Project Report

on

Signal vs. Background Classification in Higgs Boson Detection using Deep Learning

By

Saket Sontakke

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Full Form
SM	Standard Model
LHC	Large Hadron Collider
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
TPR	True Positive Rate
FPR	False Positive Rate
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives
NN	Neural Network
BDT	Boosted Decision Tree
DNN	Deep Neural Network
PNN	Parameterized Neural Network
MVA	Multivariate Analyses
QAML	Quantum Annealing
GBT	Gradient Boosted Tree
CNN	Convolution Neural Network
TMFS	Three-Stage Multi-Objective Feature Selection
DEMDL	Distributed Ensemble Model
SUSY	Super Symmetry
EDA	Exploratory Data Analysis
IQR	Interquartile Range
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
t-SNE	t-Distributed Stochastic Neighbor Embedding
SVM	Support Vector Machine

TABLE OF CONTENTS

ABSTRACT

In particle physics, the terms "signal" and "background" are essential to data analysis. A "signal" refers to the events that we are interested in. For example, the signal could indicate the presence of a new particle not predicted by the Standard Model (SM). The "background" consists of events that mimic the signal but originate from other known processes or sources. The main goal of particle physics experiments, such as those conducted at the Large Hadron Collider (LHC), is to maximize the signal-to-background ratio. Physicists traditionally increase the signal-to-background ratio by applying selection criteria (such as minimum particle momentum) and analyzing a few key discriminating variables. These criteria help filter out background events by focusing on specific characteristics that we are interested in. Instead of selecting only a few discriminating variables, machine learning or deep learning algorithms can use information from many different variables to make a decision about whether a given event looks more like signal or background. This can lead to large improvements in precision.

Keywords: Deep Learning, Particle Physics, High Energy Physics

TECHNICAL CONTENT

1. INTRODUCTION

The 2012 discovery of the Higgs boson at the Large Hadron Collider (LHC) was a historic success in particle physics, validating the existence of a particle that gives mass to other particles according to the Standard Model. Yet, the detection of rare particles like the Higgs boson in large data volumes continues to be a challenge. Particle collisions at the LHC produce enormous datasets, with signal events—those signaling phenomena such as Higgs boson production—hidden in a sea of background noise from known processes. Successful classification of signal versus background events is essential for making progress in our understanding of fundamental physics.

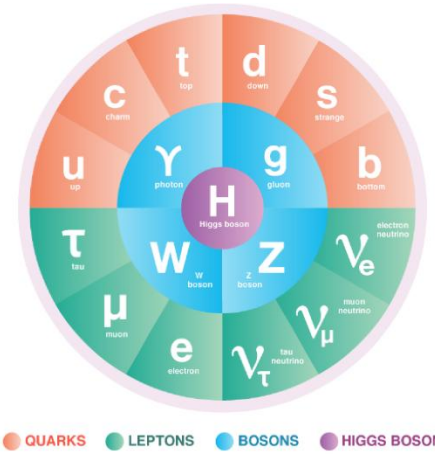

Figure 1: The Standard Model of Particle Physics
source:
https://www.energy.gov/science/doe-explainsthe-standard-model-particle-physics

In the past, physicists have used manual selection criteria and a small number of discriminating variables to improve the signal-to-background ratio. Though these approaches have been effective, they are limited by their dependence on human-specified features and lack of scalability. The latest developments in machine learning, specifically deep learning, present revolutionary capabilities in this area. Through their use of high-dimensional data and automatic feature extraction, deep learning models can greatly enhance classification accuracy and efficiency and outperform standard methods in accuracy and scalability.

This project seeks to investigate the use of deep learning methods for signal and background event classification in Higgs boson discovery. Based on Monte Carlo simulation data, the objective is to learn models that are able to differentiate between signal and background

processes through the analysis of various kinematic properties and calculated features. With the use of deep neural networks, this research intends to help enhance data analysis workflows in high-energy physics experiments so that more efficient identification of rare particles and new phenomena beyond the Standard Model may be achieved.

2. PERFORMANCE METRICS

To evaluate the performance of the classifiers, we use Receiver Operating Characteristic (ROC) curves and measure the Area Under the Curve (AUC) as our primary metric. Additionally, we assess the discovery significance, a standard measure in high-energy physics, to quantify the improvement in signal detection.

2.1. Area Under the ROC Curve

The Receiver Operating Characteristic (ROC) curve is a plot that illustrates how well a classifier distinguishes between positive and negative classes. It is created by plotting the True Positive Rate (TPR) given by eq. (1) against the False Positive Rate (FPR) given by eq. (2) at different decision thresholds. The Area Under the Curve (AUC) is a numerical value that summarizes the overall performance of the classifier by measuring the total area beneath the ROC curve.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

Where:

- TP (True Positives) = correctly classified signal events
- FP (False Positives) = background events misclassified as signal
- TN (True Negatives) = correctly classified background events
- FN (False Negatives) = signal events misclassified as background

The Area Under the ROC Curve (AUC) [1] is mathematically defined in eq. (3).

$$AUC = \int_0^1 TPR(FPR)d(FPR) \quad (3)$$

Visually, the AUC can be understood as the total area under the ROC curve. This area is made up of many infinitesimal vertical strips, each having:

- A width of $d(FPR)$ (a small change in the False Positive Rate).
- A height of $TPR(FPR)$ (the True Positive Rate at that point).

By summing up the areas of these infinitely small strips, we obtain the total AUC value, which tells us how well the model performs in distinguishing between positive and negative samples. AUC represents the probability that a randomly chosen signal event is ranked higher by the classifier than a randomly chosen background event. The value of AUC ranges from 0 to 1:

- AUC = 1.0 - Perfect classifier (separates all positives and negatives correctly).
- AUC = 0.5 - Random classifier (no discrimination ability, equivalent to random guessing).
- AUC < 0.5 - Worse than random (predicting the wrong class more often than not).

2.2. Discovery Significance

Particles produced in high-energy collisions, such as those in the Large Hadron Collider (LHC), are extremely short-lived. They decay almost instantly into other particles, which makes direct observation impossible. Instead, physicists infer their existence by analyzing the properties of the decay products—such as charge, mass, spin, and velocity. At the LHC, millions of collisions occur every second, and the resulting data is filtered through trigger systems to identify potential signatures of rare particles. Due to the high volume of data and the complexity of the underlying physical processes, statistical fluctuations can lead to errors. These errors generally fall into three categories:

- Statistical Error: Random fluctuations in the data due to limited sample sizes.
- Systematic Error: Errors due to detector imperfections, calibration issues, or theoretical model assumptions.
- Background Noise: Other known physics processes that mimic the signal being

searched for.

To minimize errors and avoid false discoveries, particle physicists use rigorous statistical significance tests before claiming new discoveries [2].

In high-energy physics, a common goal is to detect rare signal events against a dominant background. The discovery significance quantifies how confidently an observed signal stands out from the background fluctuations. It is typically expressed in terms of standard deviations (σ) in a normal distribution.

For a given number of signal events (S) and background events (B), the statistical significance is calculated as eq. (4).

$$Z = \frac{S}{\sqrt{B}} \tag{4}$$

Where:

- S = expected number of signal events
- B = expected number of background events

The term sigma (σ) refers to the standard deviation in a normal (Gaussian) distribution. It measures how much an observed result deviates from the expected background fluctuations. If a given dataset follows a normal distribution, then the probability of obtaining a measurement within a certain number of standard deviations (σ) from the mean is:

- 1σ: 68% probability
- 2σ: 95% probability
- 3σ: 99.7% probability (evidence threshold)
- 5σ: 99.99994% probability (discovery threshold)

A 5σ significance level (Z=5) is the standard threshold in particle physics to claim a discovery, as it corresponds to a p-value of approximately 3×10^{-7}, meaning the probability of the signal being a statistical fluctuation is about 1 in 3.5 million.

3. LITERATURE SURVEY

3.1. Traditional and Deep Learning Methods

Early approaches relied on shallow neural networks (NNs) and boosted decision trees (BDTs). Baldi et al. (2014) [3] demonstrated that deep neural networks (DNNs) outperform these methods by automating feature extraction from raw kinematic data. Their work showed that DNNs achieve up to an 8% improvement in the area under the receiver operating characteristic curve (AUC) compared to BDTs, eliminating the need for manually engineered high-level features. DNNs also improved discovery significance (e.g., 5.0σ vs. 3.7σ for BDTs in Higgs benchmarks), underscoring their ability to capture non-linear correlations in high-dimensional data.

Baldi et al. (2016) [4] expanded this framework with parameterized neural networks (PNNs), which integrate physics parameters (e.g., particle mass) as inputs. This allows a single model to interpolate across parameter values, reducing computational costs and improving performance for intermediate values. PNNs matched the accuracy of dedicated models trained at specific masses, even in high-dimensional scenarios, highlighting their flexibility for multi-parameter optimization.

3.2. Ensemble Learning and Stacking

To address the limitations of single classifiers, Alves (2017) [5] proposed stacked generalization, combining outputs from XGBoost, shallow NNs, and naive Bayes using logistic regression. This ensemble method achieved higher statistical significance (5.5σ vs. 3.7σ for standalone BDTs) in Higgs boson searches. Stacking proved particularly effective in multivariate analyses (MVA), where combining classifiers improved sensitivity by capturing complementary correlations. While slightly less accurate than DNNs in cut-and-count analyses, stacking required fewer computational resources, making it practical for large-scale experiments.

3.3. Scalability with Big Data Frameworks

Handling massive datasets (e.g., 11 million events) necessitated scalable solutions. Azhari et

al. (2020) [6] evaluated ML methods (Logistic Regression, Decision Trees, Random Forest, Gradient Boosted Trees) using PySpark, a distributed computing framework. Gradient Boosted Trees (GBT) outperformed others, achieving 83% accuracy and 0.91 AUC on Kaggle Higgs data. Their study emphasized PySpark's efficiency in preprocessing and training, enabling rapid experimentation on terabyte-scale datasets. However, GBT's performance plateaued for imbalanced datasets, underscoring the need for adaptive sampling techniques.

3.4. Parametric Neural Networks

Anzalone et al. (2022) [7] refined parametric neural networks for high-energy physics by introducing the Affine Parametric Neural Network (AffinePNN), which uses an innovative affine conditioning mechanism to integrate the signal mass feature throughout the network. This approach enables a single model to effectively interpolate between different mass hypotheses and improves classification performance on both balanced and imbalanced datasets.

Through experiments on the HEPMASS and HEPMASS-IMB datasets, the study demonstrates that careful design choices—such as balanced training and optimized background mass distribution—lead to superior ROC-AUC and significance ratios compared to traditional methods. These results suggest that the AffinePNN not only streamlines the detection process by replacing multiple individual classifiers but also offers a robust framework for future applications in particle physics.

3.5. Learning-to-Rank for Higgs Detection

Köppel et al. (2022) [8] demonstrated a learning-to-rank approach for Higgs detection by combining CNNs with a DirectRanker to order Higgs boson candidates based on signal likelihood. Their method employs pairwise training—leveraging quadratic combinations of signal–background pairs—to address class imbalance and enhance robustness in low-signal scenarios. By integrating CNN feature extraction with the DirectRanker, they achieved a significant improvement in performance, with a Z_0 significance score of 2.78 compared to 2.43 for conventional BDTs on balanced datasets. Furthermore, transfer learning was effectively applied by pre-training on approximate simulations to boost performance on

precise detector data across different configurations.

3.6. Feature Selection and Distributed Learning for Large Datasets

Babu and Malathi (2023) [9] tackled the challenges of dimensionality reduction and scalability in large-scale Higgs datasets through a hybrid approach. Their work introduces a Three-Stage Multi-Objective Feature Selection (TMFS) that combines metrics such as the correlation coefficient, Fisher score, and information gain, among others, to iteratively select optimal features and reduce dimensionality by 30–50%. Complementing this, their Distributed Ensemble Model (DEMDL) stacks SVM and DL4jMlpClassifier with a RIPPER meta-classifier to achieve 87–89% accuracy, outperforming standalone models. Distributed training across multiple machines further enhanced scalability, reducing computational overhead while maintaining high precision (94%) and recall (97%).

3.7. Distributed Evolutionary Neural Networks for Big Data

Haritha et al. (2023) [10] introduced a Distributed Evolutionary Neural Network (DENN) model that integrates Genetic Algorithms (GA) with Artificial Neural Networks (ANNs) in a distributed computing environment to address challenges in big data classification. By leveraging Apache Spark, the model efficiently parallelizes genetic operations such as selection, crossover, and mutation, significantly accelerating convergence. The GA replaces traditional gradient descent, mitigating issues like local optima and sluggish learning in high-dimensional spaces. The model was benchmarked using large-scale datasets—SUSY, HEPMASS, and HIGGS—demonstrating comparable or better classification accuracy (up to 67.30%) and ROC-AUC values than conventional ANN-GA models. Most notably, DENN achieved up to 80% reduction in training time and showed strong scalability and speedup trends up to six processing nodes. These results underline DENN's potential to deliver high-performance classification in big data environments while maintaining accuracy and computational efficiency.

Section	Method	Key Features	Performance Highlights
3.1 Baldi et al. (2014), (2016)	Traditional and Deep Learning Methods	DNNs automate feature extraction, outperform BDTs by up to 8% AUC. Capture non-linear correlations.	AUC: 0.885 (HIGGS) AUC: 0.879 (SUSY)
3.2 Alves (2017)	Ensemble Learning and Stacking	Stacked generalization of XGBoost, NNs, Naive Bayes using logistic regression.	AUC: 0.917
3.3 Azhari et al. (2020)	Big Data Scalability (PySpark + GBT)	GBT via PySpark on 11M event dataset; efficient preprocessing/training.	AUC: 0.70
3.4 Anzalone et al. (2022)	Parametric Neural Networks (AffinePNN)	Affine conditioning to integrate mass; balances training & background distribution.	AUC: 0.9323
3.5 Köppel et al. (2022)	Learning-to-Rank (DirectRanker + CNNs)	CNNs with pairwise training to rank Higgs likelihood; effective in low-signal.	AUC: 0.92
3.6 Babu & Malathi (2023)	Feature Selection + Distributed Learning	TMFS + DEMDL (SVM, DL4j, RIPPER); 30–50% feature reduction.	Accuracy: 88.9127% (Machine 2)
3.8	Distributed genetic algorithm-based artificial neural network (ANN)	Distributed Genetic Algorithm (GA) integrated with ANN using Apache Spark framework. (ANN) model to address challenges in big data classification, improving convergence time and scalability.	AUC: 0.672 (HIGGS) AUC: 0.792 (SUSY) AUC: 0.910 (HEPMASS)

Table 1: Summarized Literature Survey

DETAILS OF DESIGN/TECHNOLOGY

4. DATASET DESCRIPTION

4.1. HIGGS Dataset

The HIGGS dataset was created to facilitate the classification of events into signal and background categories, specifically for identifying processes that produce Higgs bosons. The data was generated using Monte Carlo simulations, replicating the conditions of particle collisions in accelerators like the Large Hadron Collider (LHC).

- Number of Instances: 11,000,000
- Features:
    - The first 21 features represent kinematic properties measured by particle detectors.
    - The last 7 features are high-level features derived from the first 21 through physicists' domain knowledge to enhance discrimination between signal and background events.
- Task: Binary classification (signal vs. background).
- Test Set: The last 500,000 examples are reserved for testing.
- Applications: This dataset is widely used to evaluate machine learning models for high-energy physics, with benchmark results available for Bayesian Decision Trees and 5-layer neural networks.

4.2. SUSY Dataset

The SUSY dataset aims to distinguish between signal processes that produce supersymmetric particles (SUSY) and background processes that do not. It was also generated using Monte Carlo simulations to replicate particle collision scenarios.

- Number of Instances: 5,000,000
- Features:
    - The first 8 features are low-level kinematic properties directly measured by particle detectors.
    - The last 10 features are high-level features derived from the low-level ones using physics intuition to improve classification accuracy.

- Task: Binary classification (signal vs. background).
- Test Set: The last 500,000 examples are reserved for testing.
- Applications: This dataset is used extensively in machine learning research to develop and benchmark algorithms for classification tasks in particle physics.

ANALYTICAL AND/OR EXPERIMENTAL WORK

5. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a fundamental step in the data analysis process, where we investigate and visualize the characteristics of the dataset. The primary objective of EDA is to summarize key properties, identify patterns, detect anomalies, and uncover relationships between features. This process helps in understanding the data better, which is essential for making informed decisions during model building.

5.1. Class Distribution Analysis

Class distribution analysis is a crucial step in Exploratory Data Analysis (EDA), especially for binary classification datasets. This process involves examining the frequency of occurrences for each class to detect any potential class imbalance. In machine learning, class imbalance can significantly impact the performance of predictive models, as algorithms may become biased toward the majority class.

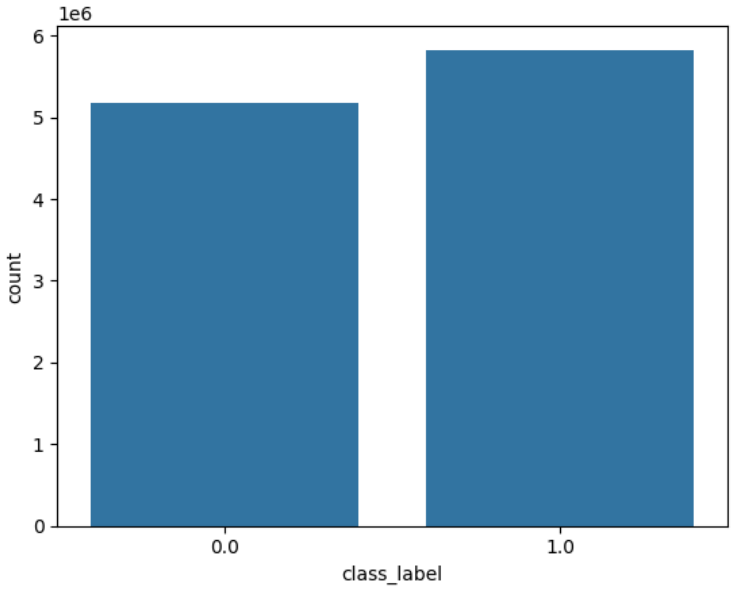

Figure 2: Class distribution plot of HIGGS dataset

In the HIGGS dataset, the class distribution is as follows:

- Class 1.0: 5,829,123 occurrences (positive class)
- Class 0.0: 5,170,877 occurrences (negative class)

There is a slight class imbalance in the dataset, with the positive class having approximately 659,246 more samples than the negative class. This difference accounts for roughly 6.4% of the total dataset size.

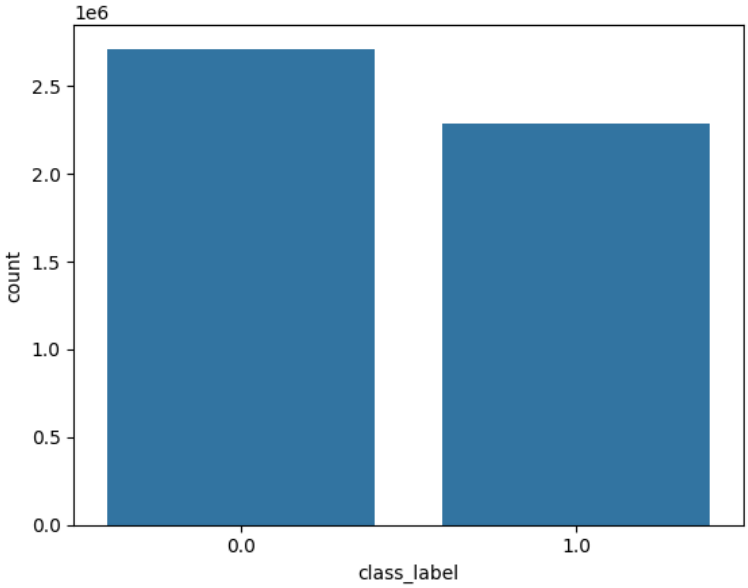

Figure 3: Class distribution plot of SUSY dataset

In the SUSY dataset, the class distribution is as follows:

- Class 0.0: 2,712,173 occurrences (negative class)
- Class 1.0: 2,287,827 occurrences (positive class)

There is a slight class imbalance, with the negative class having approximately 424,346 more samples than the positive class. This difference accounts for roughly 8.5% of the total dataset size.

However, the imbalance is not significant enough to warrant resampling techniques, as the classes are relatively balanced. Therefore, no additional measures, such as oversampling or undersampling, were applied during the analysis. The datasets weree used in their original form, as the minor imbalance is unlikely to significantly impact the performance of the model.

5.2. Correlation Analysis

Correlation analysis is a statistical method used to measure and interpret the strength and

direction of the linear relationship between two variables. The most common metric is the Pearson correlation coefficient, which ranges from -1 to +1. A coefficient close to +1 indicates a strong positive linear relationship (as one variable increases, the other tends to increase), while a coefficient close to -1 indicates a strong negative linear relationship (as one increases, the other tends to decrease). A correlation near zero suggests little to no linear relationship. When dealing with many variables at once, it is helpful to visualize the correlations in a correlation matrix—a grid of pairwise correlation coefficients. The diagonal entries are always 1 because each variable is perfectly correlated with itself. Symmetry about the diagonal reflects the fact that the correlation of X with Y is the same as Y with X.

Correlation analysis is often used to:

1. Identify predictive features: Variables with higher correlation to a target variable (like a class label) may be more relevant in classification or regression tasks.
2. Spot multicollinearity: High correlation between two features can sometimes degrade performance in certain models or cause interpretability issues.
3. Guide feature engineering: If two variables are highly correlated, sometimes one can be removed or transformed to simplify the model.

Color Coding: The cells are colored to make patterns easier to spot:

- Dark Red: Strong positive correlation (close to 1.0).
- Dark Blue: Strong negative correlation (close to -1.0).
- Light Colors: Weak or no correlation (close to 0).

5.2.1. Correlation analysis of HIGGS dataset

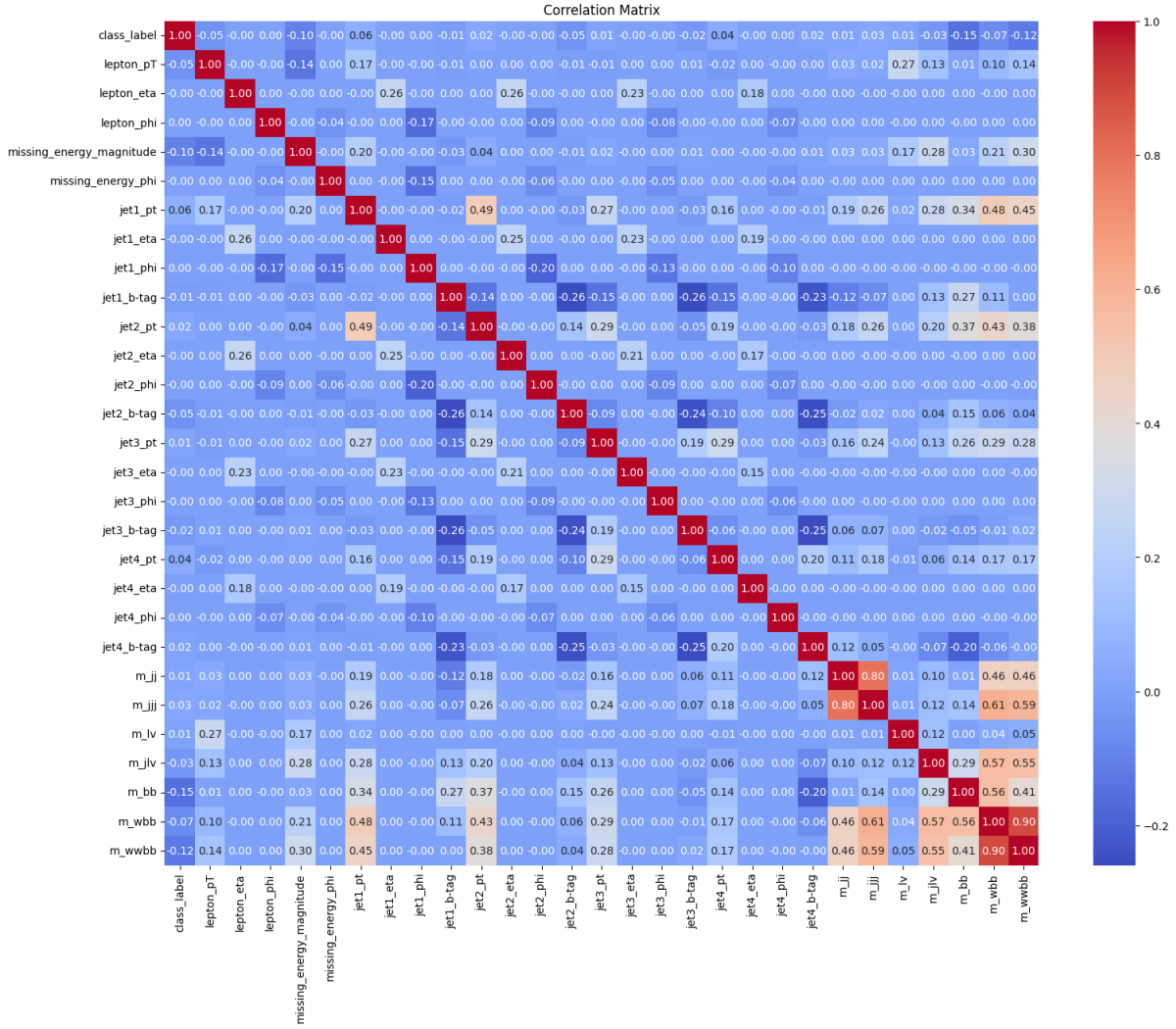

Figure 4: Correlation Matrix for features in HIGGS dataset

Diagonal Elements:

- The diagonal cells (e.g., lepton_pt with lepton_pt) show a correlation of 1.0 and are colored dark red.
- This is expected because every variable is perfectly correlated with itself.

Positive Correlations:

- Some pairs of variables show strong positive correlations (red shades):
  - m_jj and m_jjj: 0.81 (mass of two jets vs. three jets).
  - m_lv and m_jlv: 0.97 (mass of lepton+vector vs. jet+lepton+vector).
  - m_wwbb and m_wbb: 0.91 (mass of WWbb vs. Wbb combinations).

- These high values suggest these variables are closely related, possibly redundant or physically linked in the context of particle physics.

Negative Correlations:

- Negative correlations (blue shades) are less common but present:
  - jet3_b-tag and jet4_b-tag: -0.25 (b-tagging scores for jets 3 and 4).
  - m_bb and jet4_phi: -0.29 (mass of b-jets vs. azimuthal angle of jet 4).
- These indicate that as one variable increases, the other tends to decrease.

Weak Correlations:

- Many cells show values close to 0 (light colors), indicating little to no linear relationship.
- For example, class_label has near-zero correlations with most variables, suggesting it may not be strongly linearly tied to these features.

Notable Moderate Correlations:

- lepton_pt and missing_energy_magnitude: 0.26 (lepton momentum vs. missing energy).
- jet1_pt and jet2_pt: 0.15 (momentum of jets 1 and 2).
- jet1_b-tag and jet2_b-tag: 0.26 (b-tagging scores for jets 1 and 2).
- These suggest some dependency between these pairs, though not as strong as the mass-related examples.

Insights and Applications

- Multicollinearity: High correlations (e.g., m_wwbb and m_wbb at 0.91) may indicate multicollinearity, where variables provide overlapping information. This could complicate statistical models unless addressed (e.g., by removing redundant features).
- Feature Importance: Variables with low correlations to class_label (near 0) might be less useful for predicting it in a classification task, while those with stronger correlations (if any) could be more predictive.
- Scientific Insights: The correlations reflect physical relationships in the data. For example, the strong link between m_jj and m_jjj makes sense in particle physics, as both involve jet masses, just with different numbers of jets.

Correlation analysis of SUSY dataset

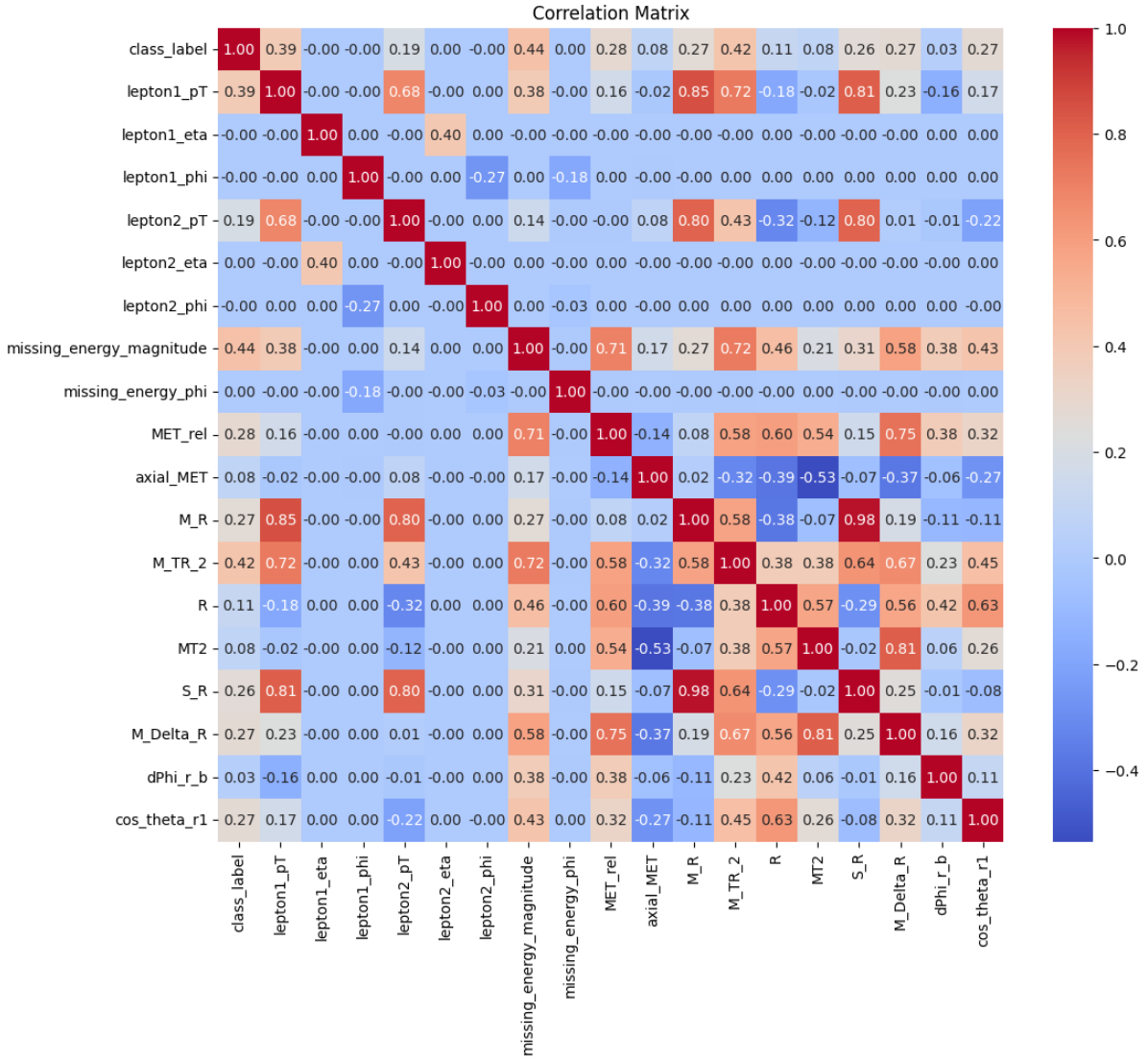

Figure 5: Correlation Matrix for features in SUSY dataset

Diagonal Elements:

- The diagonal cells (e.g., lepton1_pT with lepton1_pT) show a correlation of 1.0 and are colored dark red.
- This is expected because every variable is perfectly correlated with itself.

Positive Correlations:

- Some pairs of variables show strong positive correlations (red shades):
  - lepton1_pT and lepton2_pT: 0.68 (momentum of the first and second leptons).
  - M_TR_2 and S_R: 0.81 (some measure of transverse mass vs. an unknown variable).

    - M_R and M_TR_2: 0.72 (possibly related mass variables).
- These high values suggest closely related features, possibly indicating redundancy or a physical relationship in the dataset.

Negative Correlations:

- Negative correlations (blue shades) are present:
    - lepton1_pT and R: -0.18 (momentum of first lepton vs. R variable).
    - lepton2_pT and R: -0.32 (momentum of second lepton vs. R variable).
    - axial_MET and M_R: -0.32 (some missing transverse energy component vs. mass variable).
- These indicate that as one variable increases, the other tends to decrease.

Weak Correlations:

- Many cells show values close to 0 (light colors), indicating little to no linear relationship.
- For example, class_label has near-zero correlations with most variables, suggesting it may not be strongly linearly tied to these features.

Notable Moderate Correlations:

- lepton1_pT and missing_energy_magnitude: 0.38 (lepton momentum vs. missing energy).
- MET_rel and M_TR_2: 0.58 (some transverse energy component vs. transverse mass variable).
- S_R and M_R: 0.75 (indicating a moderate relationship between these physical quantities).

Insights and Applications:

- Multicollinearity: High correlations (e.g., M_TR_2 and S_R at 0.81) may indicate redundancy, which could complicate statistical models unless addressed.
- Feature Importance: Variables with low correlations to class_label (near 0) might be less useful for predicting it in a classification task, while those with stronger correlations (if any) could be more predictive.
- Scientific Insights: The correlations reflect physical relationships in the data. For example, the strong link between M_R and M_TR_2 suggests that they may be derived from similar physical principles.

5.3. Histogram Analysis

A histogram is a type of bar plot that shows how data are distributed across different value ranges (bins). On the x-axis, you have intervals (bins) of the variable's possible values; on the y-axis, you have the frequency (count) of data points that fall into each bin. Histograms help you see:

- Central tendency: Where the data tend to cluster.
- Spread (variance): How widely the data are dispersed.
- Skewness: Whether the data are more spread out on one side of the distribution.
- Modality: Whether there is a single peak (unimodal), two peaks (bimodal), or more.
- Presence of outliers: Unusually high or low values that appear far from the rest.

Based on the histogram shapes, the features can be broadly grouped into the following categories:

1. Discrete or Binary Distributions:

   Features that are not continuous and exhibit distinct jumps or only a few values. These often represent categorical or count data, which may require different treatment such as one-hot encoding or specific count data models.
2. Skewed Distributions:

   Features that display a pronounced tail on one side (either right-skewed or left-skewed). These may require transformations (e.g., log, square root) to normalize their distribution or other preprocessing to manage the asymmetry.
3. Approximately Normal (Symmetrical) Distributions:

   Features with bell-shaped curves, suggesting that these variables are centered around a mean with relatively symmetric dispersion. They may be suitable for standard scaling and methods that assume normality.
4. Approximate Uniform Distributions:

   Features where values are evenly spread across a range with approximately equal frequencies. These distributions indicate no strong concentration around any particular value. They may require binning or other transformations depending on the analysis context.

5. Multimodal/Bimodal Distributions:

   Features showing more than one peak, indicating the presence of subgroups or multiple underlying processes. These might benefit from further investigation to determine if splitting into subgroups or using mixture models is appropriate.

5.3.1. Histogram analysis of HIGGS dataset

1. Discrete or Binary Distributions:

   The feature class_label falls under this category with discrete values 0 and 1.

2. Skewed Distributions:

   The features- Transverse Momentums (_pT), Missing Energy Magnitude, and Invariant Mass show higher frequencies at lower values with a long tail extending to the right, indicating a right-skewed (positively skewed) distribution.

3. Approximately Normal (Symmetrical) Distributions:

   In the Pseudorapidity plots the histogram is bell-shaped with the highest frequency in the middle, suggesting a normal (Gaussian) distribution.

4. Approximately Uniform Distributions:

   In the Azimuthal Angle plots the data is spread evenly across the range, indicating a uniform distribution.

5. Multimodal/Bimodal Distributions:

   The b-tag plots display three distinct peaks, which aligns with the nature of b-tagging—a method used in particle physics to identify jets from bottom quarks. This discrete pattern reflects the categorical assignment of jets into distinct groups.

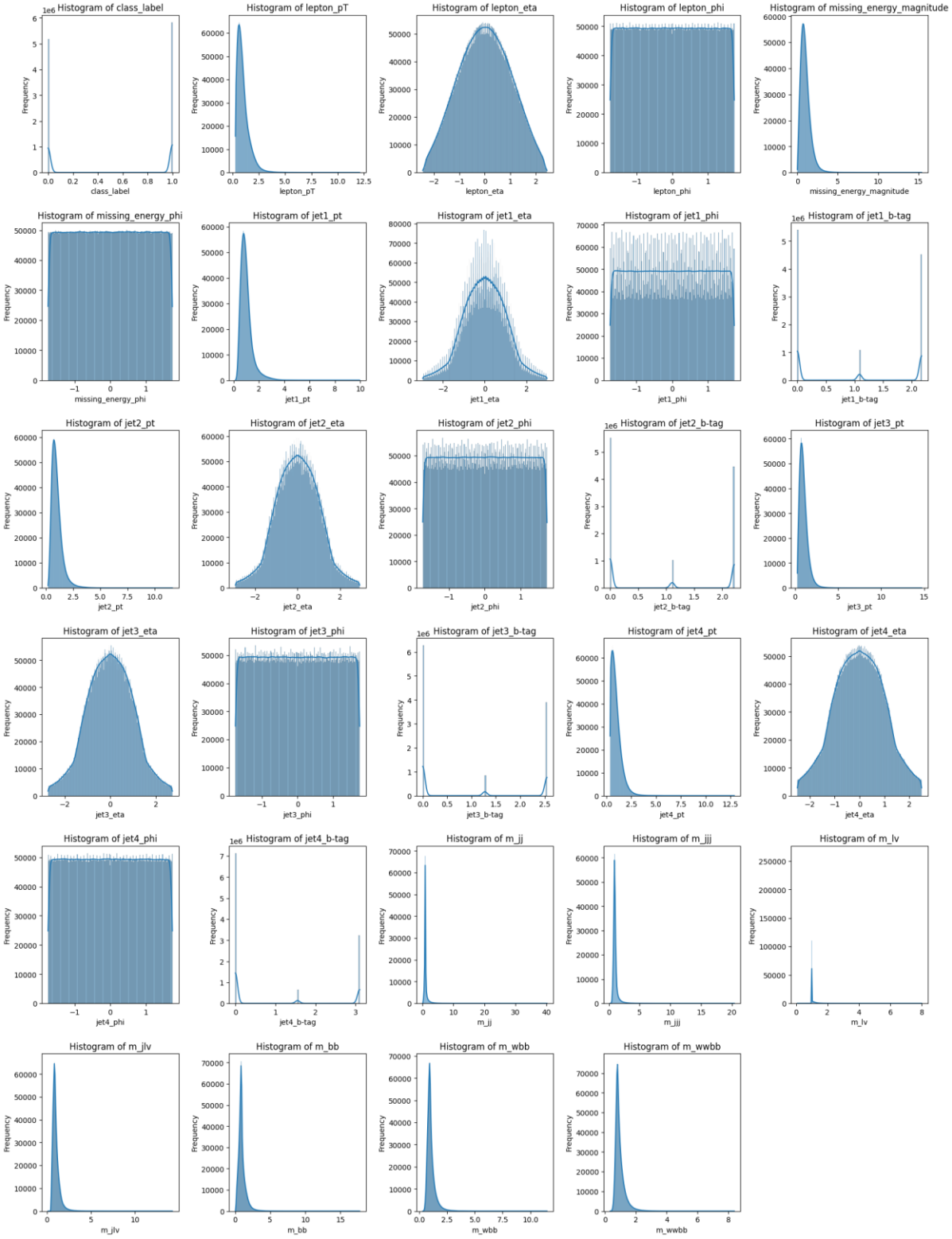

Figure 6: Histogram plots of features in HIGGS dataset

5.3.2. Histogram analysis of SUSY dataset

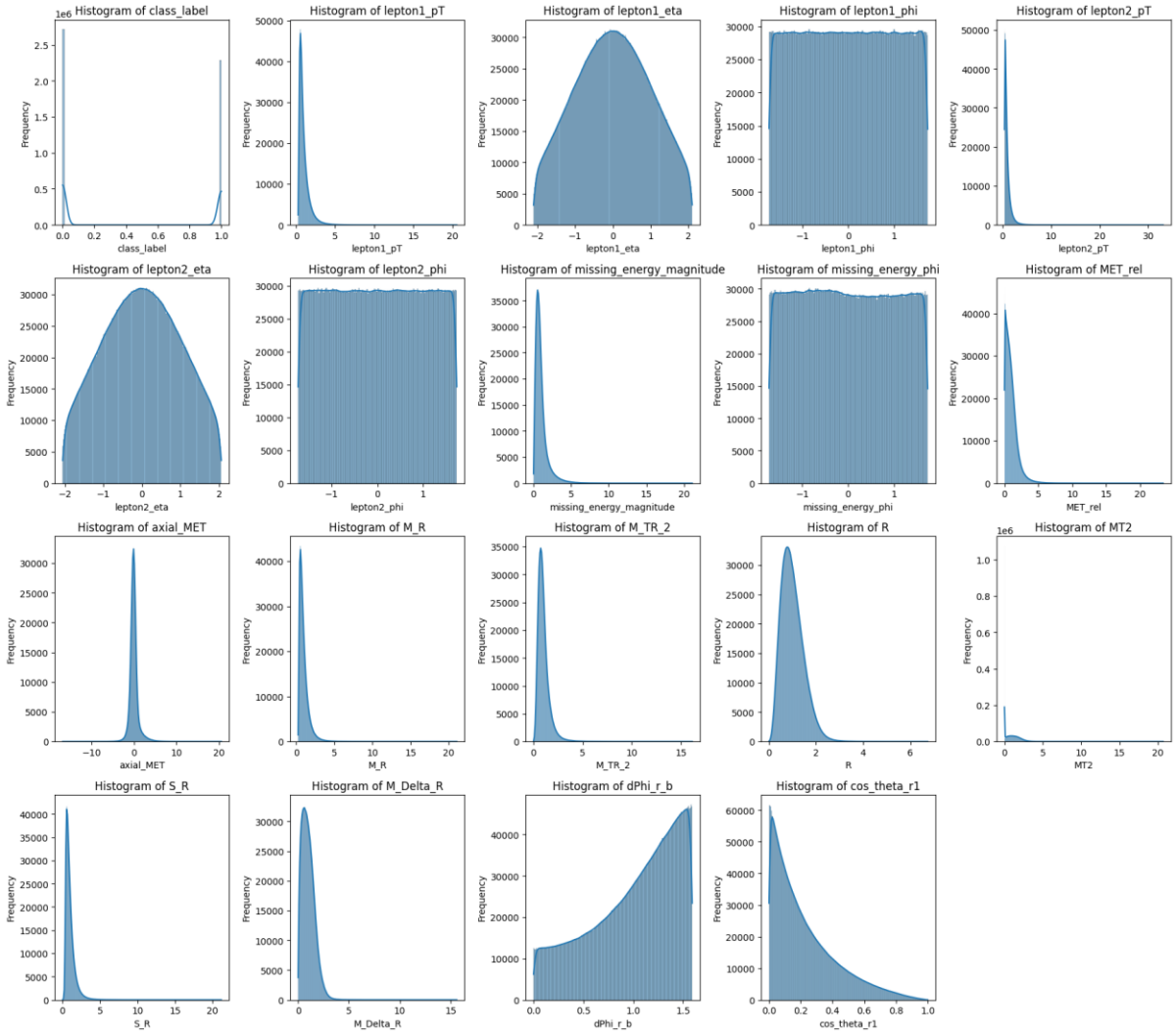

Figure 7: Histogram plots of features in SUSY dataset

1. Discrete or Binary Distributions:

   The class_label feature falls under this category as it takes discrete values (0 and 1), indicating a classification problem.

2. Skewed Distributions:

   Features such as lepton1_pT, lepton2_pT, missing_energy_magnitude, MET_rel, axial_MET, M_R, M_TR_2, R, MT2, S_R, M_Delta_R show a pattern where most of the data is concentrated on the left side with a long tail extending to the right, indicating right-skewed (positively skewed) distributions.

3. Approximately Normal (Symmetrical) Distributions:
   The pseudorapidity (eta) features (lepton1_eta, lepton2_eta) exhibit a bell-shaped curve, with the highest frequency around the center, suggesting a normal (Gaussian) distribution.
4. Approximately Uniform Distributions:
   Features such as lepton1_phi, lepton2_phi, missing_energy_phi have histograms where data appears to be evenly distributed across the range, indicating a uniform distribution.
5. Multimodal/Bimodal Distributions:
   The b-tagging feature (dPhi_r_b) appears to have multiple peaks, suggesting a multimodal or bimodal distribution. This aligns with the nature of b-tagging in particle physics, which categorizes jets into distinct groups based on bottom quark identification.

5.4. Box Plot Analysis

A box plot (also known as a box-and-whisker plot) is a standardized way of displaying the distribution of data based on:

1. Minimum – The smallest data point, excluding outliers.
2. First Quartile (Q1) – The median of the lower half of the data (25th percentile).
3. Median (Q2) – The middle value of the dataset (50th percentile).
4. Third Quartile (Q3) – The median of the upper half of the data (75th percentile).
5. Maximum – The largest data point, excluding outliers.

Key Features of a Box Plot:

- The box represents the interquartile range (IQR) (Q1 to Q3), which contains the middle 50% of the data.
- The line inside the box represents the median (Q2).
- The whiskers extend from the box to the smallest and largest values within 1.5 times the IQR from Q1 and Q3.
- Outliers are shown as individual points beyond the whiskers.

5.4.1. Box Plot analysis of HIGGS analysis

1. The first plot ("class_label") appears as a solid box – This means that the variable is categorical, with only 2 unique values (e.g., binary labels like 0 and 1).
2. Most numerical variables show a long tail with outliers – Variables like lepton_pT, missing_energy_magnitude, jet_pT, m_jj, m_bb, and m_wbb have many extreme values, suggesting that the data contains several large observations that fall beyond 1.5 times the IQR.
3. Some variables have a symmetric distribution – Features like lepton_eta, lepton_phi, jet_eta, jet_phi, and missing_energy_phi have their boxes centered, with whiskers extending evenly, indicating a more balanced spread of values.
4. Several variables have very small IQRs – Features such as b-tag values and m_lv suggest that most of the data is concentrated within a narrow range, but outliers extend beyond the whiskers.
5. A few variables have values near zero – Certain computed or transformed features, such as m_wbb and jet_b-tag, seem to have a restricted range, potentially due to specific feature engineering or domain-related constraints.

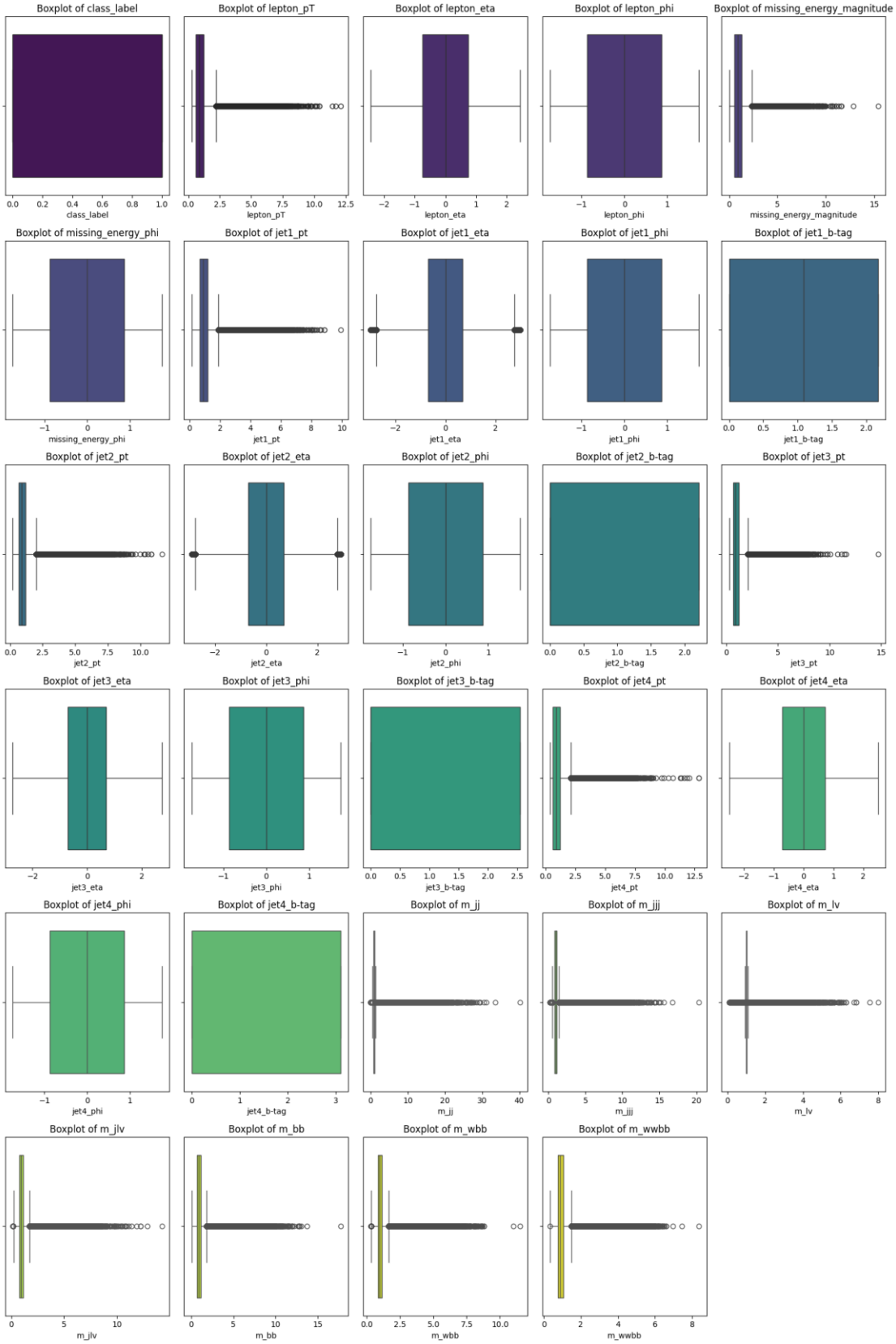

Figure 8: Box plots of features in HIGGS dataset

5.4.2. Box Plot analysis of SUSY analysis

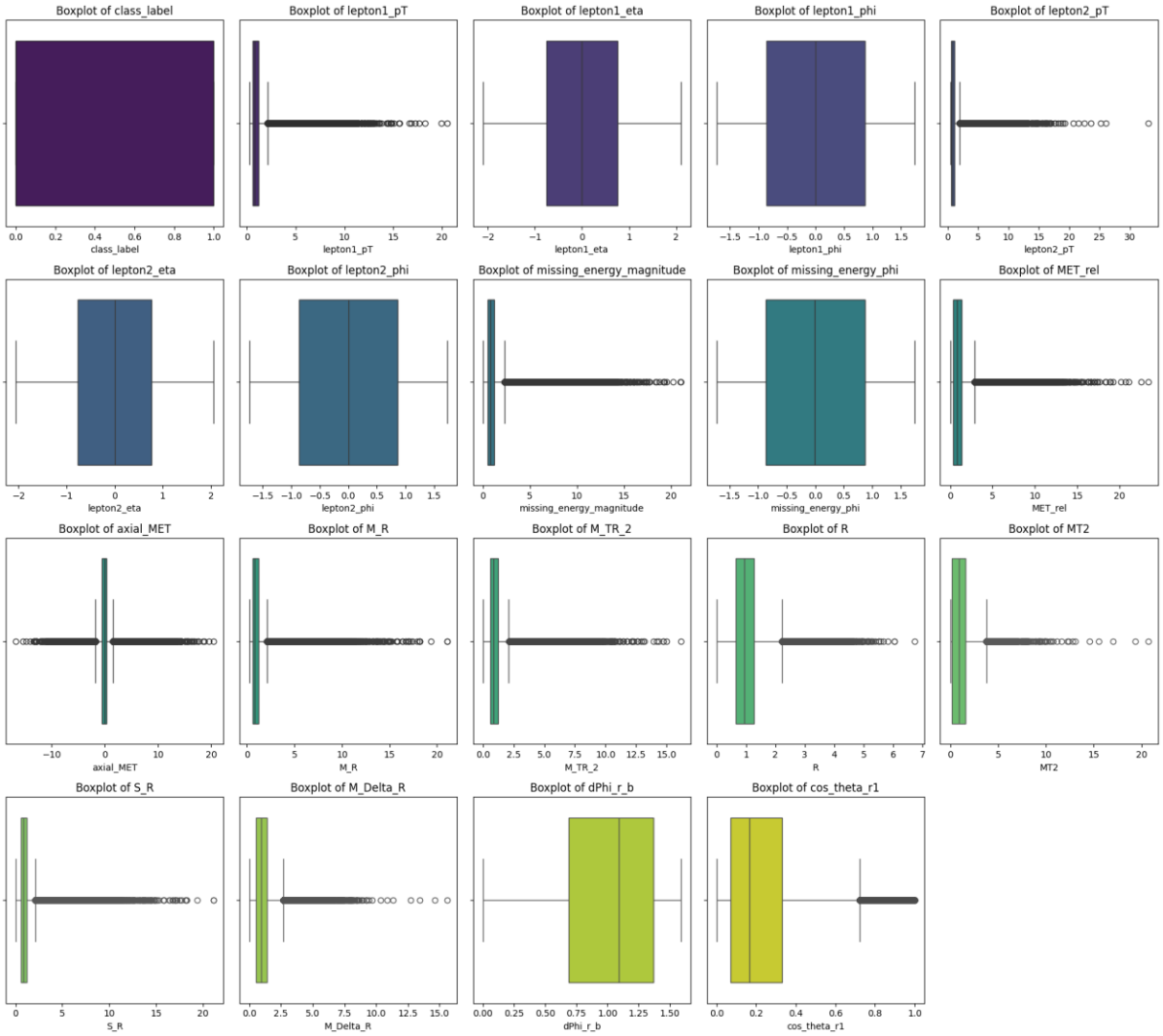

Figure 9: Box plots of features in SUSY dataset

1. The first plot ("class_label") appears as a solid box – This means that the variable is categorical, with only 2 unique values (e.g., binary labels like 0 and 1).
2. Most numerical variables show a long tail with outliers – Variables like lepton1_pT, lepton2_pT, missing_energy_magnitude, MET_rel, M_TR_2, R, MT2, S_R and M_Delta_R have many extreme values, suggesting that the data contains several large observations that fall beyond 1.5 times the IQR.
3. Some variables, like lepton1_eta, lepton1_phi, lepton2_eta, lepton2_phi and

missing_energy_phi have a symmetric distribution – Their boxes appear centered with whiskers extending evenly.

4. Several variables have very small IQRs – This suggests that most of the data is concentrated within a narrow range, but outliers extend beyond the whiskers.
5. A few variables, such as axial_MET and M_R, have values near zero – These might be transformations or computed features with specific ranges.

6. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a dimensionality reduction technique used in machine learning and statistics to transform high-dimensional data into a lower-dimensional form while preserving as much variance as possible. It is widely used in exploratory data analysis, feature extraction, and visualization. PCA works by identifying the principal components in a dataset, which are the directions along which the data varies the most. These components are found using linear algebra techniques such as Singular Value Decomposition (SVD) or Eigen decomposition. Steps in PCA:

1. Standardization: Since PCA is sensitive to scale, the data is first standardized (zero mean, unit variance).
2. Compute the Covariance Matrix: This matrix captures the relationships between different features.
3. Compute Eigenvalues and Eigenvectors: The eigenvectors represent the principal components, while the eigenvalues represent the amount of variance captured by each component.
4. Sort the Principal Components: Components are sorted based on their eigenvalues (variance contribution).
5. Select the Top Components: A subset of components is chosen to retain the most significant variance while reducing dimensionality.

- X-axis (Number of Components): Represents the number of principal components included in the model.
- Y-axis (Cumulative Explained Variance): Shows the cumulative proportion of variance explained by the selected principal components.

- Blue Dotted Line with Markers: Represents the cumulative explained variance as more components are added.
- Red Dashed Line (95% Variance Threshold): A common heuristic in PCA is to retain enough components to explain at least 95% of the variance. The intersection of this line with the cumulative variance curve indicates the minimum number of components required to reach this threshold.

6.1. PCA on HIGGS dataset

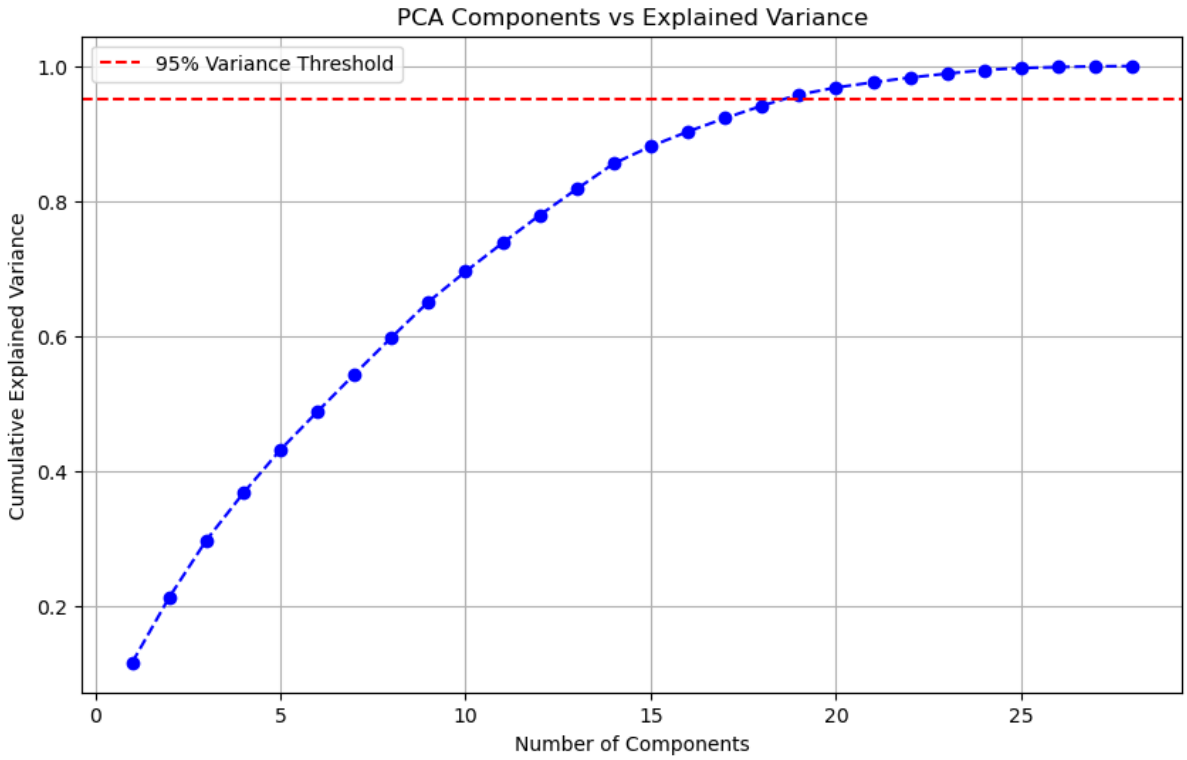

Figure 10: PCA of HIGGS dataset

The curve starts at a low variance and rises as more components are added. The curve flattens as it approaches 1.0, meaning adding more components contributes less additional variance. The red dashed line shows that around 19 components are sufficient to explain 95.76% of the variance. Beyond this, additional components contribute little to improving variance retention. Selecting only the required number of components (e.g., 19 instead of all 28) helps reduce dimensionality while retaining most of the useful information.

6.2. PCA on SUSY dataset

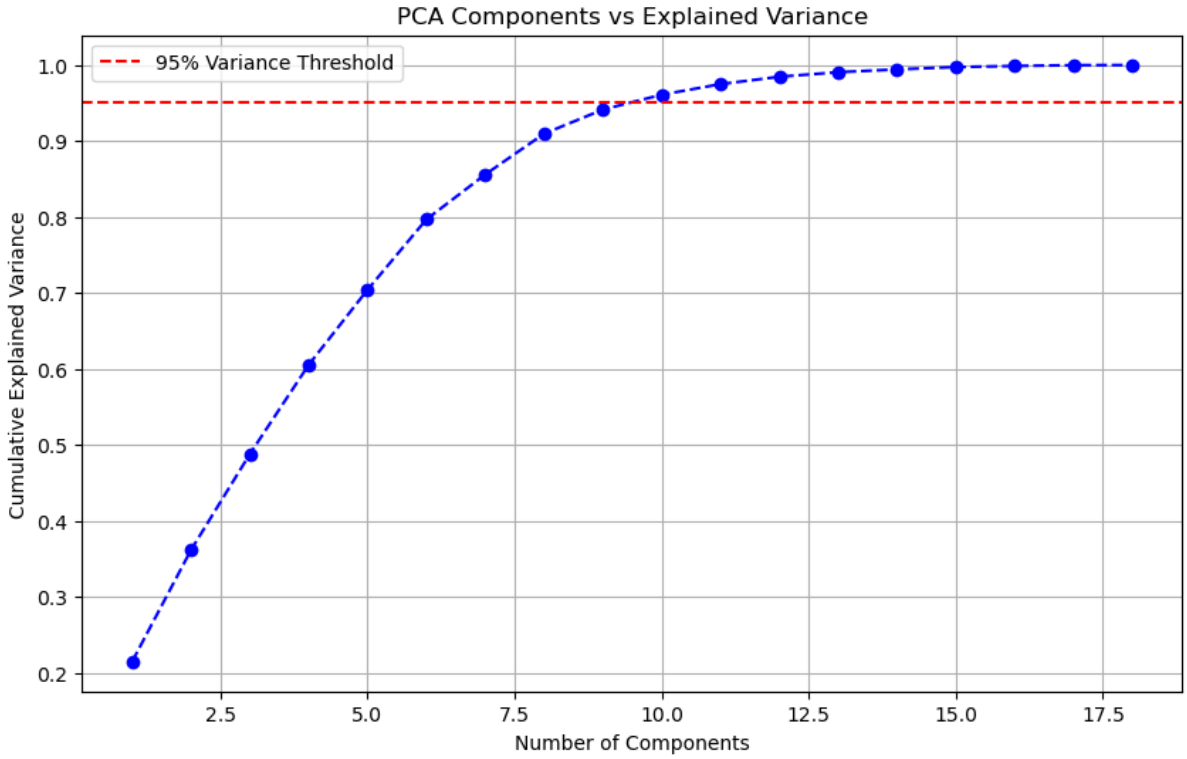

Figure 11: PCA of SUSY dataset

The curve starts at a low variance and rises as more components are added. The curve flattens as it approaches 1.0, meaning adding more components contributes less additional variance. The red dashed line shows that around 10 components are sufficient to explain 96.09% of the variance. Beyond this, additional components contribute little to improving variance retention. Selecting only the required number of components (e.g., 10 instead of all 28) helps reduce dimensionality while retaining most of the useful information.

To evaluate the effectiveness of feature dimensionality reduction, Random Forest classifiers were applied to three benchmark high-energy physics datasets- HIGGS, SUSY, and HEPMASS, both with and without Principal Component Analysis (PCA). The performance was assessed using two key metrics: test accuracy and ROC AUC. The results demonstrate that applying PCA prior to training generally led to a decrease in performance across all datasets. While PCA can be beneficial for reducing noise and computational cost in some

contexts, in this case, it appears to have discarded informative features crucial for distinguishing signal from background events. The comparison is summarized in the **Table 2: Comparison of performance of Random Forest with and without PCA** *Table 2.*

Dataset	PCA Applied	Test Accuracy	ROC AUC
HIGGS	No	0.7248	0.80
HIGGS	Yes	0.6574	0.72
SUSY	No	0.8002	0.87
SUSY	Yes	0.7667	0.84
HEPMASS	No	0.8574	0.94
HEPMASS	Yes	0.8017	0.88

Table 2: Comparison of performance of Random Forest with and without PCA

7. TSNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm used for dimensionality reduction, especially for visualizing high-dimensional data in a lower-dimensional space (typically 2D or 3D). It is widely used in exploratory data analysis to understand clusters and patterns in data.

How t-SNE Works:

1. Computes Pairwise Similarities: It calculates the probability distribution of pairwise similarities between data points in the high-dimensional space.
2. Reduces Dimensions: It maps these points to a lower-dimensional space while preserving local relationships.
3. Minimizes KL Divergence: The algorithm optimizes the layout so that similar points in high dimensions remain close in the 2D or 3D visualization.

7.1. TSNE on datasets

1. Scatter Plot Representation: The image is a 2D visualization of high-dimensional data reduced using t-SNE.

2. Color Coding (Blue & Orange): The data points are labeled into two categories (0.0 in blue and 1.0 in orange).
3. Clustering Pattern: The clusters indicate how data points with similar characteristics are grouped together.
4. Overlapping Distribution: Some orange (label 1.0) points are scattered within the blue (label 0.0) region, indicating some level of mixing between the two classes.

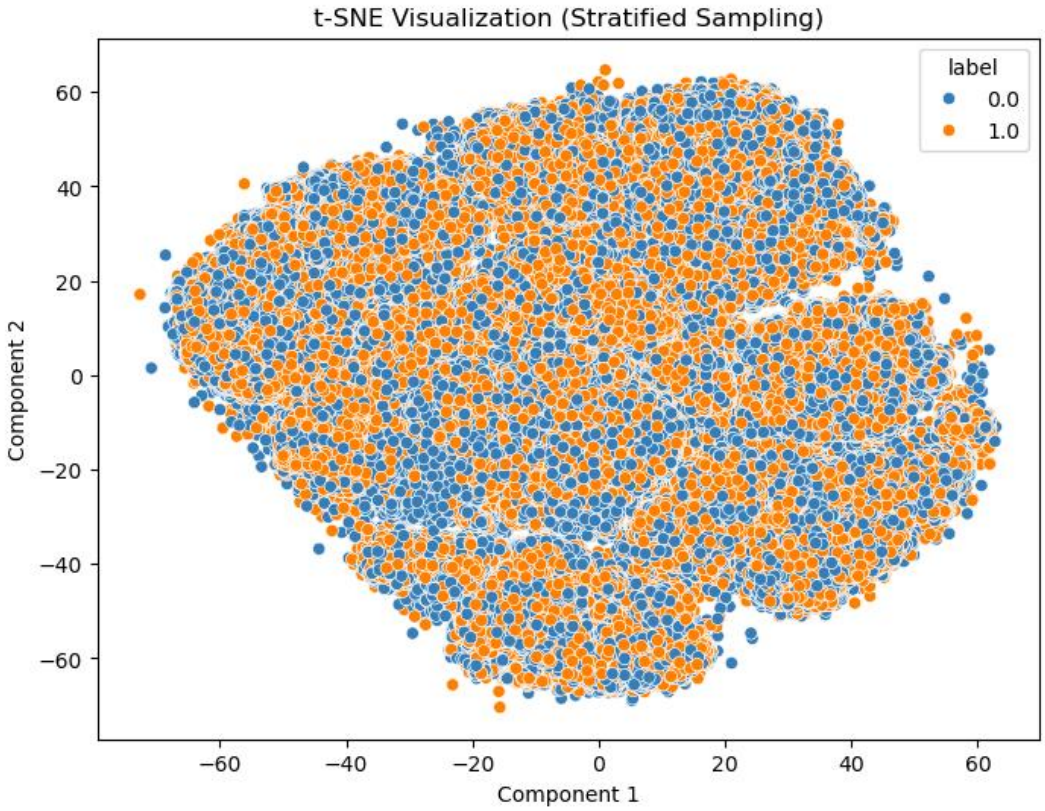

Figure 12: TSNE on HIGGS dataset

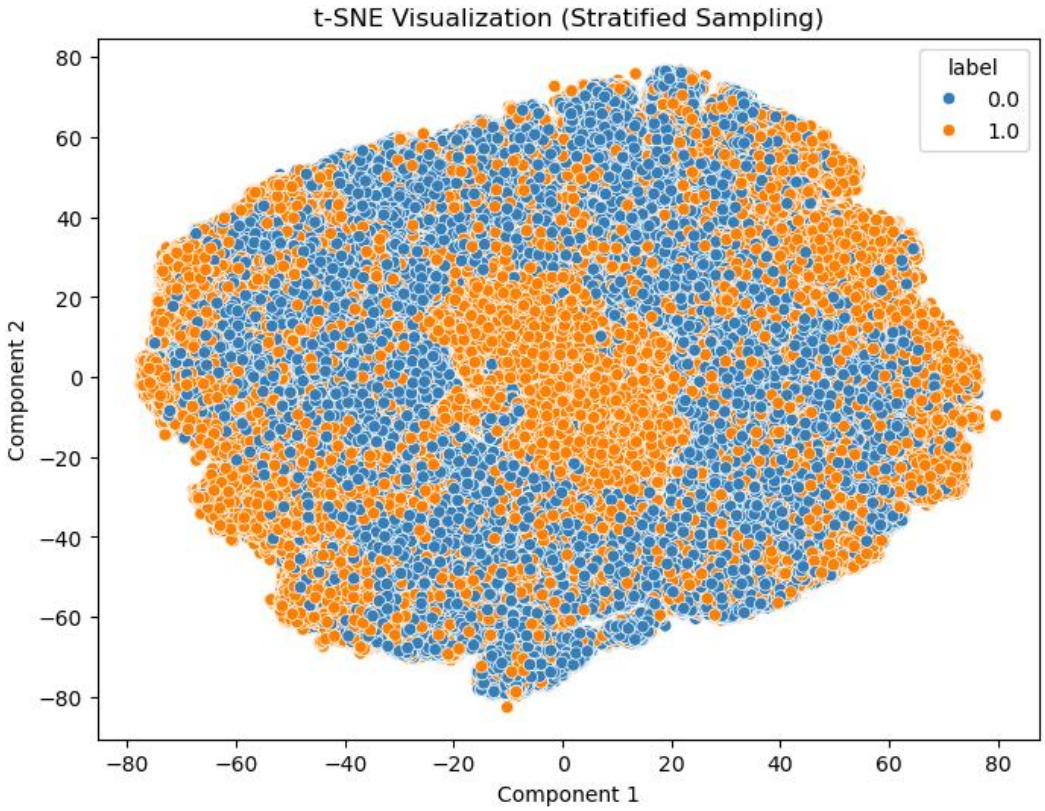

Figure 13: TSNE on SUSY dataset

8. COMPARATIVE ANALYSIS OF VARIOUS ALGORITHMS

Dataset	Model	Test Accuracy	ROC AUC
HIGGS	RF	0.7248	0.80
	LR	0.6416	0.68
	SVM	0.6644	0.72
	XGBOOST	0.7609	0.84
	NN	**0.7749**	**0.86**
SUSY	RF	0.8002	0.87
	LR	0.7880	0.86
	SVM	0.7958	0.87
	XGBOOST	0.8030	0.88
	NN	**0.8047**	**0.88**
HEPMASS	RF	0.8574	0.94
	LR	0.8353	0.92
	SVM	0.6772	0.78
	XGBOOST	**0.8762**	**0.95**
	NN	0.8641	0.95

Table 3: Comparative Analysis of various algorithms over all datasets

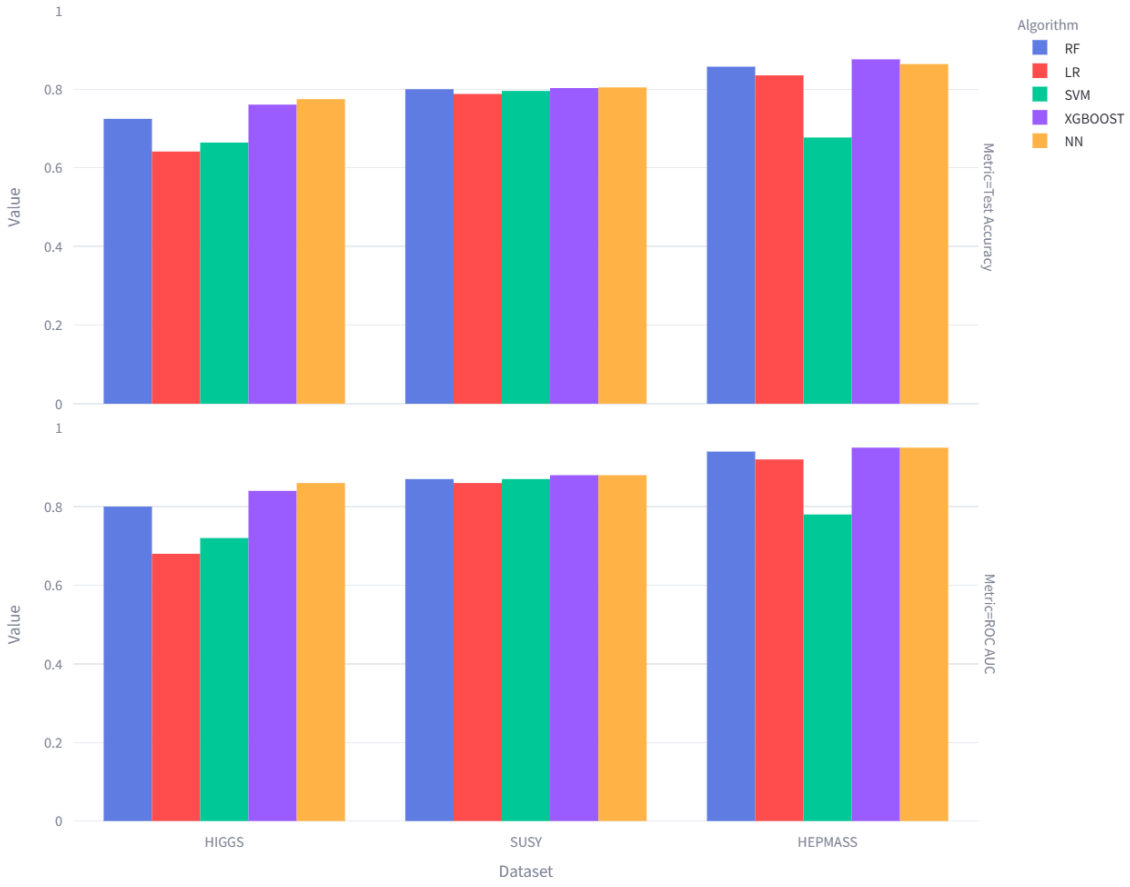

Figure 14: Bar chart showing comparison between various algorithms

From the comparative analysis of Random Forest and SVM across the HIGGS, SUSY, and HEPMASS datasets, we can draw the following conclusions:

- HIGGS: Neural Network (NN) achieved the best performance (Accuracy: 0.7749, ROC AUC: 0.86), followed by XGBOOST.
- SUSY: NN and XGBOOST led the results (Accuracy: ~0.804, ROC AUC: 0.88), while Random Forest and SVM showed similar but slightly lower metrics.
- HEPMASS: XGBOOST posted the highest accuracy (0.8762) and, along with NN, reached the top ROC AUC (0.95); SVM lagged significantly behind.

REFERENCES

[1] Nathan, "Mathematics behind ROC-AUC - Data Science | Machine Learning | by Nathan Aïm | Analytics Vidhya," *Medium*, Dec. 15, 2021. [Online]. Available: https://medium.com/analytics-vidhya/mathematics-behind-roc-auc-interpretation-e4e6f202a015

[2] "Why do physicists mention 'five sigma' in their results? | CERN," Mar. 18, 2025. https://home.cern/resources/faqs/five-sigma

[3] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Communications*, vol. 5, no. 1, Jul. 2014, doi: 10.1038/ncomms5308.

[4] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, "Parameterized neural networks for high-energy physics," *The European Physical Journal C*, vol. 76, no. 5, Apr. 2016, doi: 10.1140/epjc/s10052-016-4099-4.

[5] A. Alves, "Stacking machine learning classifiers to identify Higgs bosons at the LHC," *Journal of Instrumentation*, vol. 12, no. 05, p. T05005, May 2017, doi: 10.1088/1748-0221/12/05/t05005.

[6] M. Azhari, A. Abarda, B. Ettaki, J. Zerouaoui, and M. Dakkon, "Higgs Boson Discovery using Machine Learning Methods with Pyspark," *Procedia Computer Science*, vol. 170, pp. 1141–1146, Jan. 2020, doi: 10.1016/j.procs.2020.03.053.

[7] L. Anzalone, T. Diotalevi, and D. Bonacorsi, "Improving parametric neural networks for high-energy physics (and beyond)," *Machine Learning Science and Technology*, vol. 3, no. 3, p. 035017, Sep. 2022, doi: 10.1088/2632-2153/ac917c.

[8] M. Köppel *et al.*, "Learning to rank Higgs boson candidates," *Scientific Reports*, vol. 12, no. 1, Jul. 2022, doi: 10.1038/s41598-022-10383-w.

[9] V. D. Babu and K. Malathi, "Three-stage multi-objective feature selection with distributed ensemble machine and deep learning for processing of complex and large datasets," *Measurement Sensors*, vol. 28, p. 100820, Jun. 2023, doi: 10.1016/j.measen.2023.100820.

[10] K. Haritha, S. Shailesh, M. V. Judy, K. S. Ravichandran, R. Krishankumar, and A. H. Gandomi, "A novel neural network model with distributed evolutionary approach for big data classification," *Scientific Reports*, vol. 13, no. 1, Jul. 2023, doi: 10.1038/s41598-023-37540-z.

[11] "UCI Machine Learning Repository." https://archive.ics.uci.edu/dataset/280/higgs

[12] "UCI Machine Learning Repository." https://archive.ics.uci.edu/dataset/279/susy

[13] "UCI Machine Learning Repository." https://archive.ics.uci.edu/dataset/347/hepmass

[14] "Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014 | CERN Open Data Portal," 2014. https://opendata.cern.ch/record/328