Machine Learning Techniques and Algorithms used in HEP

X. Tintin[1], E. Piedra[1], P. Sarzosa[1], R. Rocha[1], R. Trujillo[1], D. Mena[1]

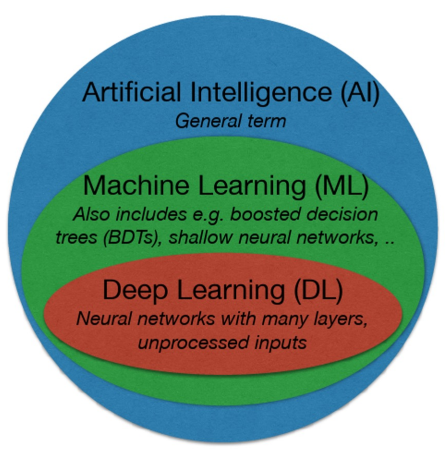
xavier.tintin@cern.ch, enrique_piedra2001@hotmail.com, pablo.sarzosa@epn.edu.ec, richard.rocha@epn.edu.ec, romina.trujillo@epn.edu.ec, david.mena06@epn.edu.ec

Facultad de Ingeniería de Sistemas, Escuela Politécnica Nacional, Quito, Ecuador

Introduction

This poster is a brief theoretical revision on present-day Machine Learning applications in HEP research, as an attempt to encourage collaboration between physicists and computer scientists to find innovative, optimal, and new computer efficient techniques to analyze CERN Open Data.

Possible Applications



- ✓ Object identification, classification, anomaly detection in big and noisy data sets
- ✓ Optimization of distributed computing, storage, and networks
- ✓ Ultra-fast on-edge inference under strict latency constraints

Why is it important to use ML in HEP?

- ✓ Machine learning offers an exciting opportunity to improve the calibration of nearly all reconstructed objects in HEP detectors
- ✓ ML significantly increases the efficiency in data measurement due to thorough data analysis, reducing data volume needed for CERN's break-through discoveries.

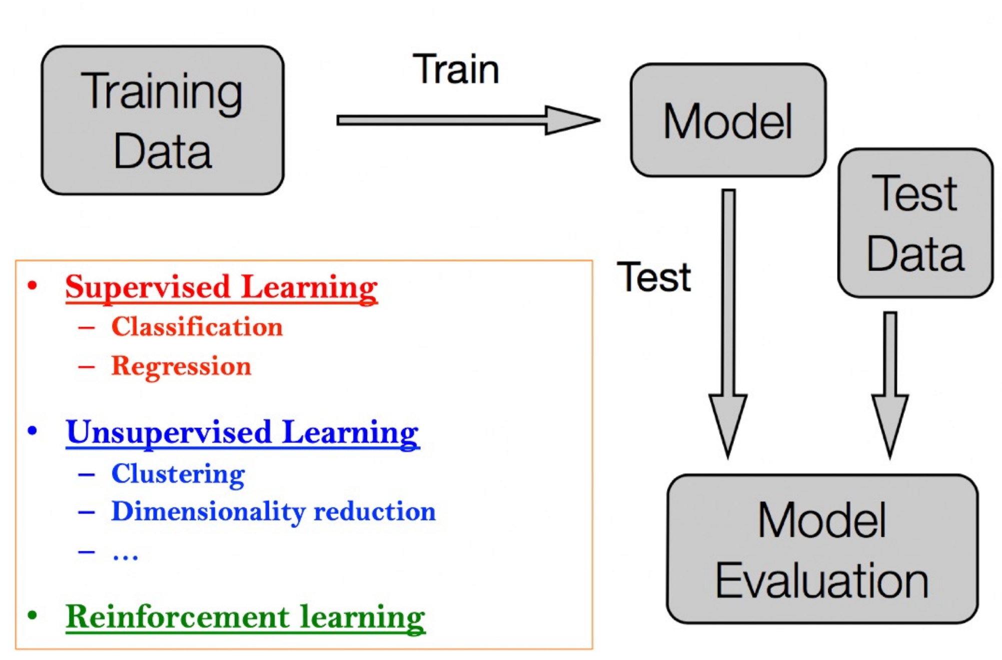
Objective: Identify and understand the best Machine Learning techniques used in HEP.

Methodology

- **ML platforms used for data analysis**

	TMVA	TensorFlow	Theano	Scikit Learn	R	Spark ML	VW	libFM	RGF	Torch
ROOT [T, C]	✓									
CSV [F]		✓	✓	✓	✓	✓	x	x	x	✓
libSVM [M]							x	✓	x	
VW [M]							✓			
RGF [M]									✓	
NumPy [R]		✓	✓	✓	✓	✓	x	x	x	✓
Avro [S, R]					✓	✓				
Parquet [S, C]					✓	✓				
HDF5 [S]		x	x	x						✓
R df [R]					✓					

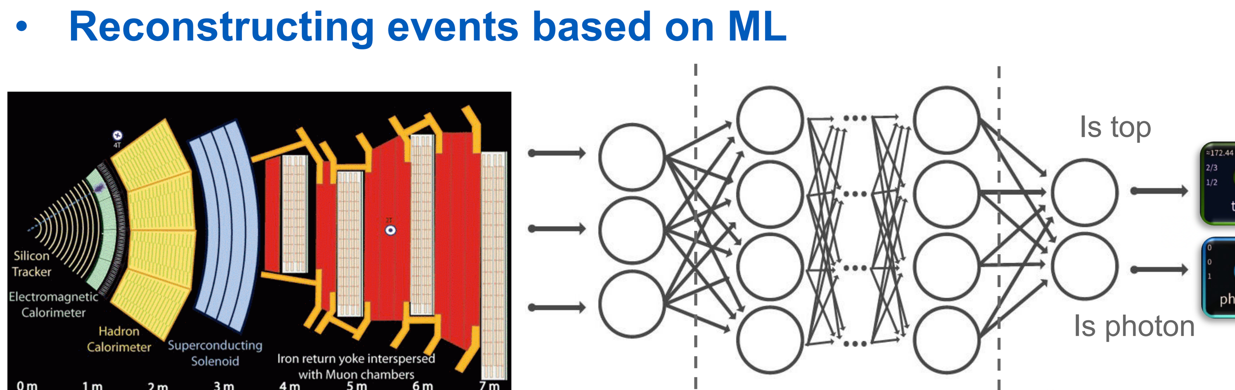
- **Machine Learning classification**



Various methods and techniques can be used to generate results closer to what is expected, such as:

- E2E
- CNN
- Forward Pass
- Backward Pass
- Stochastic Gradient Descent Algorithm,
- Feedforward Neural Network, etc.

- **Reconstructing events based on ML**



Real-world applications

- **The End-to-end approach**

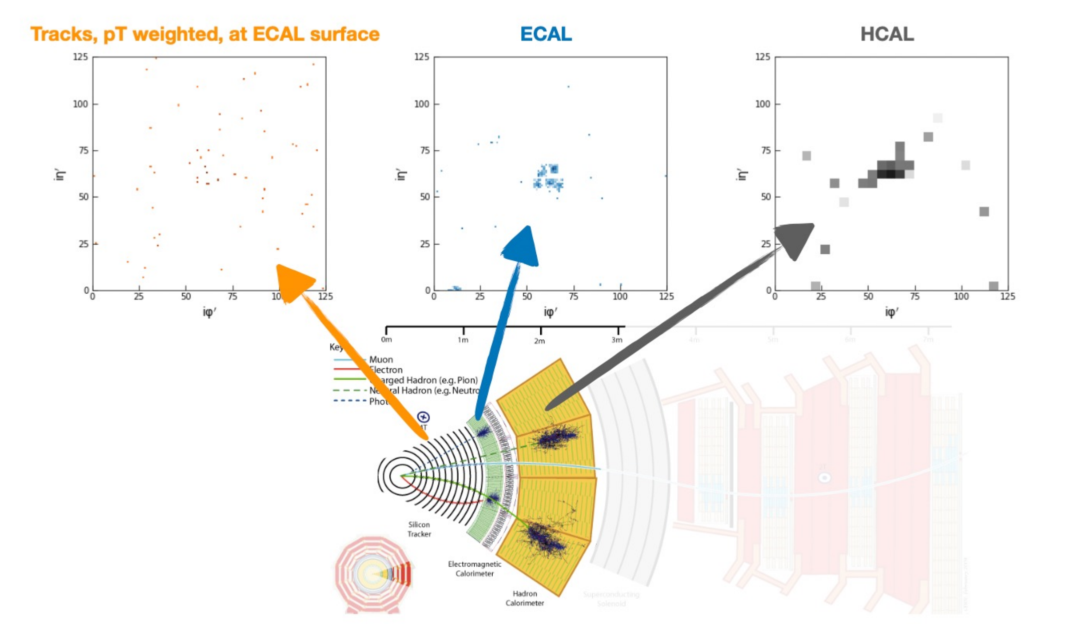


Figure A. The E2E approach consists of building a (CNN)-based jet or event discriminator by using the low-level detector inputs directly. [1]

- Each color of the circle represents a different type of particle. To transform physical objects to their representation as an image, the following must be considered: Resolution, Representation out of Range and Superposition.

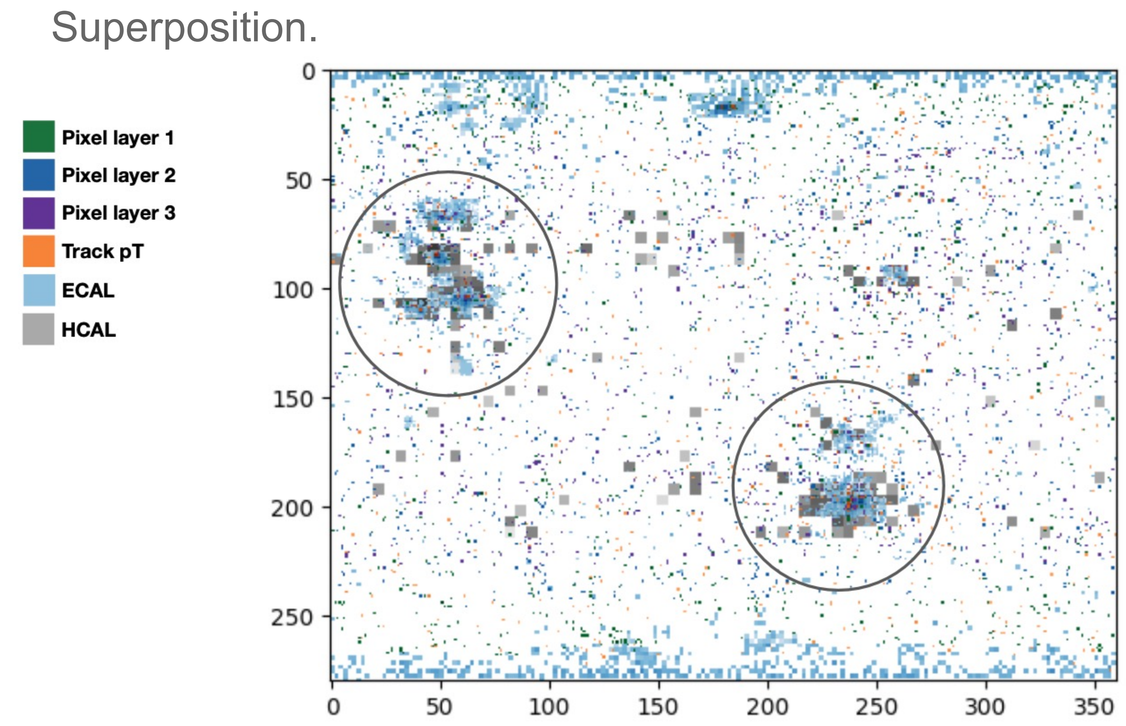


Figure B: E2E detector images including new tracker layers [1]

- **FNN and CNN**

When training the models with the same data but using an algorithm it is expected that different results will be obtained, when comparing FNN with CNN it is possible to observe that better results are obtained in the last 2 classes, in addition to the fact that the training time is much faster.

	CNN	FNN
True Label / Predicted Label	Total	Total
$Drell-Yan$	3955	3774
$W+jets$	3876	4049
$t\bar{t}$	4698	4710

Figure F: Results of the confusion matrix between CNN and FNN [2].

- **Neural Conditional Reweighting**

Reweighting is a simple but effective tool to improve an AI model. Neural Conditional Reweighting is an extended reweighting method that uses conditional cases, it also uses an algorithm for minimizing an error variable learned by the neural network that prejudices the prediction called bias.

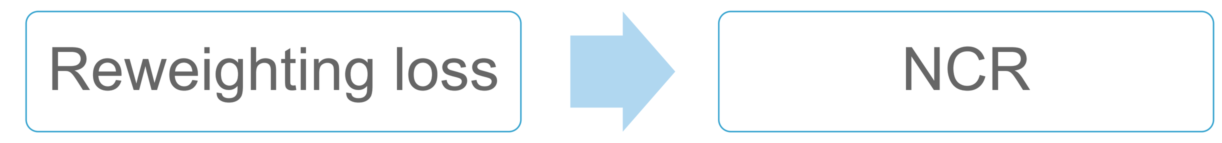


Figure C. NCR is derived from other methods

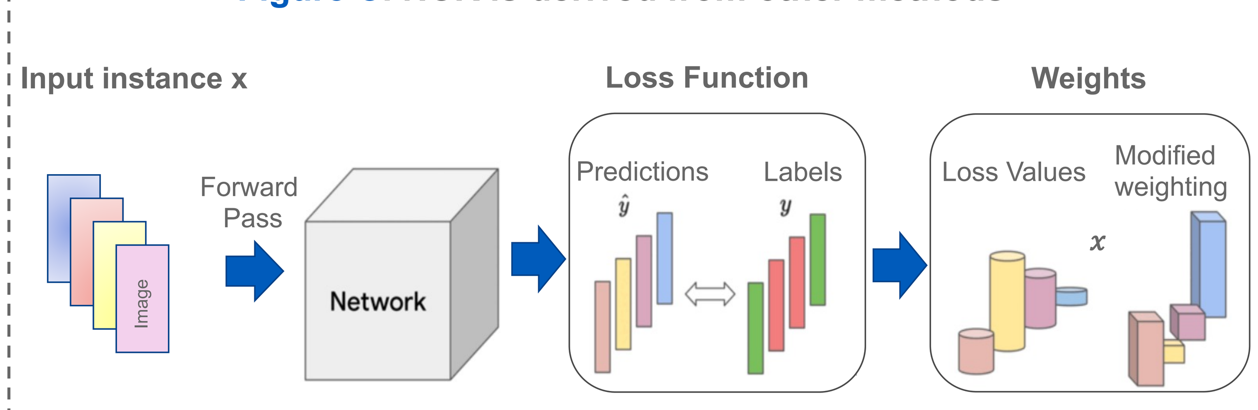


Figure D. Calculation process of reweighted loss values.

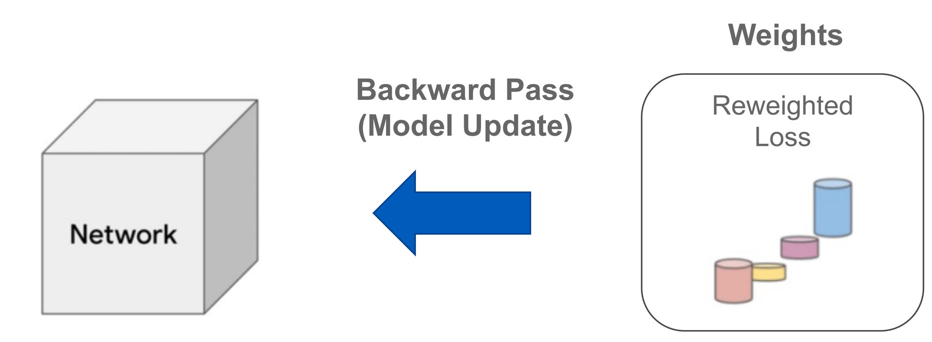


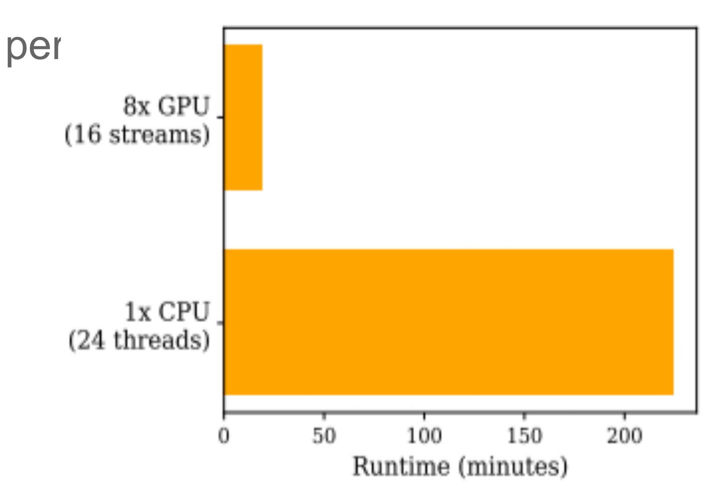
Figure E. Update of the Neural Network given the reweighted value vector.

Why NCR and not Reweighting?

- ✓ NCR is an approach to particle level truth information.
- ✓ It was leveraged by a custom loss function that allows the achievement of neural conditional reweighing. This is relevant for creating LHC simulation datasets, where full simulations are too computationally expensive to cover the entire phase space, and fast simulations can be used to fill in the gaps.
- ✓ As a specific example, it was applied neural conditional reweighting to the energy response of high-energy jets, which improved the modeling of physics objects in parametrized fast simulation packages.

GPU

One of the important findings for parallel data processing is the use of an 8 GPU NVidia GTX 1080 system, with two compute streams per … up to 12 times.



For a complete analysis with 270 million events, 100 GB of numerical data in a multi-GPU system. By using 8 NVidia GTX 1080 GPUs, 2 compute streams per device, we can reduce the analysis execution time by a factor of 12 times, compared to using multiple threads on the CPU alone. [3]

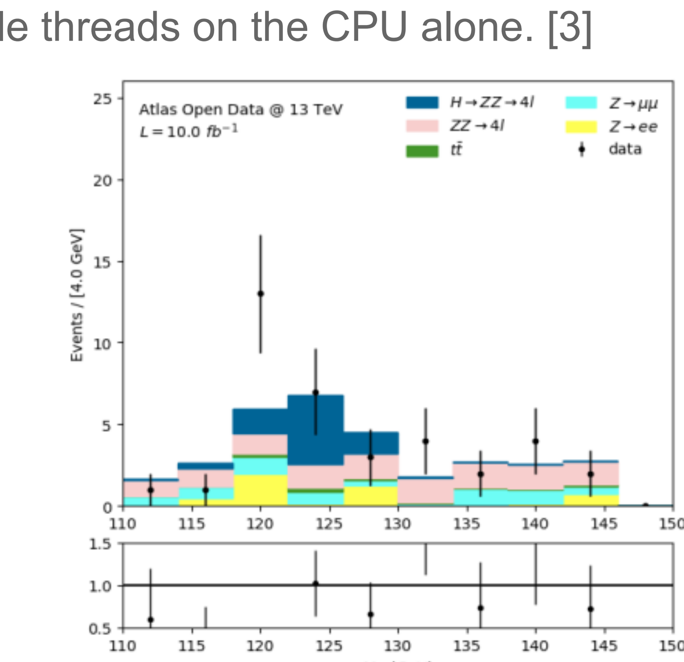


Figure F. Shows the mass of 4 invariant leptons of the implemented reference analysis of H → ZZ → 4l using Atlas Open Data.

Conclusions

- **Applying machine learning gives us the possibility of finding events that otherwise would have not been observed at the LHC.**
- **Transform physical objects to images gives us the opportunity to use various machine learning techniques to obtain their characteristics.**
- **NCR is particularly relevant in high-energy physics experiments for reweighting detector effects are conditioned on particle-level truth information.**
- **Multi-GPU architecture accelerates kernel processing, reducing response time by 12 times.**

References

[1] Madrazo, C. F., Heredia, I., Lloret, L., & de Lucas, J. M. (2019). Application of a Convolutional Neural Network for image classification for the analysis of collisions in High Energy Physics. In EPJ Web of Conferences (Vol. 214, p. 06017). EDP Sciences.

[2] Fernández Madrazo, C., Heredia Cacha, I., Lloret Iglesias, L., & de Lucas, J. M. (2017). Application of a Convolutional Neural Network for image classification to the analysis of collisions in High Energy Physics. arXiv e-prints, arXiv-1708.

[3] Pata, J., Dutta, I., Lu, N., Vlimant, J. R., Newman, A., Spiropulu, M., ... & Ruini, D. (2021). Data Analysis with GPU-Accelerated Kernels. In *40th International Conference on High Energy Physics* (Vol. 390, p. 908). SISSA.

Escuela Politécnica Nacional
Facultad de Ingeniería de Sistemas
Ciencias de la Computación