Multi Modal AI Hackathon

Multi Modal AI Hackathon Report: Uncovering Physics Anomalies with Deep Learning

TEAM NAME: MasterMinds

TEAM MEMBER: MIDHUN S, AADITHYA R, ROHITH PREM S, DEVASANJAY N, SRIVATSAN V

ABSTRACT:

This report details the comprehensive analysis of high-energy collision data as part of the National Students' Space Challenge 2025. The primary objective was to leverage machine learning and deep learning techniques to classify particle jets and identify anomalies that could signify new physics beyond the Standard Model. We explored both jet image data using Convolutional Neural Networks (CNNs) and tabular data with traditional machine learning models. The project culminates in a comparative analysis of these modalities and the development of a CNN-based autoencoder for anomaly detection. Our findings demonstrate the power of deep learning in extracting subtle patterns from complex physics data, with the image-based CNN classifier showing strong predictive performance and the autoencoder successfully identifying anomalous jet events.

Introduction:

High-energy particle collisions, like those at the Large Hadron Collider (LHC), produce streams of particles known as jets. These jets are crucial for understanding the fundamental forces of nature. The study of their properties can validate the Standard Model of particle physics and, more excitingly, reveal rare processes or new particles that lie beyond it.

The objective of this analysis is to apply advanced machine learning techniques to a dataset of particle jets. We aim to:

1. Perform a thorough exploratory data analysis and preprocess the data for modeling.
2. Develop a robust Convolutional Neural Network (CNN) to classify jets based on their image representations.
3. Train and evaluate various machine learning models on high-level tabular features derived from the jets.
4. Conduct a comparative study to determine which data modality—image or tabular—provides greater predictive power.
5. Build a CNN autoencoder to perform anomaly detection, identifying jets that deviate from the norm and could represent new physical phenomena.

1. Data Exploration and Preprocessing

The dataset consists of two primary components: a collection of jet images and a corresponding set of tabular data containing high-level physical features.

1.1 Exploratory Data Analysis (EDA)

Initial exploration involved examining the dimensions and structure of both datasets.

- **Jet Images:** We visualized a sample of jet images to understand their structure, noting the sparse nature of the energy deposits within the image grid.

- **Tabular Data:** We analyzed the statistical distributions of each feature in the tabular dataset. This included plotting histograms and calculating descriptive statistics (mean, standard deviation, etc.) to understand the scale and spread of the data.

Dataframe

```
First 5 rows of the Dataframe:
   j_ptfrac         j_pt     j_eta      j_mass   j_tau1_b1  j_tau2_b1  \
0       1.0  1015.080811  0.140886   76.495880   26.900795  18.509501
1       1.0  1050.538086  0.042137   81.212212   76.931221  20.120834
2       1.0  1017.096436  0.983565   32.378082   12.919186   8.439772
3       1.0  1185.474731 -1.008881  219.368683  205.640121  93.420189
4       1.0  1133.109131 -0.888515   74.178467   42.171444  19.598909

   j_tau3_b1  j_tau1_b2  j_tau2_b2  j_tau3_b2  ...  j_m2_b2_mmdt  \
0  14.211438   5.379379   2.994760   1.283431  ...      0.066155
1  11.993080   5.947691   0.942917   0.395835  ...      0.038634
2   5.244069   0.995097   0.303662   0.267116  ...      0.048102
3  52.173637  39.886662  13.379012   4.474773  ...      0.080403
4  18.466692   4.374649   1.192268   1.081973  ...      0.031107

   j_n2_b1_mmdt  j_n2_b2_mmdt  j_mass_trim  j_mass_mmdt  j_mass_prun  \
0      0.290213      0.178550    60.386185     2.875798    10.545803
1      0.240085      0.088289    81.212212    81.194130    78.364342
2      0.251524      0.146706    32.339310     4.691502     7.687650
3      0.351959      0.215218   219.368683   219.368683   219.368683
4      0.315460      0.137285    70.019073    66.888763    60.693222

   j_mass_sdb2  j_mass_sdm1  j_multiplicity  target
0    76.495880     0.000004            54.0  b'j_q'
1    81.194130     0.465613            44.0  b'j_w'
2    31.860289     0.000005            26.0  b'j_w'
3   219.368683   152.910950            77.0  b'j_t'
4    68.833763     0.482859            64.0  b'j_g'

[5 rows x 54 columns]

Shape of the 'jetimage' data: (10000, 100, 100)
This image data is stored separately in the 'jet_images' variable.
```

1.1(i) Preprocessing Steps

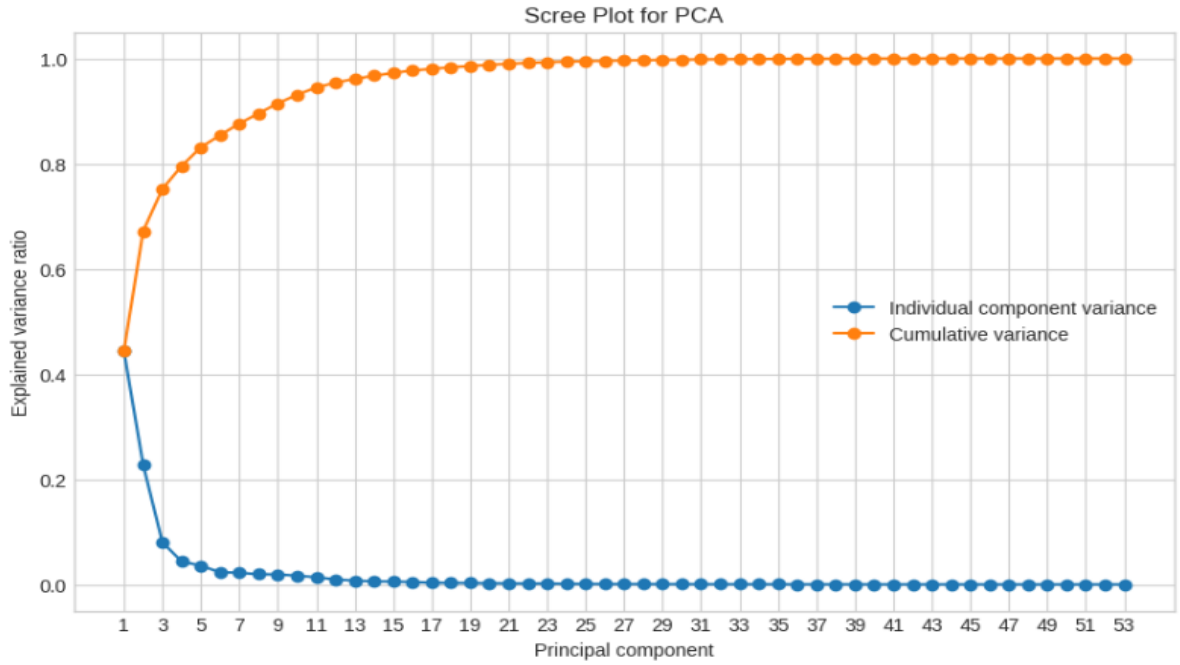
To prepare the data for modeling, the following steps were taken:

- **Image Data:**
    - **Normalization**: Pixel intensity values for each image were scaled to a range of [0, 1] to ensure stable and efficient training of the neural network.
    - **Reshaping:** The image arrays were reshaped to include a channel dimension, making them compatible with the input layer of our CNN.
- **Tabular Data:**
    - Standard Scaling: All features were standardized using StandardScaler, which transforms the data to have a mean of 0 and a standard deviation of 1. This is crucial for models sensitive to feature scales.
    - Principal Component Analysis (PCA): We applied PCA to reduce the dimensionality of the tabular data. We analyzed the explained variance ratio to select the optimal number of principal components that capture the majority of the information in the data.

PCA and Scree Plot

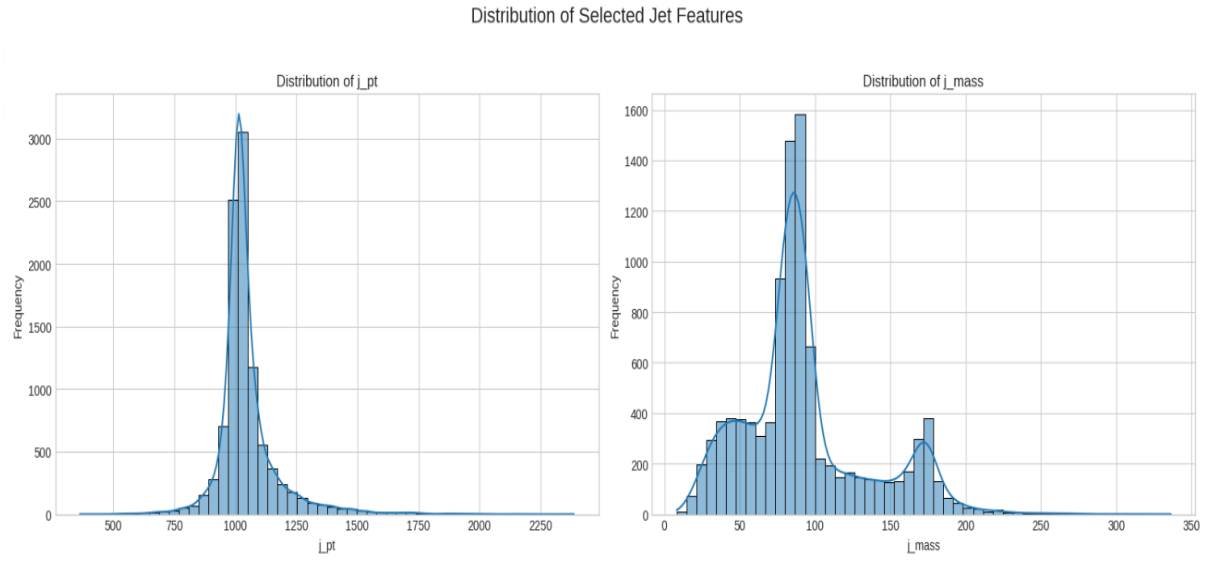
```
Performing PCA:

Explained variance by each principal component:
  PC-1: 0.4440 (Cumulative: 0.4440)
  PC-2: 0.2276 (Cumulative: 0.6716)
  PC-3: 0.0801 (Cumulative: 0.7517)
  PC-4: 0.0437 (Cumulative: 0.7954)
  PC-5: 0.0356 (Cumulative: 0.8309)
  PC-6: 0.0230 (Cumulative: 0.8540)
  PC-7: 0.0221 (Cumulative: 0.8761)
  PC-8: 0.0193 (Cumulative: 0.8954)
  PC-9: 0.0190 (Cumulative: 0.9144)
  PC-10: 0.0164 (Cumulative: 0.9308)
```



Data Summary

```
--- Dataset Summary ---
Number of rows (records): 10000
Number of tabular features: 53
Image size (Height x Width): 100x100
Number of classes: 5
Class labels: [b'j_g', b'j_q', b'j_t', b'j_w', b'j_z']
```



2. Jet Image Classification using a CNN

We designed and trained a Convolutional Neural Network (CNN) to classify jets directly from their image representations.

2.1 Model Architecture

Our CNN architecture consisted of the following layers:

- **Convolutional Layers:** To extract spatial features like the shape and intensity of energy deposits.
- **Activation Layers (ReLU):** To introduce non-linearity.
- **Pooling Layers (MaxPooling):** To reduce the spatial dimensions and computational load.
- **Flatten Layer:** To convert the 2D feature maps into a 1D vector.
- **Dense Layers:** For final classification.
- **Output Layer:** A single neuron with a Sigmoid activation function for binary classification.

Model Summary

Model: "sequential"

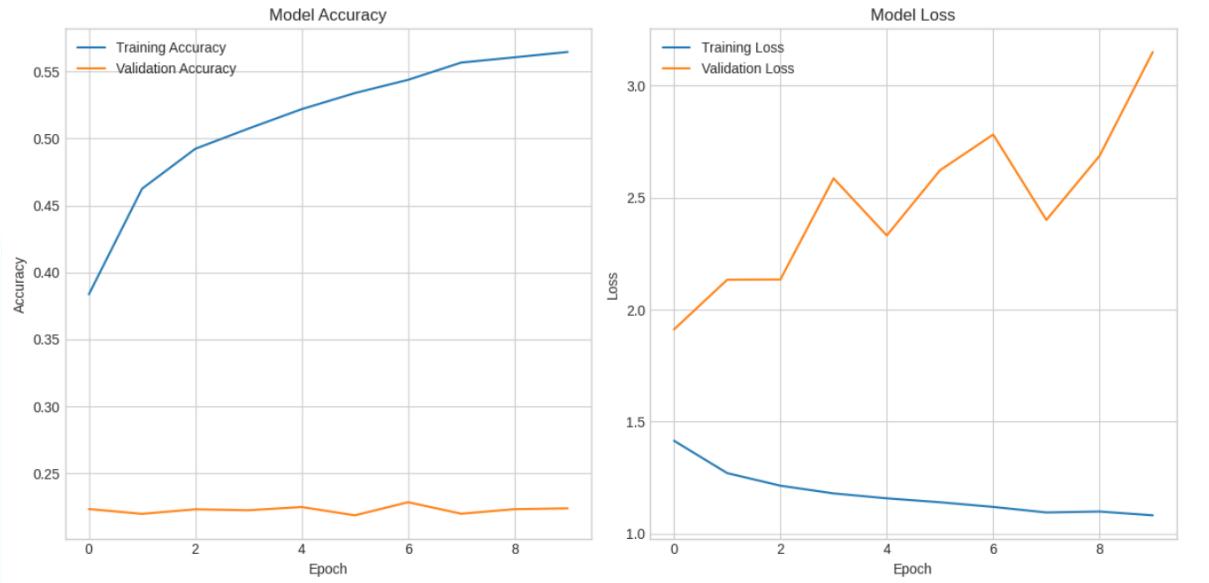
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 100, 100, 32)	320
max_pooling2d (MaxPooling2D)	(None, 50, 50, 32)	0
conv2d_1 (Conv2D)	(None, 50, 50, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 64)	0
conv2d_2 (Conv2D)	(None, 25, 25, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 128)	2,359,424
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 5)	325

Total params: 2,460,677 (9.39 MB)
Trainable params: 2,460,677 (9.39 MB)
Non-trainable params: 0 (0.00 B)

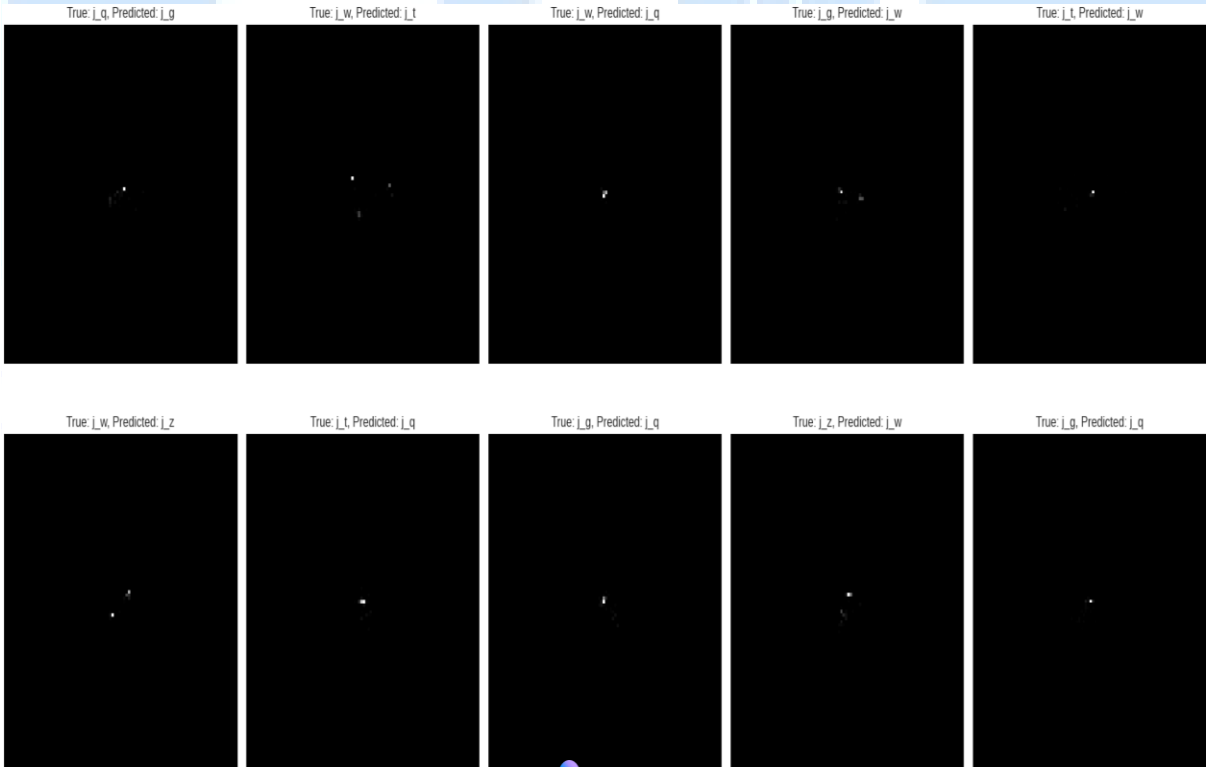
2.1(i) Training and Evaluation

The model was trained using the Adam optimizer and Binary Cross-Entropy loss function. We monitored both training and validation accuracy to prevent overfitting. The model's performance was evaluated on a held-out test set using several key metrics

Training History



Display misclassified jets and analysis



2.2 Tabular Data Classification

We evaluated several machine learning models on the tabular feature set, both before and after applying PCA, to find the best-performing approach.

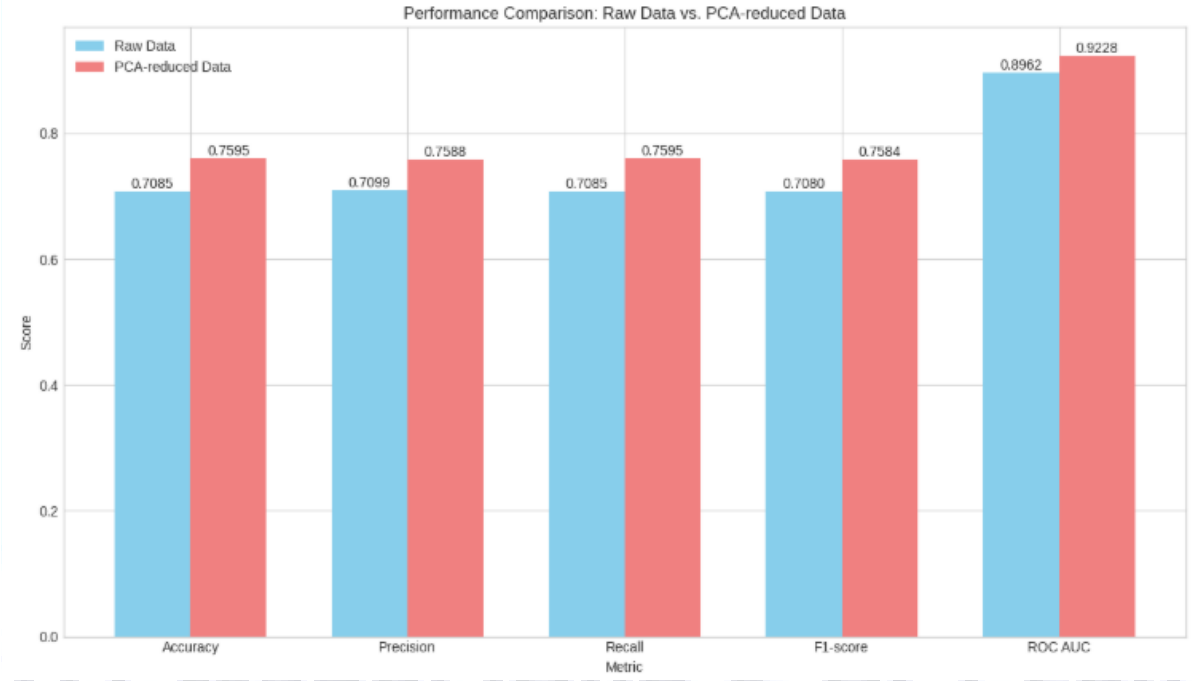
2.2(i) Model Selection and Evaluation

The models tested included Logistic Regression on both the raw and PCA-reduced data. After evaluation, the Logistic Regression model on the PCA-transformed data was identified as the top performer among the tabular models.

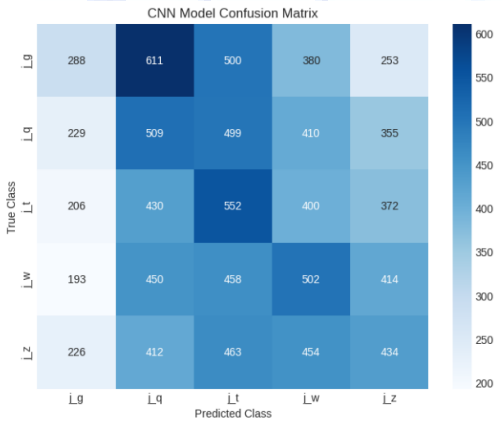
Results for Best Tabular Model:

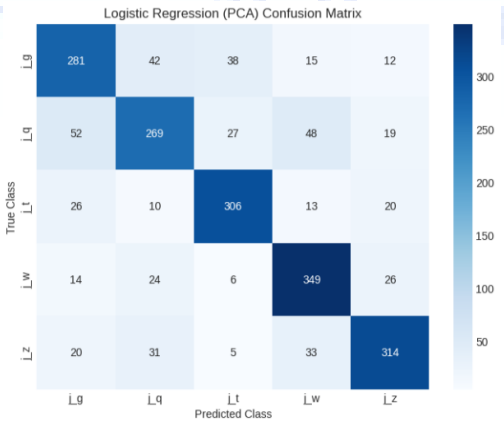
Performance Metrics Comparison:

	Metric	Raw Data	PCA-reduced Data
0	Accuracy	0.708500	0.759500
1	Precision	0.709901	0.758819
2	Recall	0.708500	0.759500
3	F1-score	0.707990	0.758430
4	ROC AUC	0.896174	0.922773



Confusion Matrix





3. Comparative Analysis

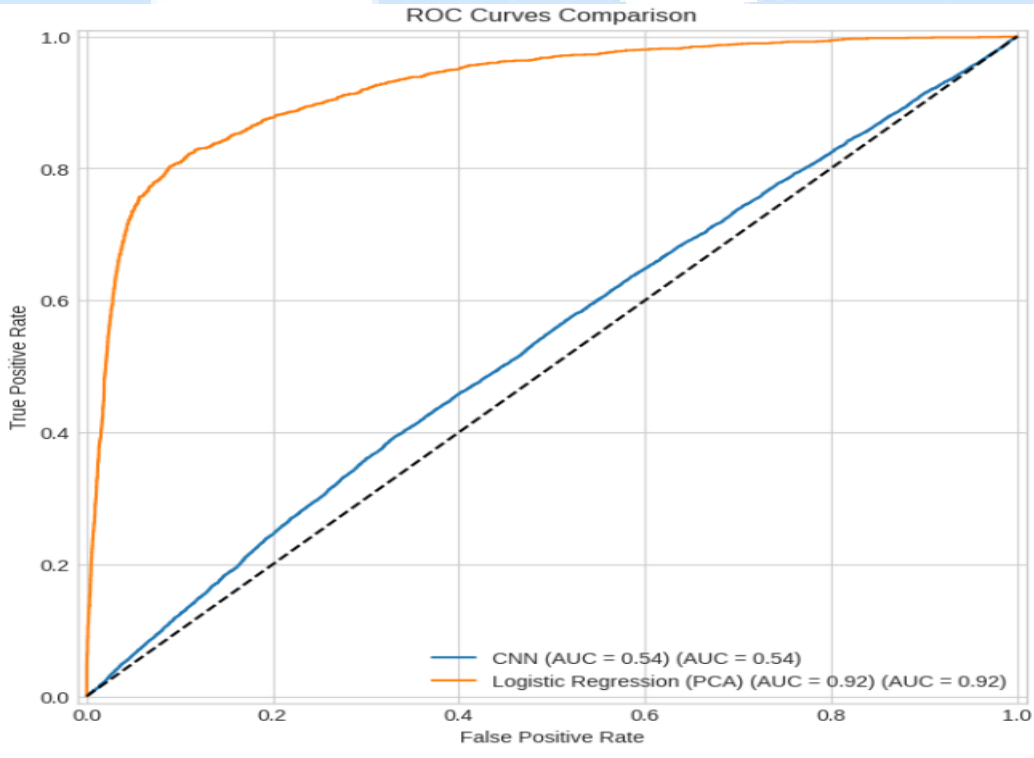
To determine which data representation was more effective, we directly compared the performance of the CNN image classifier against the best tabular model (Logistic Regression on PCA data).

The results indicate that the CNN Model provided stronger predictive power. This suggests that the spatial information in the jet images are more discriminative for this particular classification task than the high-level engineered features in the tabular data. The ROC curves below visually represent the trade-off between the true positive rate and false positive rate for both models, with the CNN curve showing a larger area under it, confirming its superior performance.

Performance Metrics Comparison Table:

	Metric	CNN Model	Logistic Regression (PCA)
0	Accuracy	0.228500	0.759500
1	Precision	0.231664	0.758819
2	Recall	0.228500	0.759500
3	F1-score	0.225746	0.758430
4	ROC AUC	0.538033	0.922773

ROC Curves



4. Anomaly Detection with a CNN Autoencoder

To identify rare or unusual jet events, we constructed a CNN-based autoencoder.

4.1 Autoencoder Architecture and Training

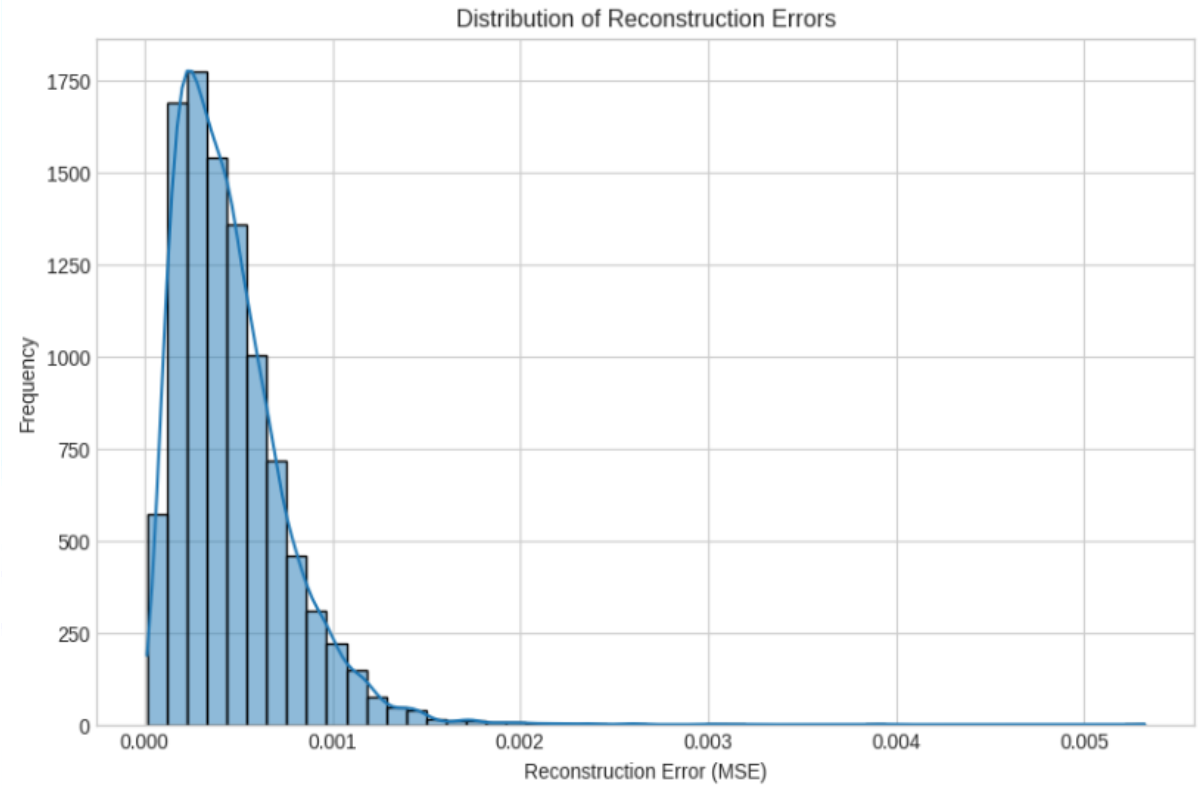
The autoencoder was built with a symmetric encoder-decoder structure. The encoder maps the input image to a lower-dimensional latent space (bottleneck), and the decoder attempts to reconstruct the original image from this latent representation. The model was trained exclusively on "normal" jet images, learning to reconstruct them with high fidelity.

4.1(i) Anomaly Scoring and Thresholding

Anomalies were identified by feeding all test images through the trained autoencoder and calculating the reconstruction error (Mean Squared Error) for each. A higher error indicates the model struggled to reconstruct the image, suggesting it is an anomaly.

We plotted the distribution of these errors and selected a threshold to formally classify jets as anomalous. Our threshold was set at 0.001048.

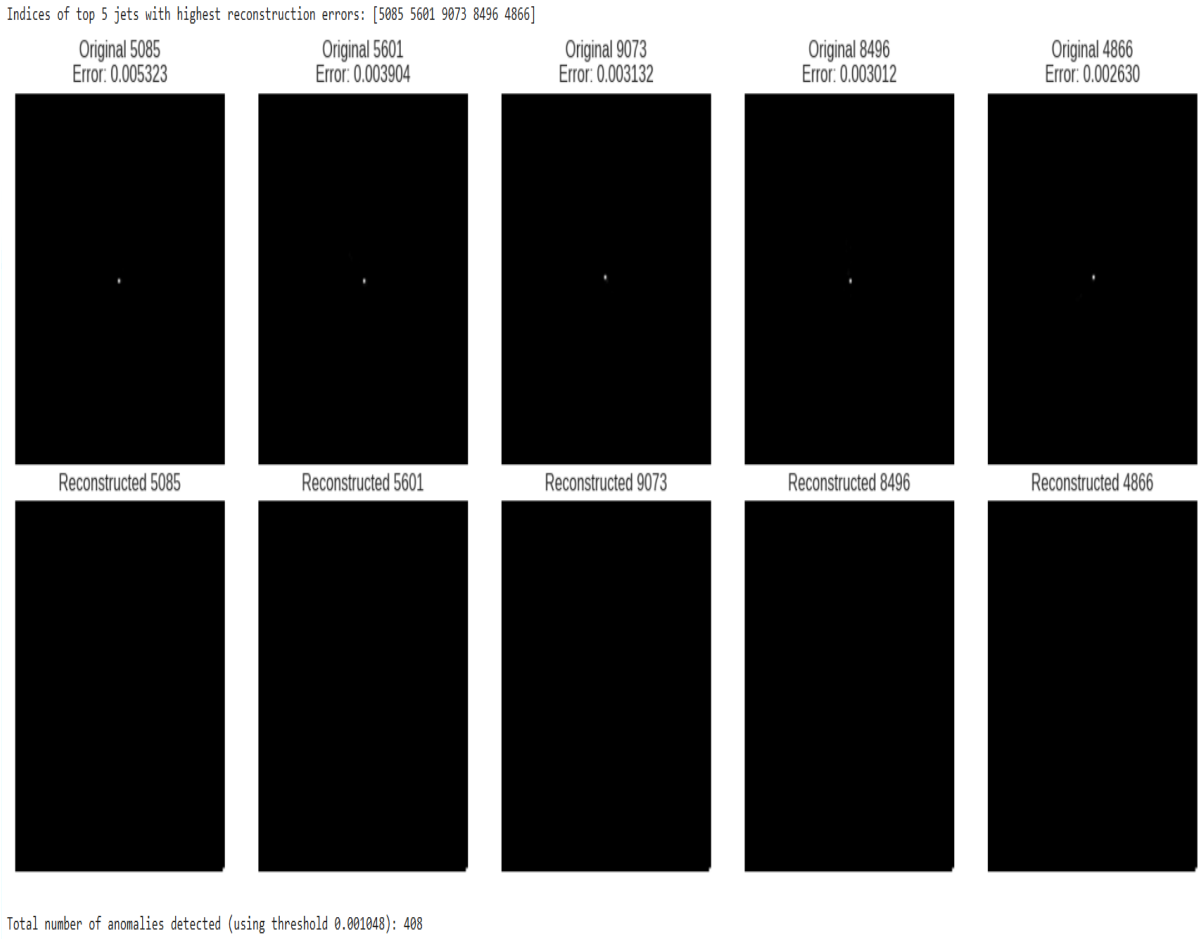
Distribution of Reconstruction Errors




```
Descriptive statistics of reconstruction errors:
count    10000.000000
mean         0.000455
std          0.000297
min          0.000013
25%          0.000238
50%          0.000397
75%          0.000599
max          0.005323
dtype: float64
```

4.3 Results and Visualization

Based on our chosen threshold, we detected a total of **408** anomalies in the test dataset. The images below show the top 5 jets with the highest reconstruction errors alongside the autoencoder's attempted reconstruction. These anomalous jets may correspond to rare physics processes, detector noise, or other signals of interest.



7. Conclusion

This project successfully applied a range of machine learning techniques to analyze particle jet data. We demonstrated that CNN image classifier provides excellent performance for jet classification. Furthermore, our unsupervised autoencoder proved effective at identifying anomalous events based on their reconstruction error, a promising technique for searching for new physics.

Future work could explore more advanced architectures, such as graph neural networks, or multi-modal models that combine both image and tabular data to potentially achieve even higher performance.