Computational Quantum Physics

Vasilis Tsioulos*

*Machine Learning Project: 2025

ABSTRACT

This study explores the performance of various classification models for analyzing Large Hadron Collider (LHC) event data, where the primary task involves distinguishing between signal events and background noise. To systematically evaluate prediction accuracy, we compare three distinct machine learning approaches.

1 Data preprocessing

The analysis begins by loading the experimental dataset from a CSV file. The data is organized into three primary categories: the signal events, low-level quantities (comprising 21 distinct features), and high-level quantities (containing 7 features). This initial segmentation allows for targeted processing of each data type according to its physical significance and feature space characteristics.

To ensure robust model performance, we implement an outlier removal procedure using the interquartile range (IQR) method. This statistical approach proved particularly effective for our particle physics data, identifying and removing approximately 1,000 measurements across both low-level and high-level feature sets while preserving the statistical properties of the distributions.

The next step is the visualization phase, where we examine all features through histogram analysis. These distributions reveal the data structure and help identify any high correlation in features. The subsequent correlation analysis provides critical insights into feature relationships, leading us to eliminate one strongly correlated feature ($r > 0.9$) that offered redundant information.

The final preprocessing step involves partitioning the refined dataset into training and testing subsets. We employ a 80-20 split to maintain proportional representation of signal and background events in both sets, our classification models train on representative data and evaluate on an unbiased test sample.

2 Classification

2.1 Artificial Neural Network

We implemented two Artificial Neural Network (ANN) models using PyTorch, designed to handle the different characteristics of our feature sets: one model processes the 21 low-level quantities while the other handles the 7 high-level quantities. Prior to training, both datasets underwent preprocessing, where

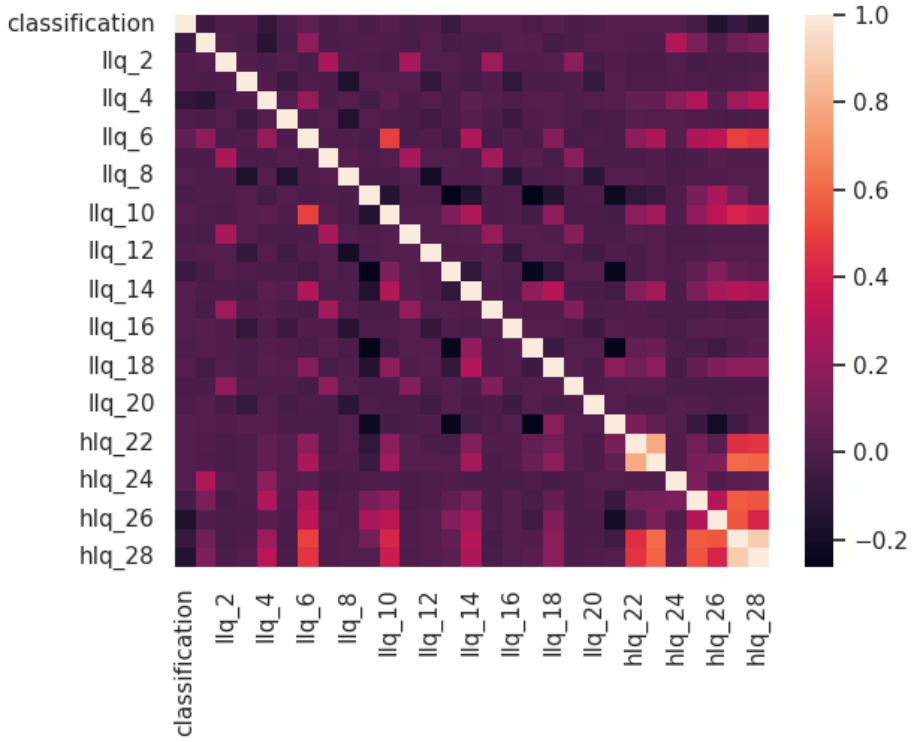

Figure 1. Correlation map

all features were scaled using StandardScaler and normalized to ensure consistent value ranges across variables. The normalization step proved critical for model convergence, particularly given the different magnitude scales present in the original LHC data.

The implemented ANN consists of four hidden layers between Linear and ReLU activation functions, followed by a final sigmoid output layer for binary classification. This architecture was designed to handle the complex features in LHC data, with the ReLU activations introducing non-linearity to capture patterns. The sequential structure progresses as: Linear (input transformation) $\rightarrow$ ReLU (non-linear activation) $\rightarrow$ Linear (feature combination) $\rightarrow$ ReLU $\rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow$ Linear $\rightarrow$ Sigmoid (probability output). The sigmoid final layer combined with BinaryCrossEntropy (BCE) provides [0,1] classification probabilities, enabling clear discrimination between signal and background events. This configuration balances model capacity with training stability for particle physics dataset.

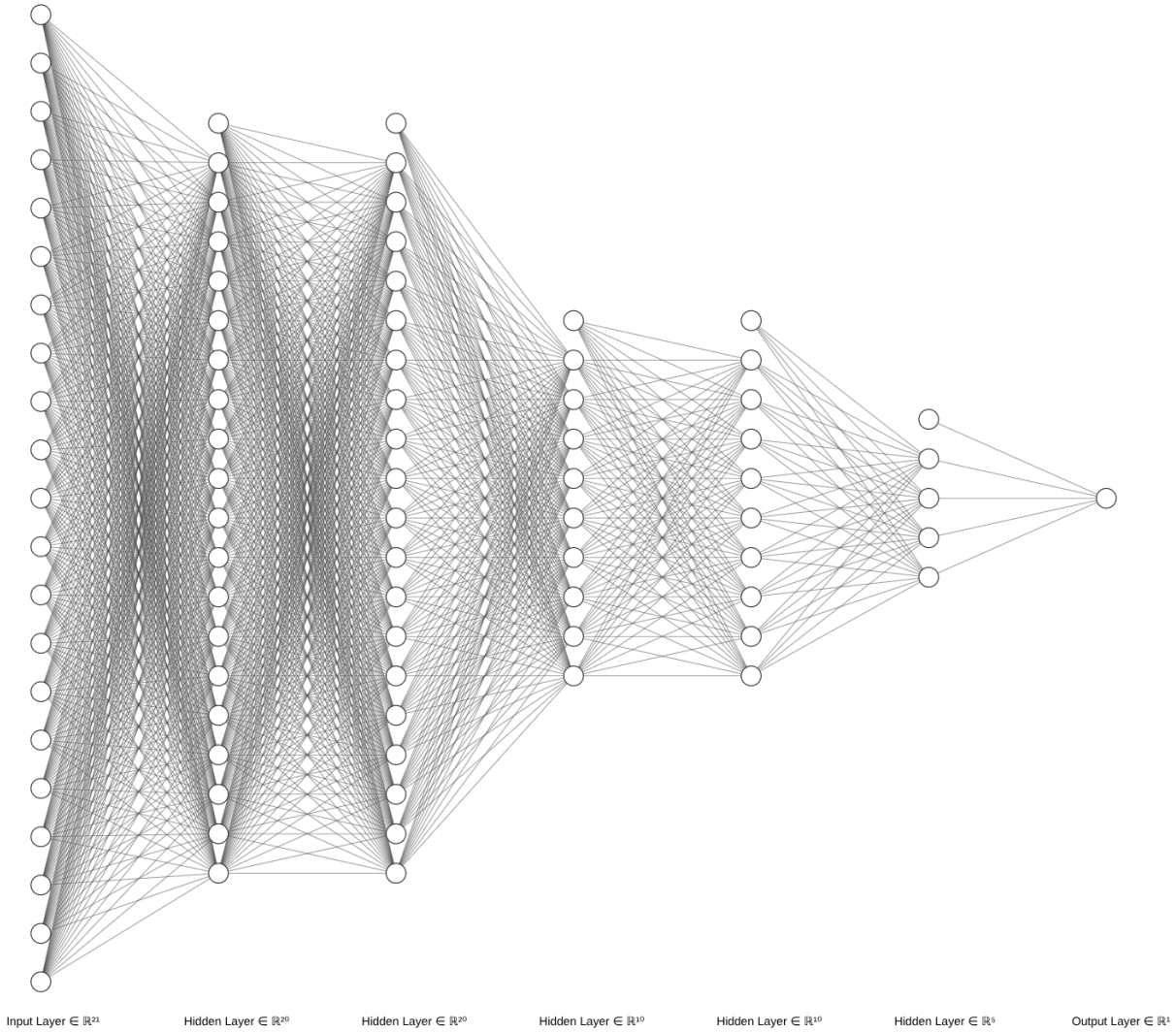

Figure 2. Artificial Neural Network layers

The accuracy and loss of both datasets were:

Data	R^2	Loss
Low-level	1.00	$4.06 \cdot 10^{-9}$
High-level	0.69	0.61

Table 1. Results of ANN for low and high level quantities

2.2 Scikit-learn

2.2.1 Random Forest Classifier

The confusion matrix of the Random Forest Classifier appears to be:

$$[low-level]\begin{bmatrix}740 & 0\\ 0 & 784\end{bmatrix}, \qquad [high-level]\begin{bmatrix}429 & 239\\ 179 & 533\end{bmatrix}$$

With accuracies of 1.0 and 0.69 respectively.

2.2.2 Logistic Regression

The confusion matrix of the Logistic Regression appears to be:

$$[low-level]\begin{bmatrix}740 & 0\\ 0 & 784\end{bmatrix}, \qquad [high-level]\begin{bmatrix}306 & 362\\ 130 & 582\end{bmatrix}$$

With accuracies of 1.0 and 0.64 respectively.

3 Conclusions

The comparative results reveal distinct challenges between feature categories. For low-level quantities, all tested models (including both ANN and scikit-learn classifiers) exhibited perfect training scores ($R^2 = 1.0$), indicating overfitting that suggests either are insufficient data preprocessing or excessive model complexity relative to feature dimensionality. Conversely, high-level quantities demonstrated more reasonable performance ($R^2 \approx 0.69$ across models), implying either the enhanced data cleaning to address underlying noise, or hyperparameter optimization using tools like GridSearchCV or TensorBoard's hyperparameter tuning.