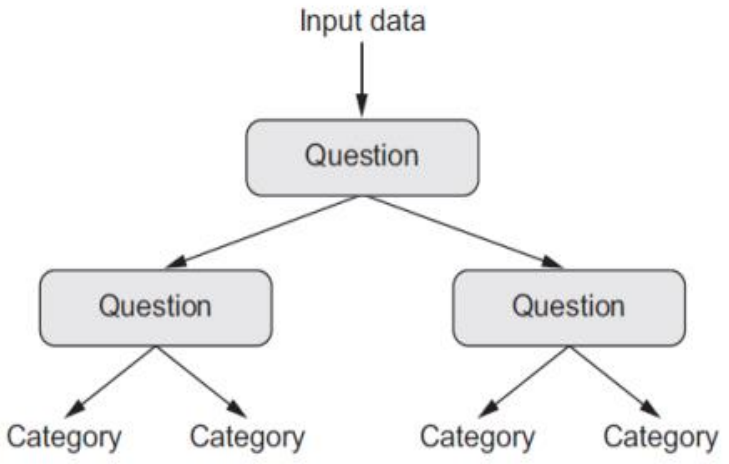
Decision Trees:

1) Follow the recipe, **choose the model (algorithm)**, fix the **hyper-parameters**, **apply** (fit) the model, and ***predict***.

1) Validate the model:

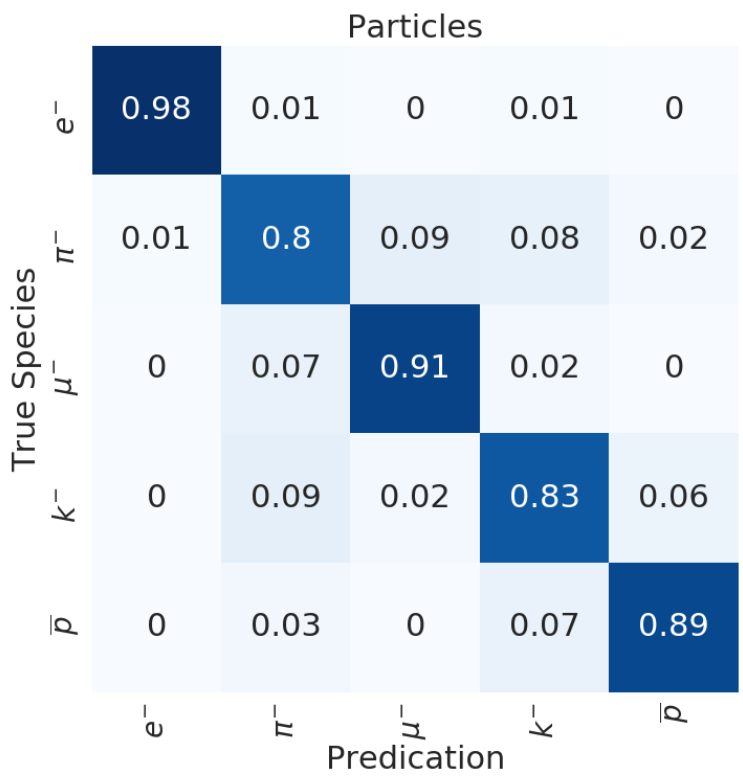
- *Tune the hyper-parameters, pickup more/less complicated model, etc…*

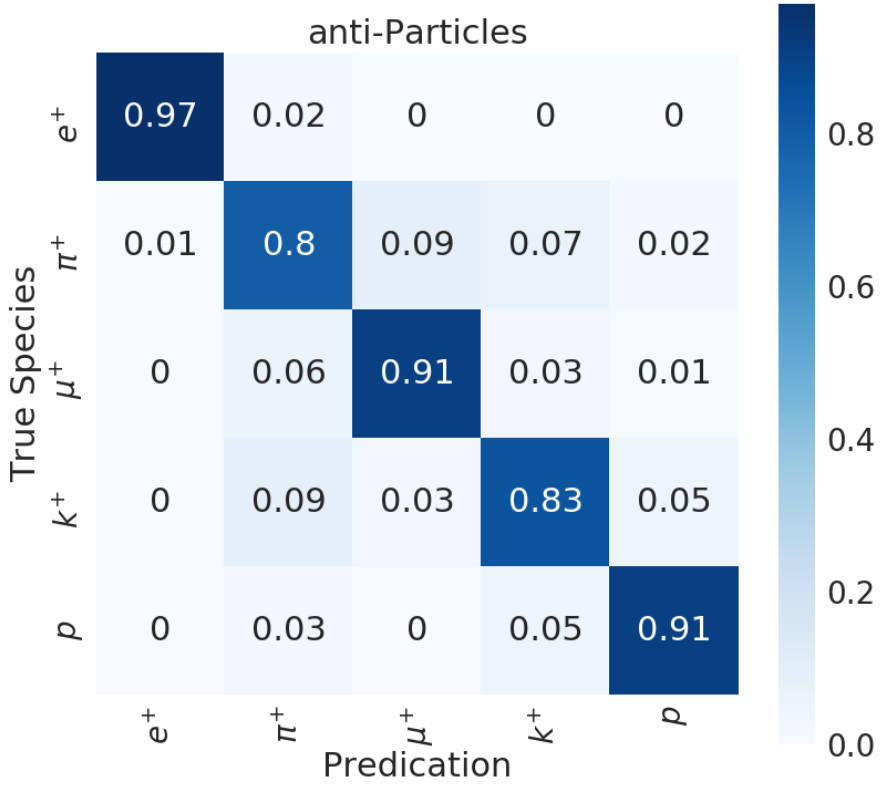
- **Random Forests** (**RF**) is used for the modeling.
- **RF** is an ensemble of decision trees.
  - ➢ *The algorithm makes a prediction for every tree in the forest.*
  - ➢ *Aggregate the predictions via hard voting.*
- **SciKit-Learn** Python Package was used.
- Data was split (*70% training*).



Evaluation metrics:

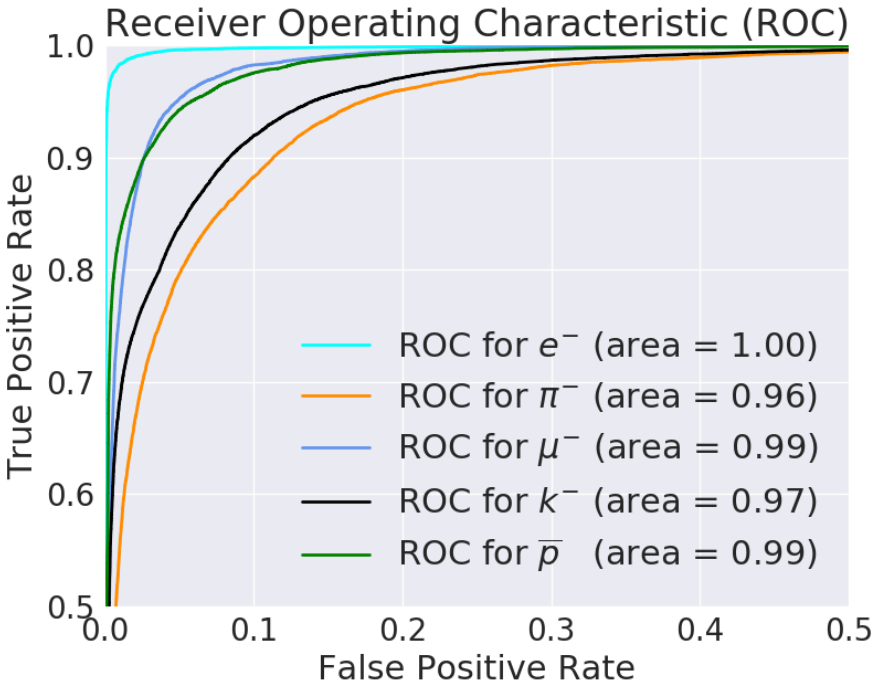
1. Confusion Matrices

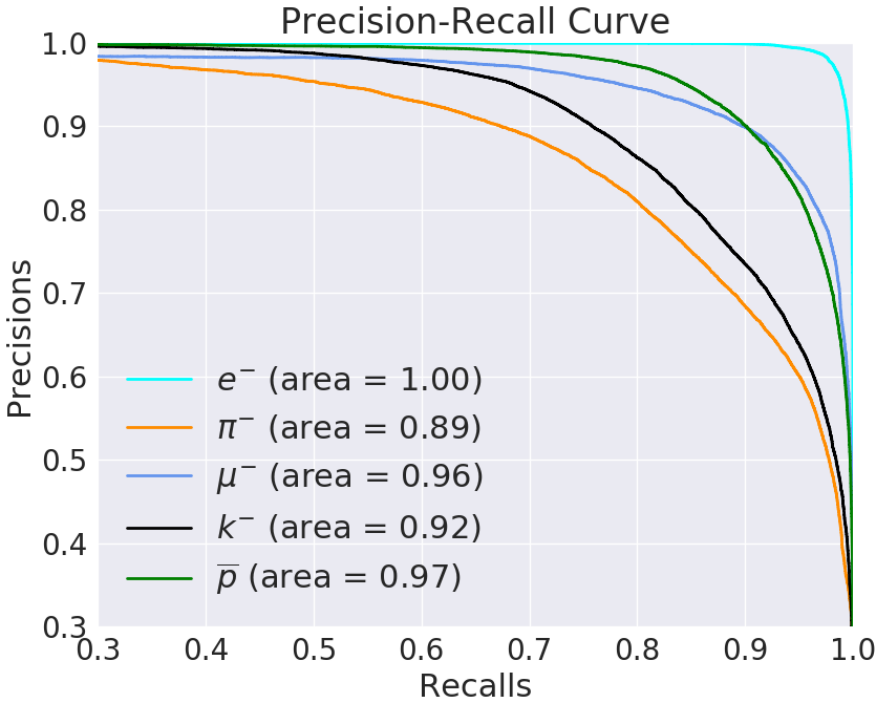




Evaluation metrics:

2. Receiver Operating Characteristic (ROC), Precision, and Recall.



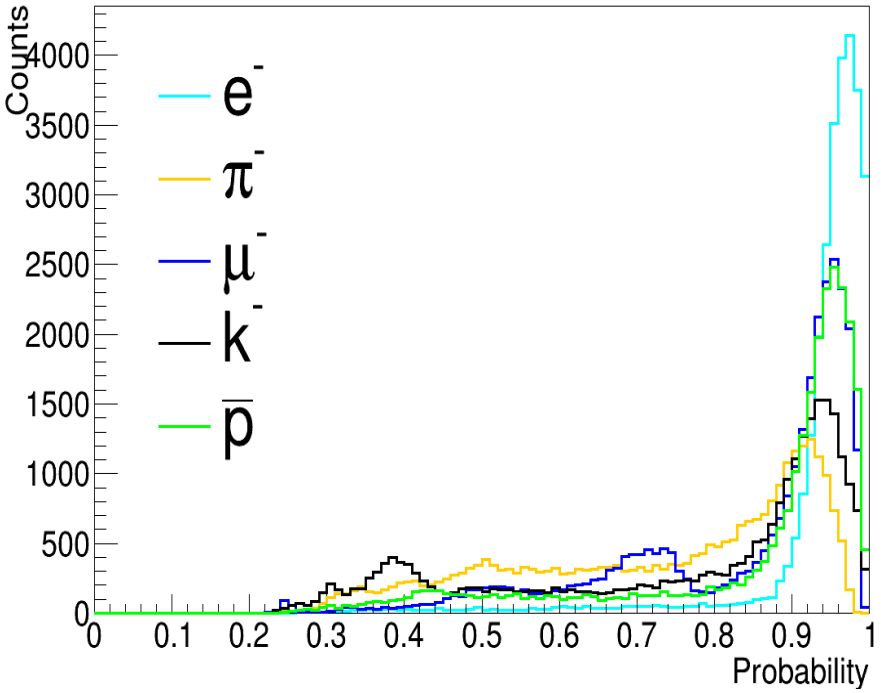


True Positives (TP) & True Negatives (TN).
False Positives (FP) & False Negatives (FN).

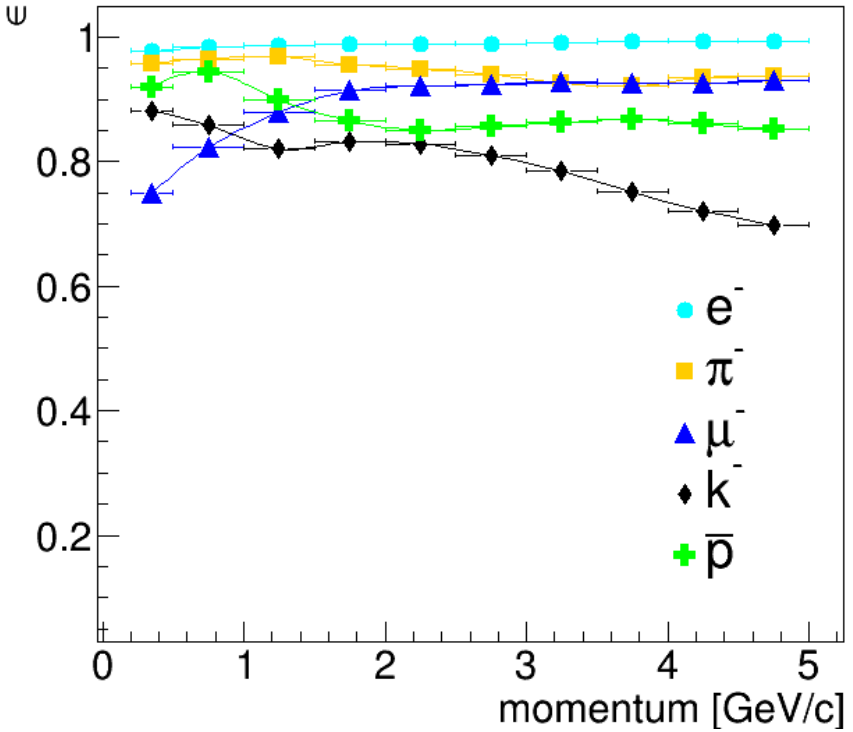
$$\text{recall} = \frac{TP}{TP + FN} \qquad \text{precision} = \frac{TP}{TP + FP}$$

Probabilities & Efficiencies:

Probability distributions



PID efficiency
p > 0.5

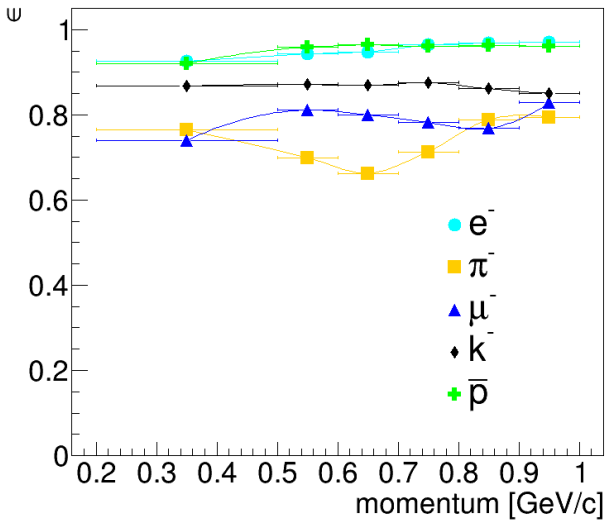


$$\epsilon = \frac{\boldsymbol{yield}\ \&\ \boldsymbol{p} > \mathbf{0.5}}{\boldsymbol{yield}}$$

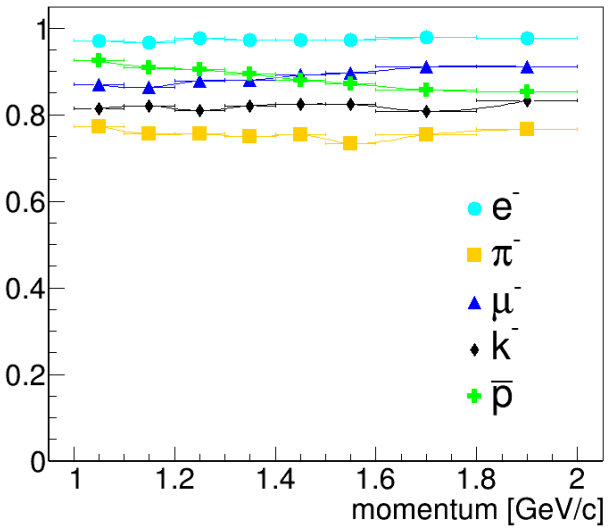
Efficiency for different momentums:

For each momentum range 250K events generated (50k per particle type).

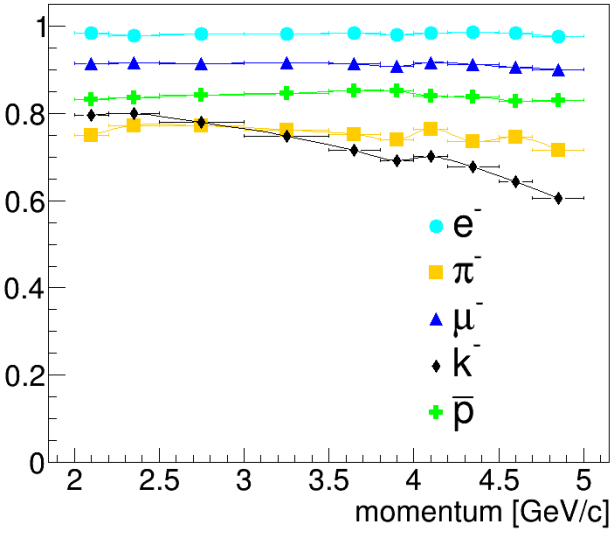
250k (0.2 – 1 [GeV/c])



250k (1 – 2 [GeV/c])



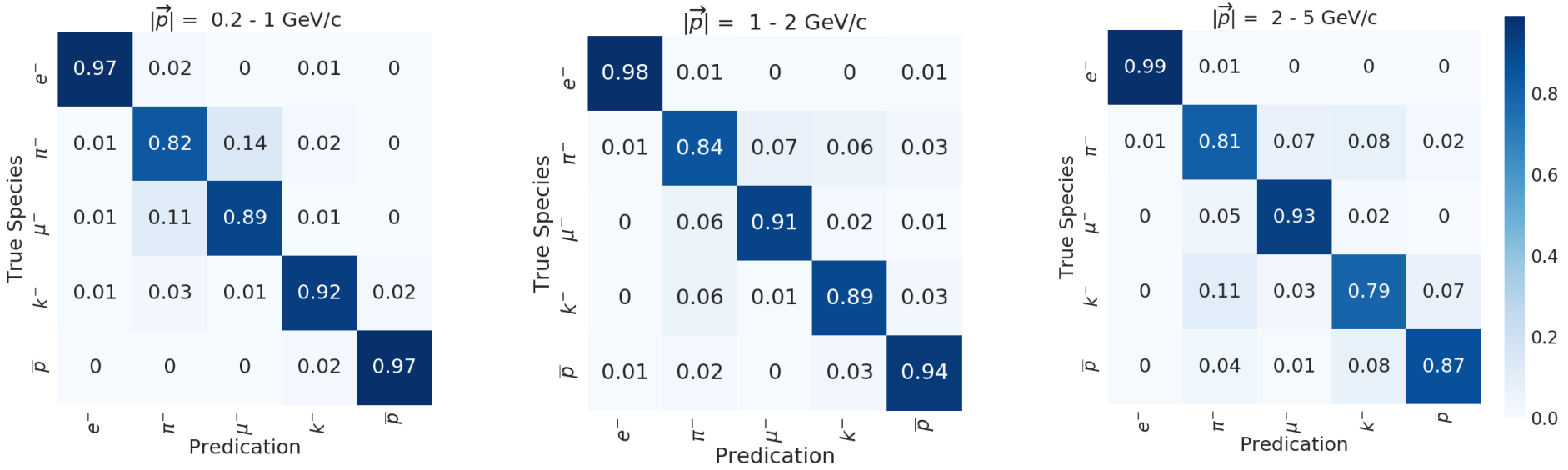
250k (2 – 5 [GeV/c])



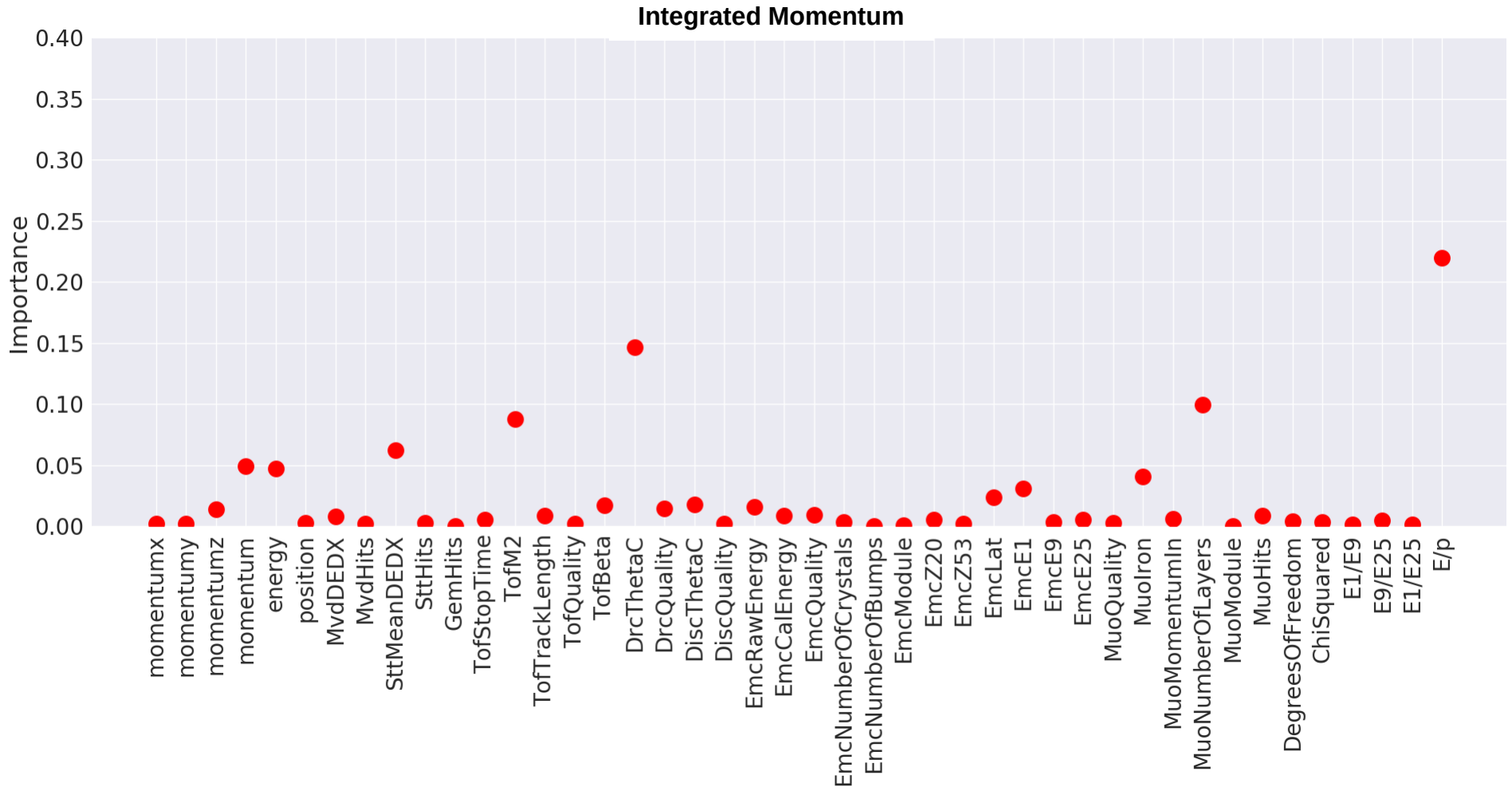
Evaluation metrics:

Confusion Matrices as function of momentum

For each momentum range 250K events generated (50k per particle type).

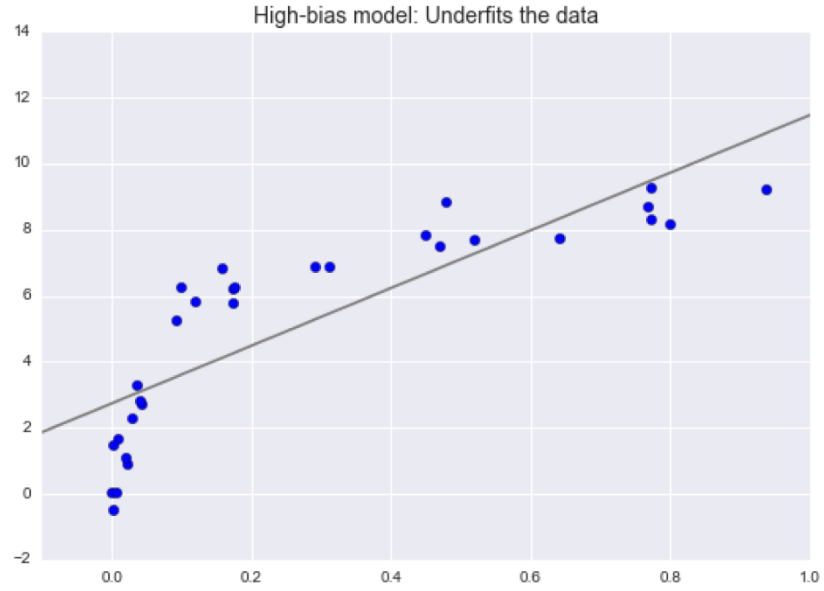


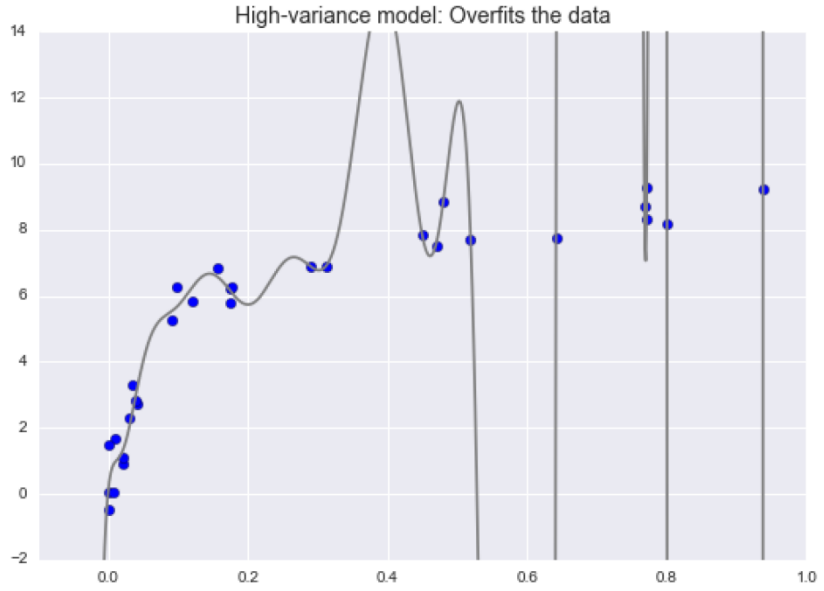
Features Significance:



Over-fitting check:

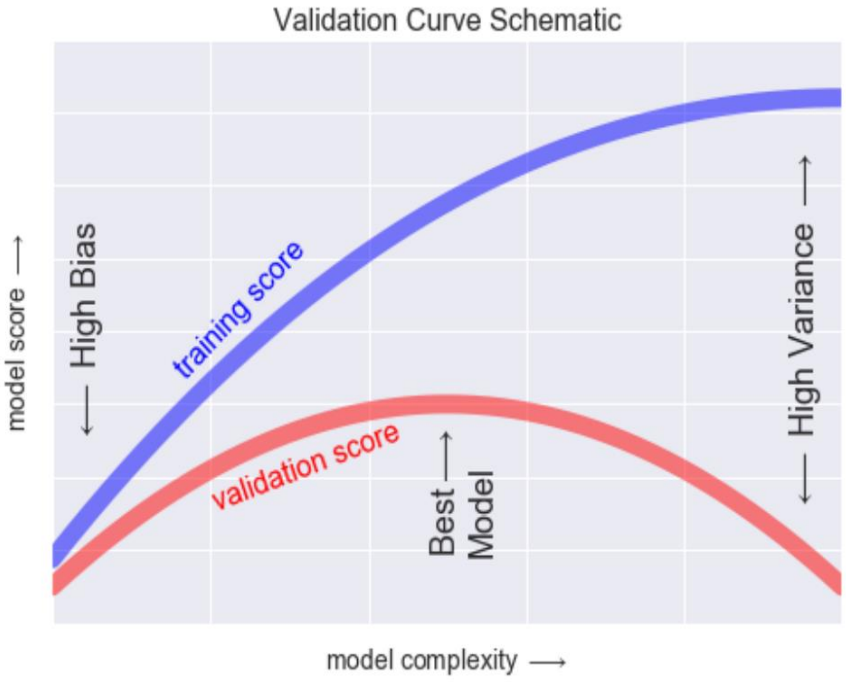
Consider these example regression fits to the same dataset

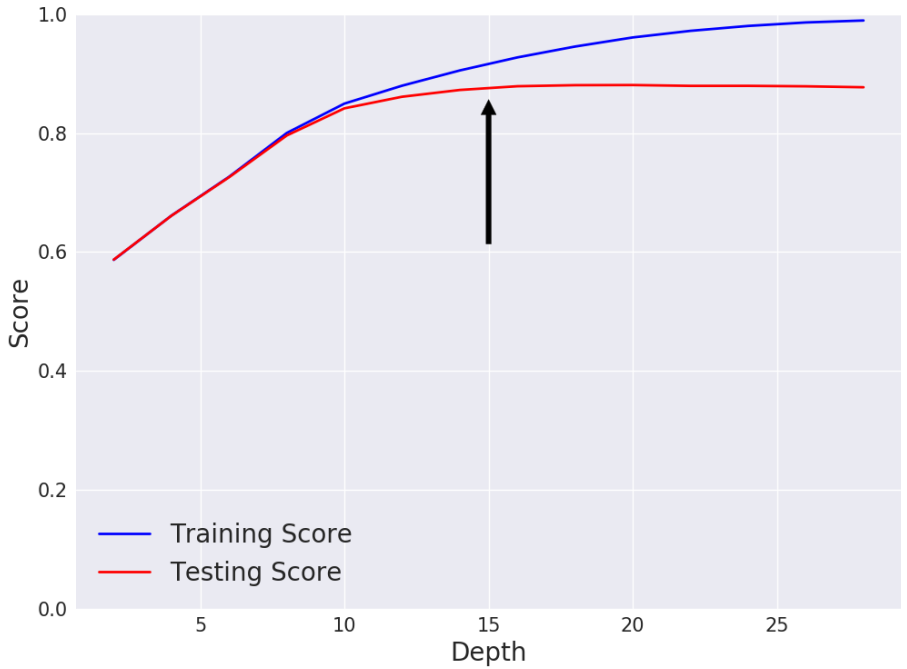




Over-fitting check:

Validation Curve:



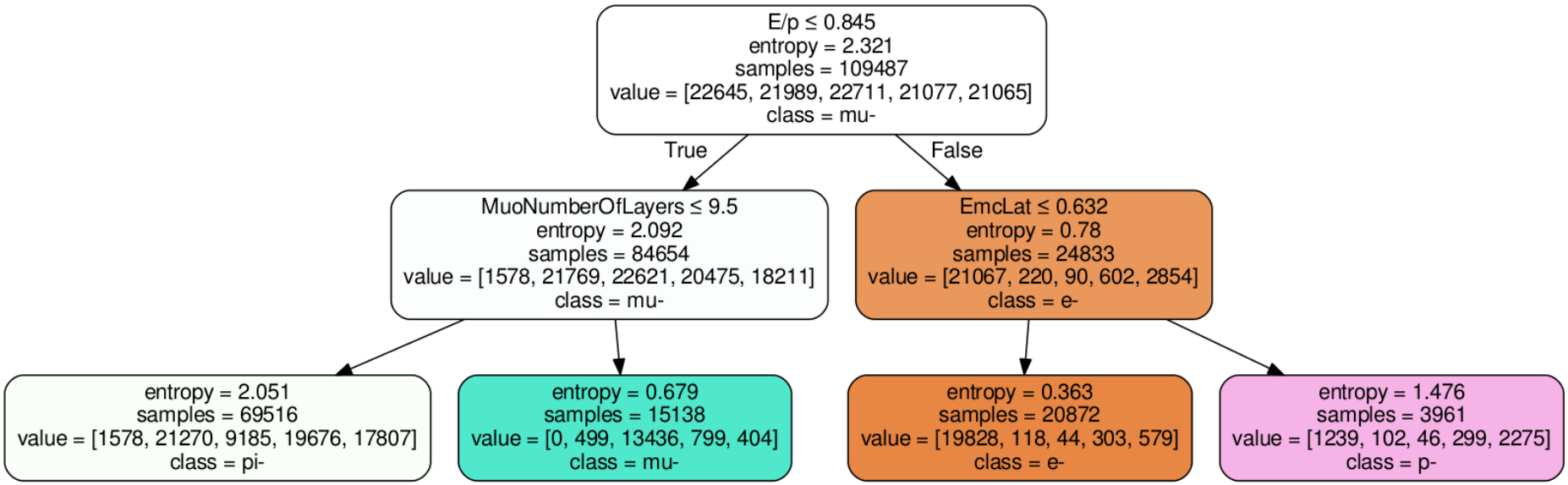


- *Grid search was used to tune the hyper-parameters of the algorithm.*

max_features	*max_depth*	*n_estimators*
35	***15***	***100***

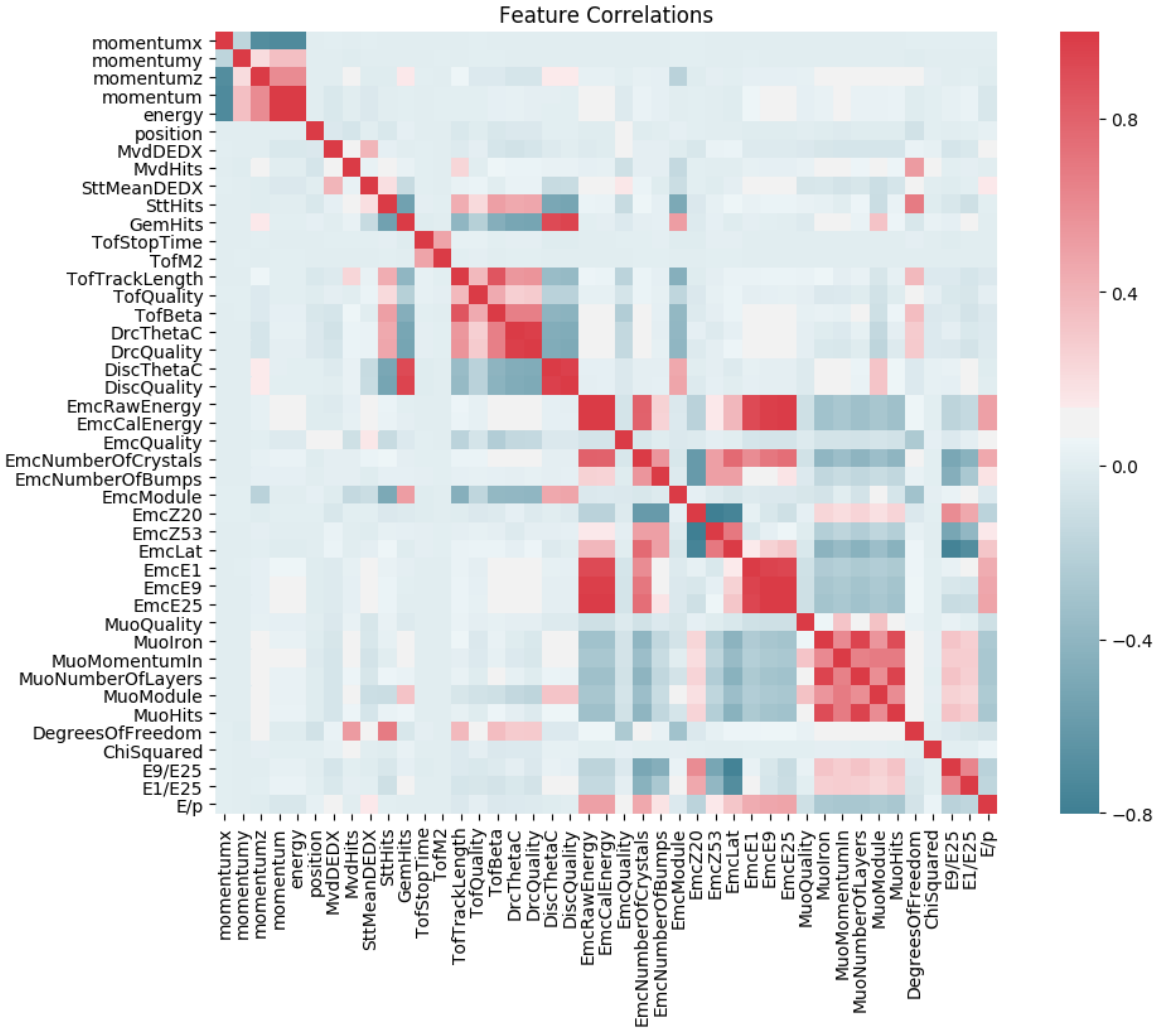
Tree visualization:

Information Gain: $$IG\left(D_p, f\right) = I\left(D_p\right) - \sum_{j=1}^{m} \frac{N_j}{N_p} I\left(D_j\right)$$



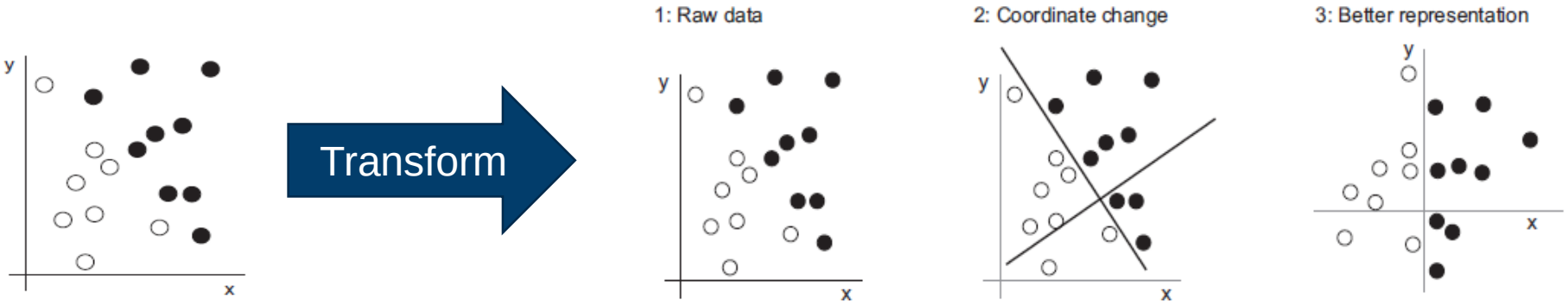
Correlation Matrix:

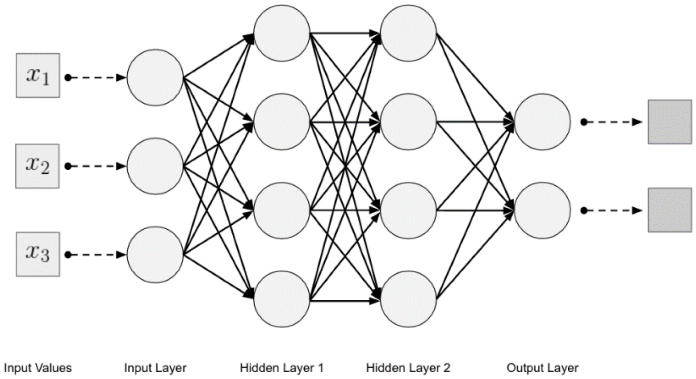
Input features :

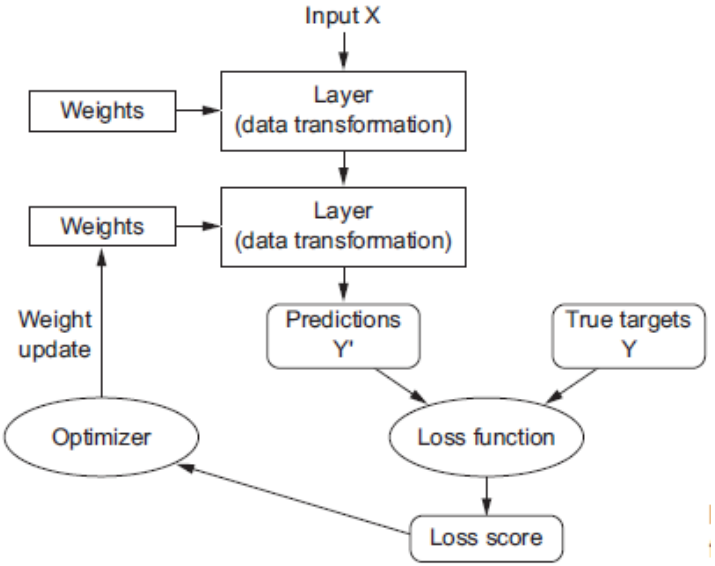


Neural Networks:

- **Neural Networks is about meaningfully *transform the data***



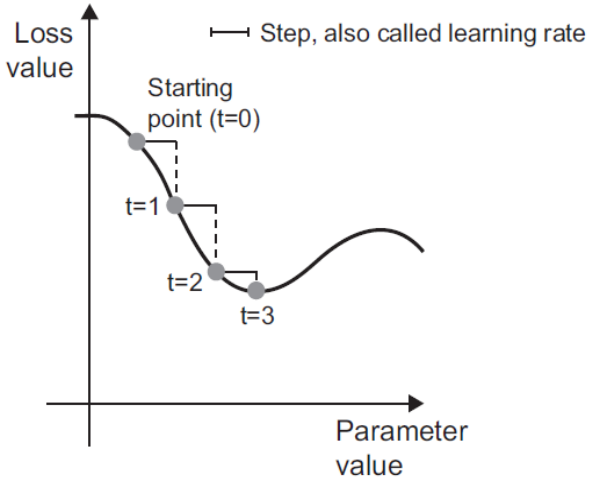


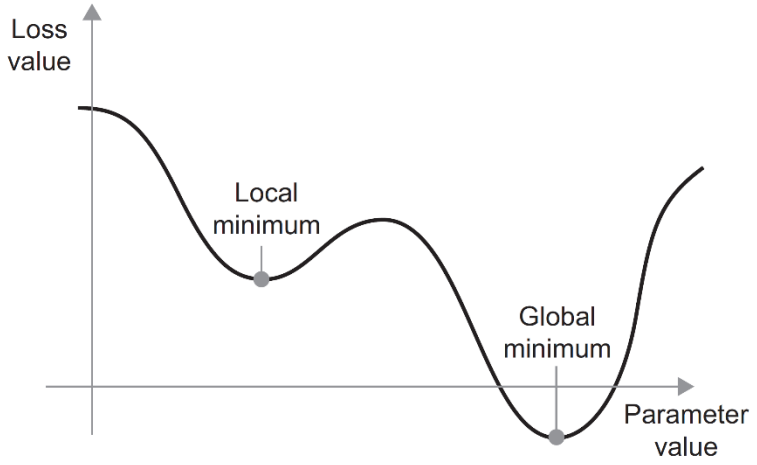


- *Training loop of DNN:*
  - ***Input data***, parameterize by ***weights*** (transform the data), predict and compute the ***loss score***, and update (***epochs***).

Deep Learning:

Artificial Neural Networks





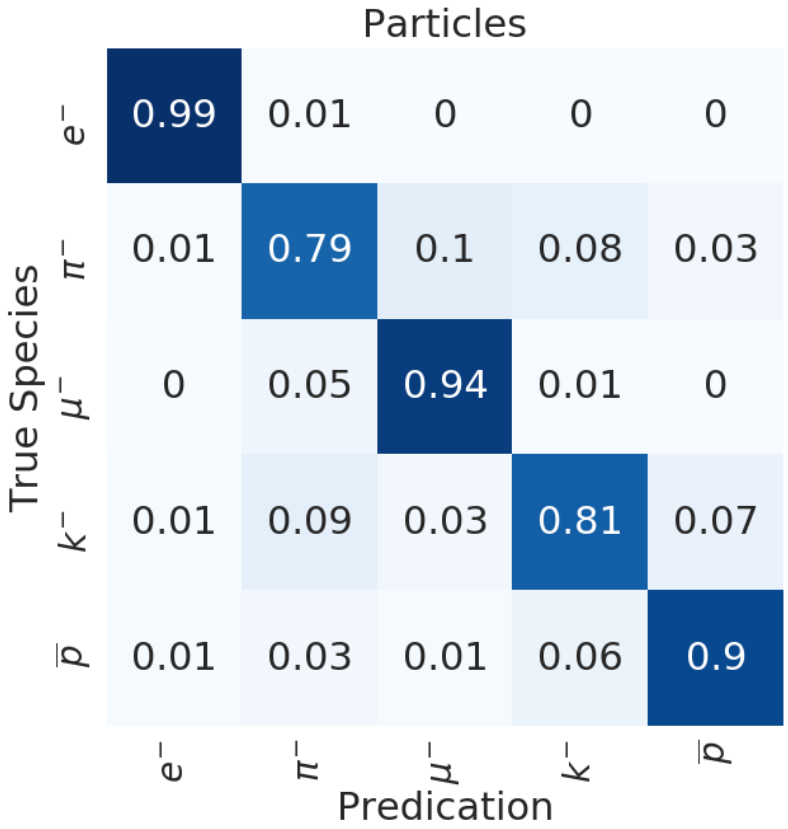
- The update step is guided by ***minimizing the loss function*** by calculating its ***gradient***.

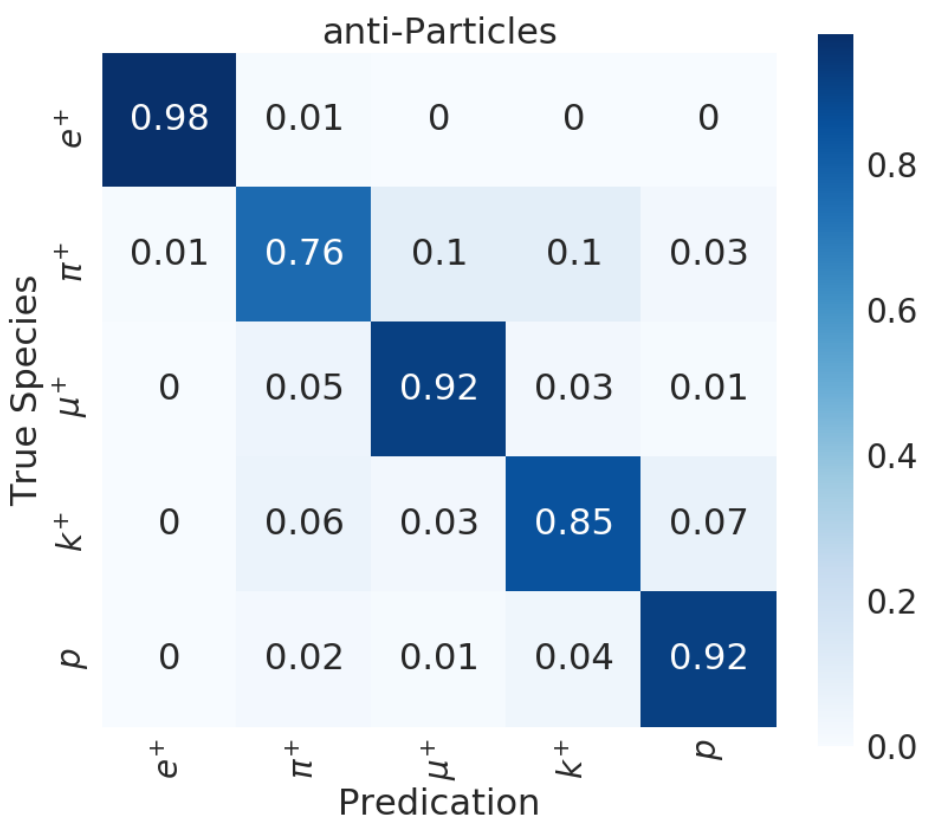
- ***Keras*** Python Package was used.

Hidden layers	*Activation function*	*Output layer*	*Learning rate*
3	***relu***	***softmax***	***0.001***

Evaluation metrics:

Confusion Matrices

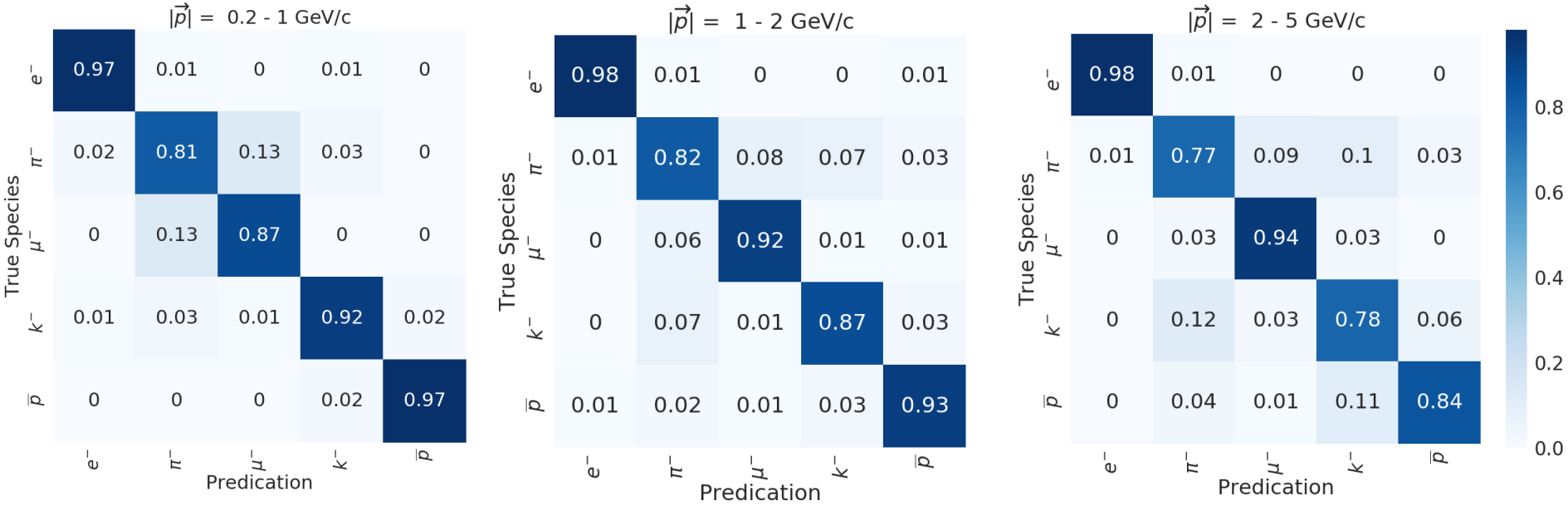




Evaluation metrics:

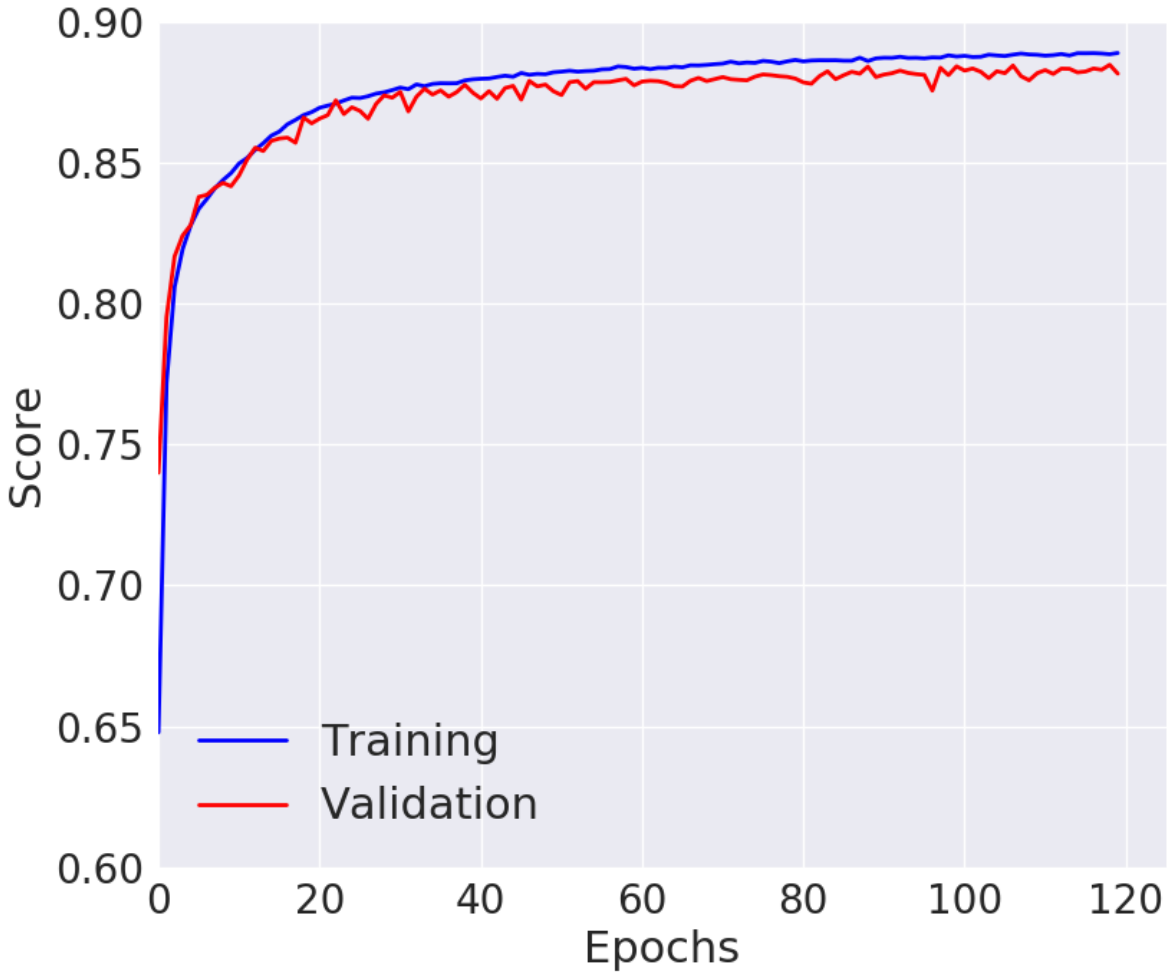
Confusion Matrices as function of momentum

For each momentum range 250K events generated (50k per particle type).



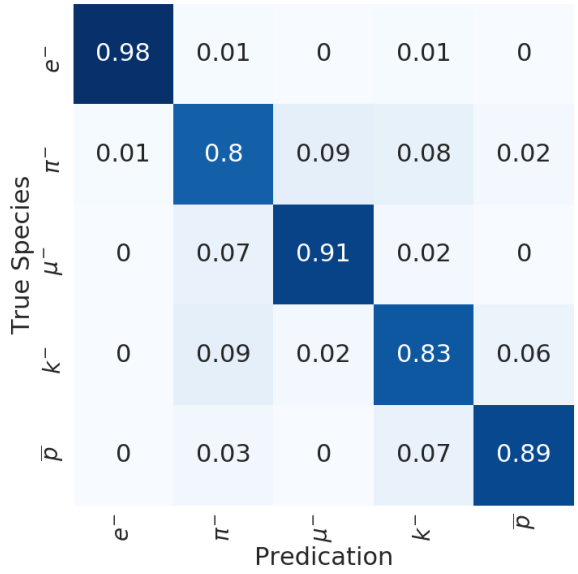
Evaluation metrics:

Validation Curve:

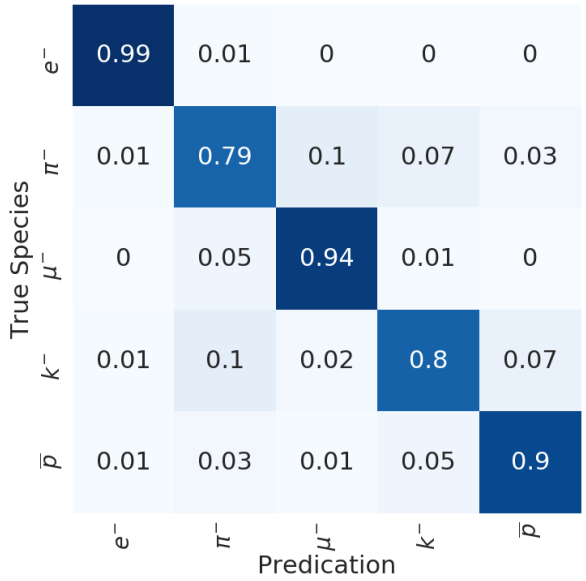


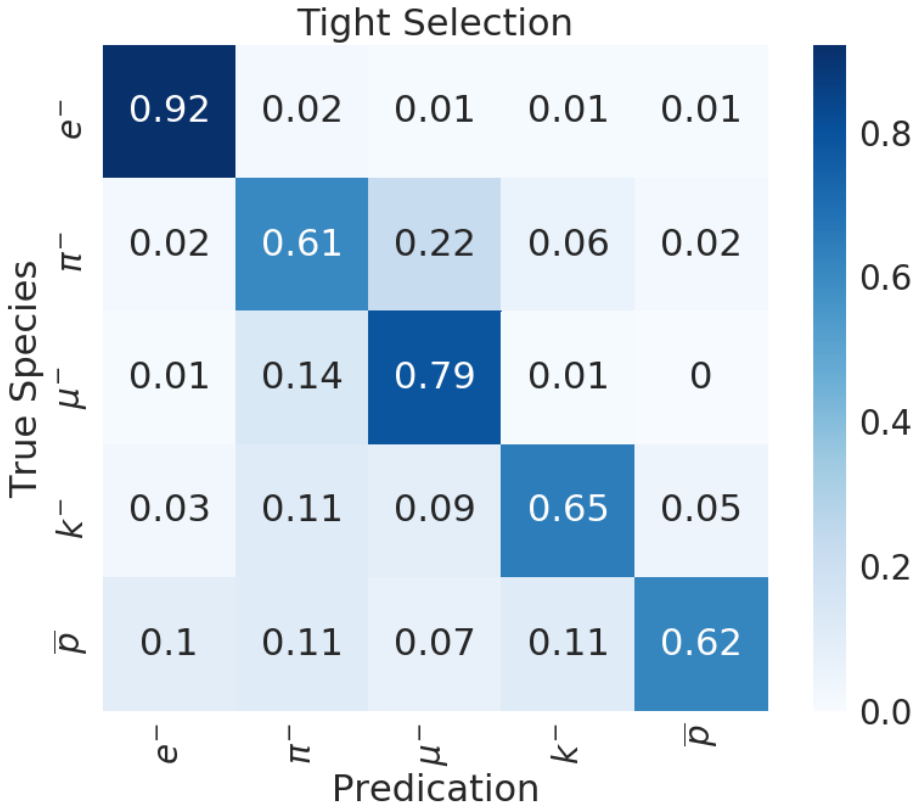
Comparison to Classical Algorithms:

RF



DNN





Conclusion

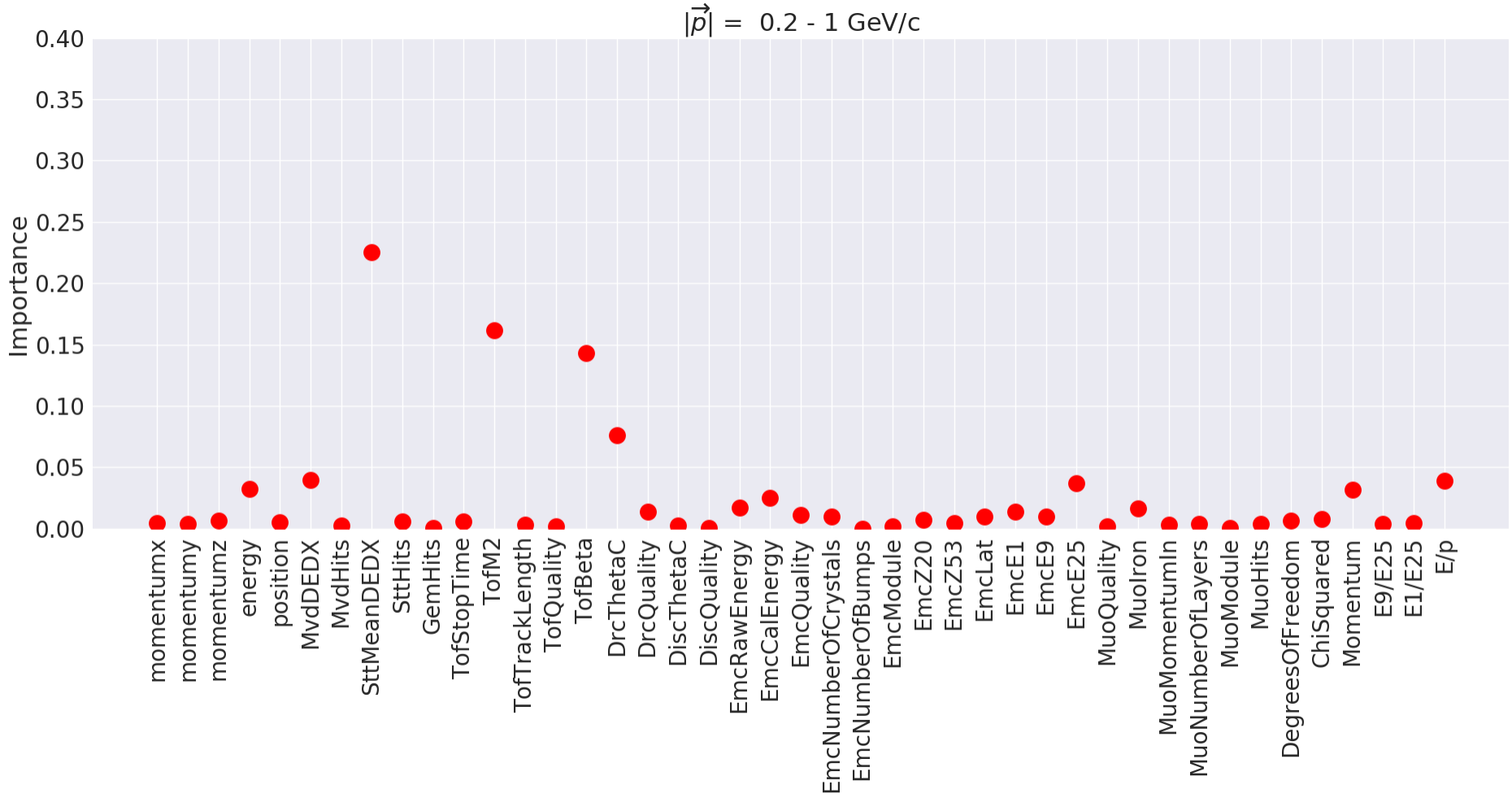
- *Decision trees* showed good performance in classifying charged particles over the specified momentum range.

- *Neural Networks* also showed good performance in the classifying task, but more training data can help to improve its accuracy.

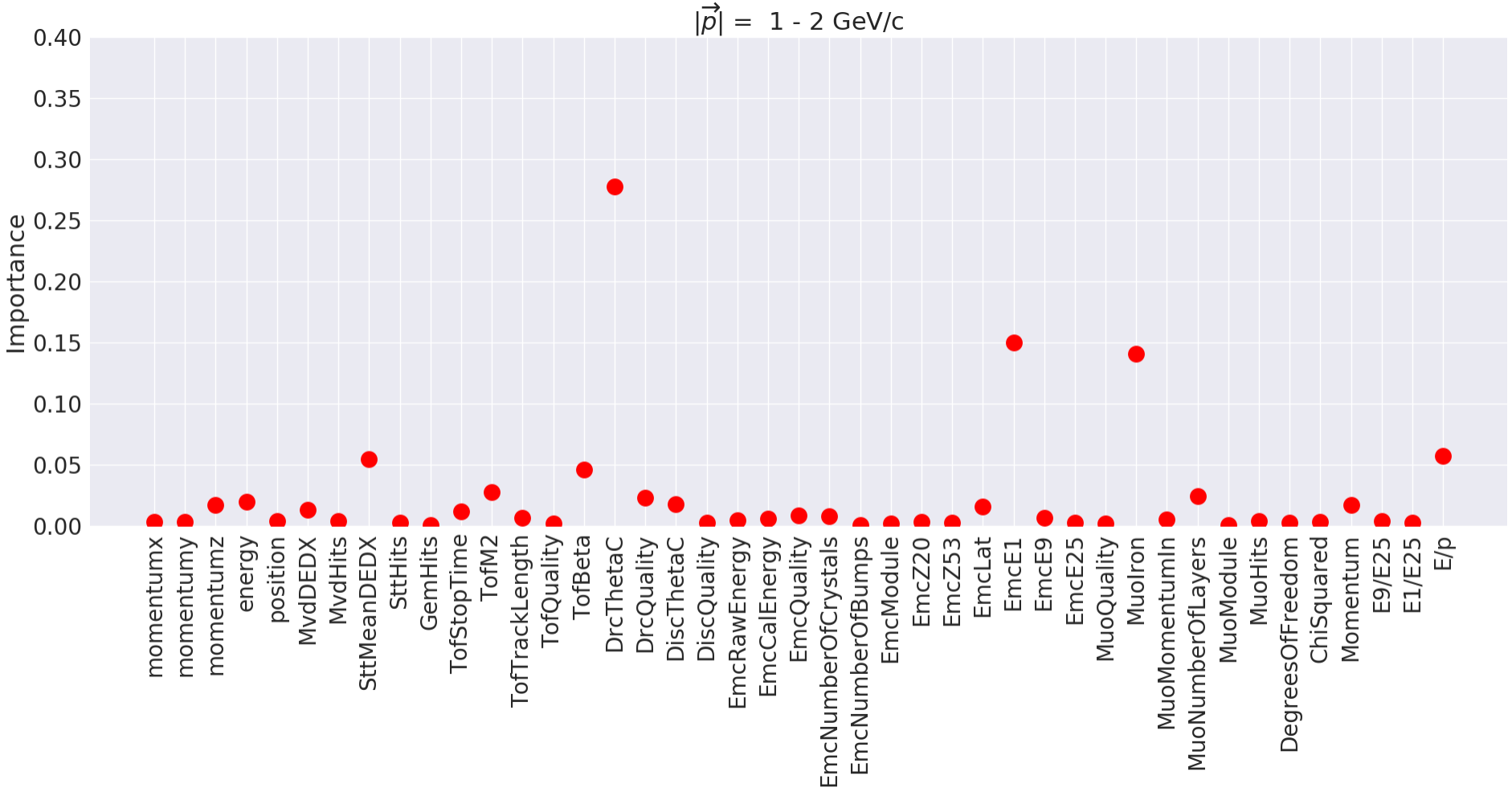
THANK YOU

BACK UP

Features Significance:



Features Significance:



Features Significance: