Neural Networks for LHC Dataset

Piyush Chauhan
2021CS11010
cs1211010@iitd.ac.in

Sumit Yadav
2021PH10825
ph1210825@iitd.ac.in

Lokesh Varadaraj B
2022PH11858
ph1221858@iitd.ac.in

Dev Agrawal
2021PH10812
ph1210812@iitd.ac.in

Srikanta Bedathur
CS Dept. ,IIT Delhi
srikanta@cse.iitd.ac.in

Abhishek M. Iyer
Physics Dept.,IIT Delhi
abhishekiyerm@physics.iitd.ac.in

Abstract

In particle physics experiments, distinguishing signal events from background noise is crucial for accurate analysis. This project focuses on identifying hadronic tau decays amidst a dominant background of QCD (Quantum Chromodynamics) jets. Tau particles can decay into charged and neutral pions, and the likelihood of detecting a pion from a tau decay compared to QCD background is approximately one in a million. Accurately classifying these events is essential for studying tau leptons and probing physics beyond the Standard Model.

Keywords

Neural Networks, Convolution NN, Particle Net, FROCC, POND

ACM Reference Format:
Piyush Chauhan, Sumit Yadav, Lokesh Varadaraj B, Dev Agrawal, Srikanta Bedathur, and Abhishek M. Iyer. 2025. Neural Networks for LHC Dataset. In . , 5 pages.

1 Current Focus

Currently, the main focus is to enhance the classification performance by deploying various model architectures, from traditional Decision Trees, Neural Networks to Graphs and transformers. We also focus on feature selection and engineering to improve model interpretability and performance.

2 Dataset Description

Each entry in the dataset, derived from ROOT files, corresponds to either a Higgs boson decay signal or a background event from QCD jets. The Higgs boson production and decay processes are simulated using MAD-GRAPH. The resulting particles are further processed with PYTHIA, which handles parton showering and hadronization to generate the final-state particles. To mimic realistic jet reconstruction as done in the ATLAS experiment, detector effects are modeled using DELPHES, employing the CMS detector configuration provided within it. Jets are formed from DELPHES E-Flow objects using the anti-kT clustering algorithm with a radius parameter of R = 0.4. Only those jets with a transverse momentum above $50GeV$ and a pseudorapidity of, $|\eta| < 2.5$ are selected.

For the training purposes, the data has 70:30 ratio of background to signal.

It should be noted that few variable values (e.g., τ_{31} , r_1) provided by the simulator do not align with their corresponding literature values. Thus, we restrained ourselves from using those in any model.

3 Methodology

3.1 Boosted Decision Trees (BDT) and Neural Networks (NN)

The features chosen for training decision trees and neural networks were motivated and validated in the study *"A Framework for Finding Anomalous Objects at the LHC"* by Chakraborty, Iyer, and Roy (arXiv:1707.07084). They are listed below:

- N_T: Number of charged tracks in the jet. QCD jets exhibit high track multiplicity, while tau jets often peak at 1 or 3.
- θ_J: Hadronic energy fraction. Tau jets, primarily composed of charged pions, tend to peak at higher θ_J than QCD jets, which include neutral pions that decay into photons.
- λ_J: Defined as $\log(1 - p_{T_0}/p_{T_J})$, indicating energy carried by the leading subjet. Tau jets have narrower substructure, resulting in smaller λ_J.
- τ_N: N-subjettiness, with ratios like τ_{31} highlighting prong-like features; tau decays typically exhibit lower values due to 1- or 3-pronged topologies.
- e_N and $r_2 = e_3/e_2$: Energy correlation functions, where QCD jets—being multi-pronged—produce higher values of r_2 than single-pronged tau jets.

Standard idea is to use BDTs to construct vetoes that exclude standard jets (photons, electrons, taus, QCD jets) from anomalous regions. The study demonstrate that using a set of variables listed above, BDTs can effectively isolate corners of phase space occupied by specific jet types.

Architecture details(BDT) : XGBoost Classifier with 5 depth and 200 estimators performed well among other tried parameters.

Architecture details(NN) : Multi Layer Perceptrons with 3 hidden layers [256,128,64], dropout(0.3) and Batch Normalizations was used for classification. Input features were kept same as the BDTs input.

2025, Delhi,India

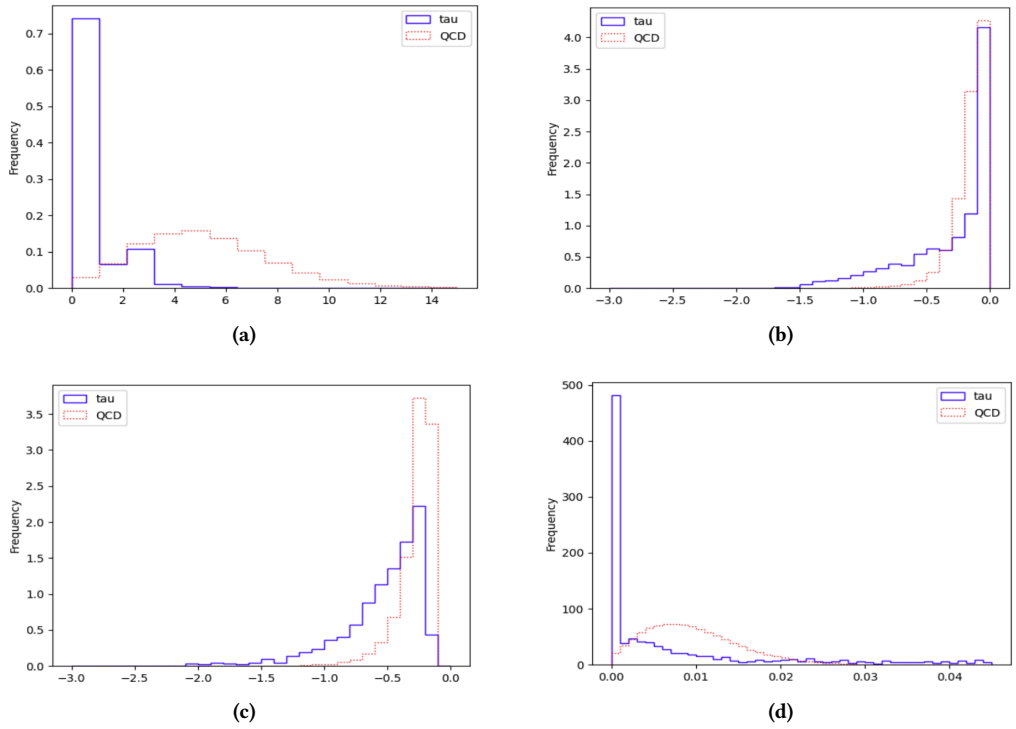


Figure 1: plots for (a) number of tracks(N_T) ,(b)θ_J , (c)λ_J and (d) energy correlation function ($ecrf_2$

3.2 Ensemble Convolution Neural Network (CNN)

Jets can also be represented as pixilated images derived from calorimeter energy deposits in η-ϕ space, and thus a CNN can be trained to classify between them. Following the work of *"Jet-Images – Deep Learning Edition"* by Oliveira et al. (arXiv:1511.05190v3). It is usually observed that for signal particles lie in a narrow region towards the center, while background is more dispersed over the whole plane.

We plan to use a ensemble of CNN and NN,motivated by the fact that CNN will capture particle level detail on which along with the jet level information available, NN may outperform previous models.

Image Prepossessing : Each image is of dimension 64×64. They are translated so that the leading subjet is at $(\eta, \phi) = (0, 0)$. The jet image is rotated so that the first principle component of the pixel intensity distribution is aligned along the vertical axis. The pixels color values are normalized according to $W_{p_T} = p_T(i)/max(p_T(i))$, where i runs over all the particles found in an event.

Architecture(CNN) : We use a lightweight CNN with three convolution layers (with ReLU and 2×2 max pooling) to extract features from images consisting of a few bright pixels on a dark background. The number of filters increases from 4 to 16, as no complex feature (line, spiral etc.) appears in the images. Flattened features pass through two fully connected layers with dropout for regularization. A Sigmoid-activated output layer produces binary class probabilities. These probabilities are then fed to a Neural Network along with jet level features stated in Section 4.1.

3.3 ParticleNet(PartNet) : Graph Based Model

Image-based jet representations have two key drawbacks. While they can capture all information from calorimeter measurements, incorporating additional details from reconstructed particles—like particle type—is difficult due to non-additive properties being combined in the same pixel. Additionally, jet images are extremely sparse: our jets have 5–7 particles in signal and 10-20 particles in background while image have 4096 pixels, leaving over 95% empty. This sparsity leads to computational inefficiency when using CNNs.

Thus, we represented jet as an unordered sets of constituent particles, similar to the work of *"Jet Tagging via Particle Clouds"* by Huilin Qu et al. (arXiv:1902.08570v3).This model is different from the previous models from the fact that it user particle level features rather than jet level features. The model uses the EdgeConv operator to capture local relational information between particles within a jet. The EdgeConv operation for each point x_i is defined as:

$$x_i' = (1/k) \sum_{j=1}^{k} h_\Theta(x_i, x_{ij})$$

$$h_\Theta(x_i, x_{ij}) = \bar{h}_\Theta(x_i, x_{ij} - x_i)$$

where $x_i \in \Re^F$ denotes the feature vector of the point x_i, and $\{i_1, \ldots, i_k\}$ are the indices of the k nearest neighboring points of x_i.

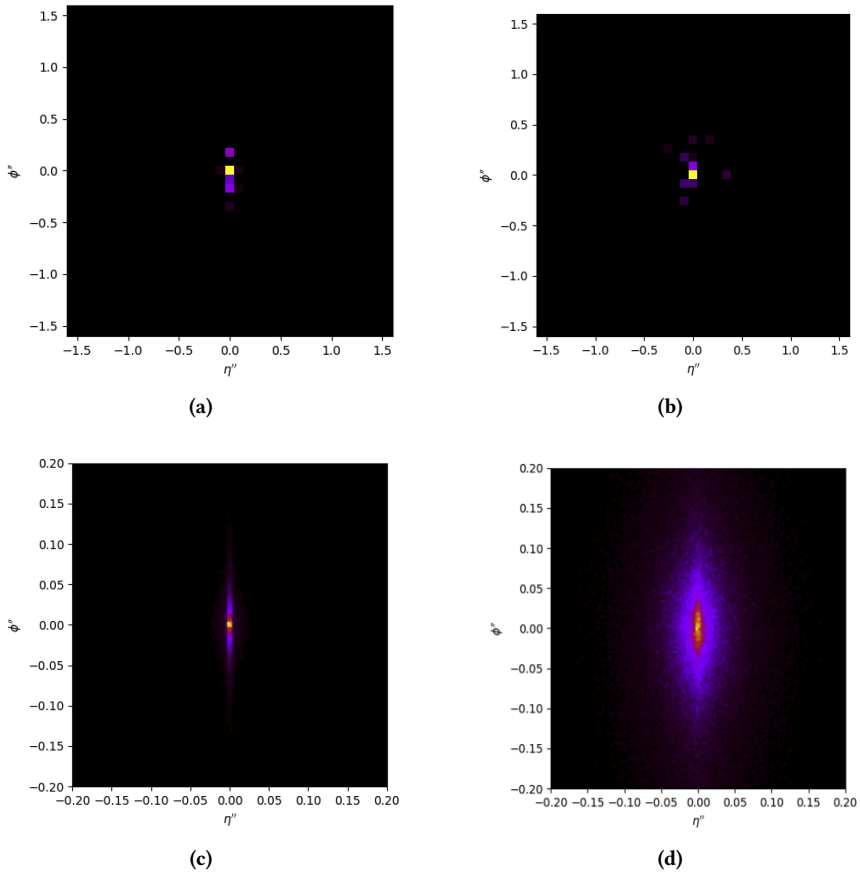


Figure 2: Image plots for (a) single Tau event ,(b) single QCD event , (c) all Tau events (d)all QCD events combined for CNN model

The edge function $h_\Theta : \mathfrak{R}^F \times \mathfrak{R}^F \rightarrow \mathfrak{R}^{F'}$ is a learnable function parametrized by a set of parameters Θ.

EdgeConv block is characterized by two hyperparameters, the number of neighbors k, and the number of channels C = (C1,C2,C3), corresponding to the number of units in each linear transformation layer.

Dataset used : For this model, we used the dataset which include particle level information. Variables used for training and testing are listed in **Table 1**.

Architecture : ParticleNet architecture is shown in the image below, consists of four EdgeConv blocks. The number of nearest neighbors k is 4 for all blocks, and the number of channels C for each EdgeConv block is (16, 16,16), (16, 16,16), (32, 32,32), (64, 64,64) respectively. The AdamW optimizer, with a weight decay of 0.0001, is used to minimize the cross entropy loss. One-cycle learning rate (LR) schedule is used in training.

3.4 Fast Random projection-based One-Class Classification(FROCC)

Apart from the literature models, we tried FROCC model as well. As our problem fits well in the description of anomaly detection and one class classification, using FROCC was a natural choice. FROCC main idea is to train on instances from a single (positive/negative) class and identify whether a new instance belongs to this known class or is an outlier.

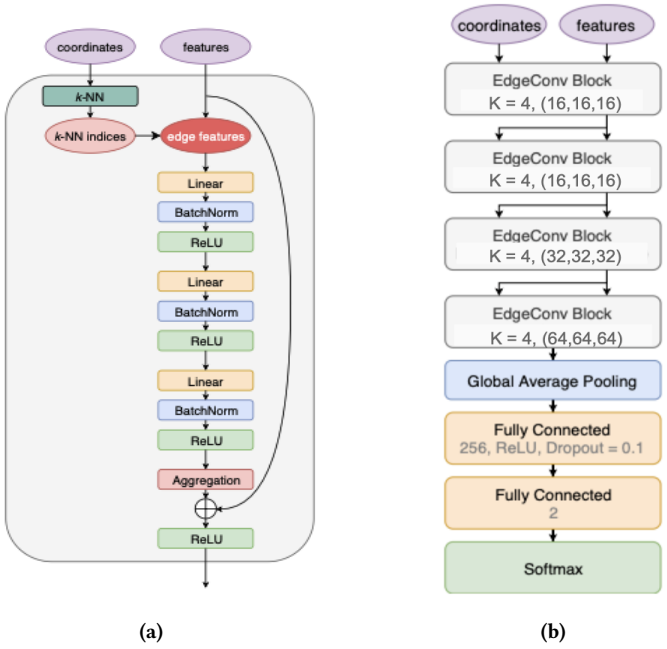
Hyperparameter of FROCC are :

- **Number of Random Projections (d):** Number of directions used.
- **Density Threshold (ϵ):** A small value ($\epsilon \in [0.01, 0.1]$) determines the minimal fraction of data to include in the interval.
- **Kernel Type (Optional):** linear or RBF kernel for non-linear data separation.

A brief of the algorithm is as follows, FROCC begins by projecting high-dimensional data onto multiple random 1-D directions. For each direction, it identifies dense intervals where most normal data points lie. These intervals define regions considered "normal" in the projected space. During test, the new point is projected onto all

Variable	Definition
$\Delta\eta$	difference in pseudorapidity between the particle and the jet axis
$\Delta\phi$	difference in azimuthal angle between the particle and the jet axis
$\log(p_T)$	logarithm of the particle's total momentum
$\log(E)$	logarithm of particle's total energy
$\log(p_T/p_T(jet))$	logarithm of the particle's pT relative to the jet pT
$\log(E/E(jet))$	logarithm of the particle's E relative to the jet E
ΔR	angular separation between the particle and the jet axis ($\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$)

Table 1: Features used in ParticleNet Model



(a) **(b)**

Figure 3: (a) Edge Convolution block and (b) Full Particle Net model

directions and checked against these intervals. If it falls within the dense region in all projections, it is classified as normal; otherwise, it is an anomaly.

Data : Same data as Neural Network/BDT was given for training
Architecture : Base FROCC model was used with $d = 2000$ and $\epsilon = 1e^{-4}$.

3.5 PrOjected Neural Density Estimation (POND)

We tried POND (PrOjected Neural Density Estimation) model as well. Brief about POND, it operates by learning a set of projection vectors that map high-dimensional input data onto one-dimensional subspaces. For each of these projection vectors, the model considers a set of fixed pivot points ,representative samples from both the positive and negative classes—against which it evaluates the similarity of a given input point. This similarity is quantified using a kernel applied to the difference between the projection of the input and the projection of each pivot point, resulting in a density-like score that reflects how close the input is to different class regions along each projection. By aggregating these scores across multiple projections and pivots, POND produces a final classification score.

Different from FROCC, POND has a larger number of hyper-parameters to tune. A list of the important ones is as follows,

- **Number of vectors(d_in)**: Number of projection vectors to take.
- **Number of Kernels(k)** : Number of pivot points per class used for scoring.
- **Scoring(score)** : Type of kernel used in scoring function (e.g., Gaussian, linear).
- **Kernels Std Dev. (std_dev)** : Standard deviation used in the kernel for scoring.

Data : Same data as Neural Network/BDT was given for training
Architecture : Base POND model was used with following hyper-parameters, $vectors = 5$, $scoring = RBF$, $stddev. = 1e^{-4}$ and $kernels = 2000$.

4 Results

Below table provides a summary of various models describe so far:

Model	Accuracy	ROC-AUC	F1-score
BDT	96.10	0.9424	0.9180
NN	96.05	0.9602	0.9416
CNN+NN	**96.15**	0.9322	**0.9434**
FROCC	91.43	0.9149	0.9095
FROCC+NN*	96.86	0.9609	0.9572
POND	93.47	**0.9765**	0.9232
PartNet**	**97.47**	**0.9792**	**0.9593**

Table 2: Results for various models,
***training NN after FROCC leads to data leakage**
****PartNet uses different representation of data and hence shown different**

From the table it is clear that by using only jet level information BDT and CNN seems to perform equivalently while if we use particle level information, PartNet outperforms any other model used. POND model also gives promising results in terms of AUC score. More experiments and scoring functions need to be tested for POND as well.

5 Future Work

We are also trying the current state of the art model, Particle Transformer(ParT) by Huilin Qu et al. (arXiv:1511.05190v3). PartNet and ParT are not tested on tau dataset by the authors and thus we would like to benchmark these model performance.

Received 8 May 2025