Google DeepMind

AIMS | African Institute for Mathematical Sciences SOUTH AFRICA

Anomaly Detection In High Energy Physics Data

An essay submitted in partial fulfillment of a master degree in Artificial Intelligence (AI) For Science at African Institute for Mathematical Sciences (AIMS), South Africa

Author:

Beria Chingnabe KALPELBE

AIMS South Africa, South Africa

University of Stellenbosch, South Africa

Supervisor:

Dr. Daniel Murnane

University of Copenhagen, Denmark

Lawrence Berkeley National Laboratory, CA, USA

June 2024

Anomaly Detection In High Energy Physics Data

Béria Chingnabé KALPELBE (beria@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr. Daniel Murnane
University of Copenhagen,Denmark & Lawrence Berkeley National Laboratory, CA, USA

16 June 2024

Submitted in partial fulfillment of a master degree in AI For Science at AIMS South Africa

GitHub repository: beria-kalpelbe/anomaly-detection-in-HEP
Dataset: cernbox.cern.ch/s/rcm6wfMI1RDhp2B

Abstract

This study aims to develop a Machine Learning (ML) model to detect anomalous events in High Energy Physics (HEP) data. Despite extensive research in this area, experimental physicists still struggle to manage the huge amount of data collected at the Large Hadron Collider (LHC). Statistical tools like Z-scores are still widely used by experimental physicists. Many years ago, ML techniques demonstrated their ability to detect anomalies in various types of data, including HEP data. In this study, both supervised techniques (decision trees and multi-layer perceptron) and unsupervised techniques (autoencoders) are explored to detect anomalies in HEP data. The supervised learning methods are utilized to learn the characteristics that differentiate signal events from background events. On the other hand, unsupervised methods learn the distribution that generated background events in order to detect signals when they appear to deviate from the background distribution. The Decision Tree (DT) model is found to be the best anomaly detector with an accuracy of 72% and an Area Under Curve (AUC) of 81%. In addition, the most important track parameters in detecting signal events are mainly the transverse momentum (p_T), accounting for 58%, and the longitudinal impact (d_z), accounting for 23%. These results can assist experimental physicists in navigating the complexities of the LHC's massive datasets and identifying rare or unexpected collision events that could lead to groundbreaking discoveries in the search for new physics. On the other hand, despite showing good performance in learning the background distribution, the explored unsupervised learning techniques exhibit very low performance (accuracy of 53% and an AUC of 50%). This could indicate the inadequacy of their architecture for this task. Further research could explore more complex architectures such as Generative Adversarial Network (GAN) and normalizing flows.

Keywords: anomaly detection, Standard Model, high energy physics.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Béria Chingnabé KALPELBE, 16 June 2024

Table of contents

1. Introduction

Identifying patterns in large-scale data that deviate significantly from the norm, known as anomalies, has become increasingly important. As data volumes grow, organisations are challenged to distinguish meaningful insights from noisy data. Anomalies, which occur infrequently, can often elude standard analysis techniques and remain hidden within the vast data sets. Importantly, anomalies can reveal specific, previously unknown behaviours of systems under experimental conditions.

HEP is an example of a field characterized by the enormous volume of data collected. The LHC, an advanced experimental machine located at European Organization for Nuclear Research (CERN), is the largest scientific experiment globally (Evans and Bryant, 2008). Designed to replicate the conditions shortly after the Big Bang, the LHC produces approximately one billion particle collisions per second, generating roughly one petabyte of collision data each second. As of June 29, 2017, the LHC had passed the milestone of 600 petabytes of data, a volume equivalent to over 20,000 years of continuous HD video recording (Gaillard, 2017).

The best theory we currently have to describe the data collected at the LHC is the Standard Model (SM), introduced by Abraham Pais and Sam Treiman in 1975. The SM outlines how 12 fundamental particles interact through three fundamental forces including electromagnetic, strong nuclear, and weak nuclear forces (Salam, 1994). However, the theory does not incorporate the theory of general relativity, which describes the gravitational force. This omission remains a significant gap in our understanding of the universe.

By colliding protons at extremely high energies, the LHC allows scientists to observe new particle behaviors and interactions that have never been seen before. Physicists call these new particle behaviors "new physics". The goal is to uncover insights that could lead to a more complete theory of fundamental particles, and possibly even integrate gravity with the other fundamental forces described by the SM (Evans and Bryant, 2008).

The search for new physics at the LHC can be seen as a problem of finding anomalies. The goal is to determine whether it is possible to identify events that exhibit behaviors that differ significantly from the patterns predicted by the SM. These unusual events could indicate the existence of previously unseen particles or interactions.

This study aims to address the following research questions:

- How effective are unsupervised or semi-supervised learning models at detecting anomalous events in LHC data?
- How can these models be optimized for real-time Anomaly Detection (AD) in the high-throughput environment of the LHC?

The main objective of this study is to develop an intelligent tool able to detect anomalous events in the LHC. More specifically, to:

- Create and train robust Artificial Intelligence (AI) models capable of identifying anomalous events in LHC data with high accuracy, minimizing false positives and false negatives;
- Improve the sensitivity of AD to identify rare events that could indicate new physics phenomena beyond the SM;

- Ensure that the proposed models generalize well across different types of data and experimental conditions to reliably detect anomalies under various scenarios.

This report is presented in five chapters: the chapter 2 provides a theoretical background and some implementations of AD in HEP; the chapter 3 describes the methodology followed in this study; the chapter 4 presents the results which are discussed in the chapter 5. The chapter 6 concludes the report with a discussion of a potential future work.

2. Literature review of anomaly detection in HEP

2.1 Theoretical background: the SM

Four hundred years ago, Galileo embarked on a quest to develop a foundational understanding of reality, which has since evolved into what we now recognize as modern science (Barboianu, 2022). He grappled with a question as old as humanity itself: "*what is everything made of*"? In other words, what are the basic building blocks of the universe from which everything is constructed? Over the past century, numerous theories and experiments have been conducted at increasingly smaller scales, all striving to create a coherent picture of the structure of matter. The culmination of these efforts is the SM, the most successful scientific theory to date. This model emerged from the collaborative contributions of many scientists around the world during the latter half of the 20th century.

The origins of the scientific understanding of the universe can be traced to early attempts to explain the phenomenon of attraction between opposite magnetic poles and repulsion between similar magnetic poles. These observable interactions are facilitated by an underlying magnetic force acting between magnetic materials. The concept of a magnetic field is utilized to model the spatial distribution of this magnetic force around and within magnetic substances (figure 2.1). More broadly, the notion of a field can be generalized as a region of space containing a particular type of particle (Serway and Jewett, 2004).

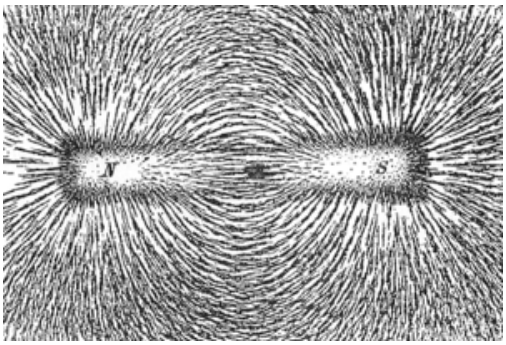


Figure 2.1: Illustration of the magnetic field

The SM is an actual description of the building blocks of the universe and interactions within them in a Quantum Field Theory (QFT) framework, a mathematical framework that combine Classical Field Theory (CFT), special relativity and quantum mechanics.

The SM of Particle Physics describes the fundamental constituents of matter in the universe, which is composed of twelve distinct types of particles that interact with three fundamental forces. Many of the particles interact with a unique particle, the Higgs boson, which plays a pivotal role in the model (figure 2.2).

Each particle is either a fermion (matter particle) or a boson (force particle). The distinction between them lies in the quantum world. Fermions obey the Pauli Exclusion Principle (PEP), which states that it is not possible to put two fermions on top of each other in space. They are the building blocks of matter. On the other hand, bosons are not constrained by the PEP, so can occupy the same location or state.

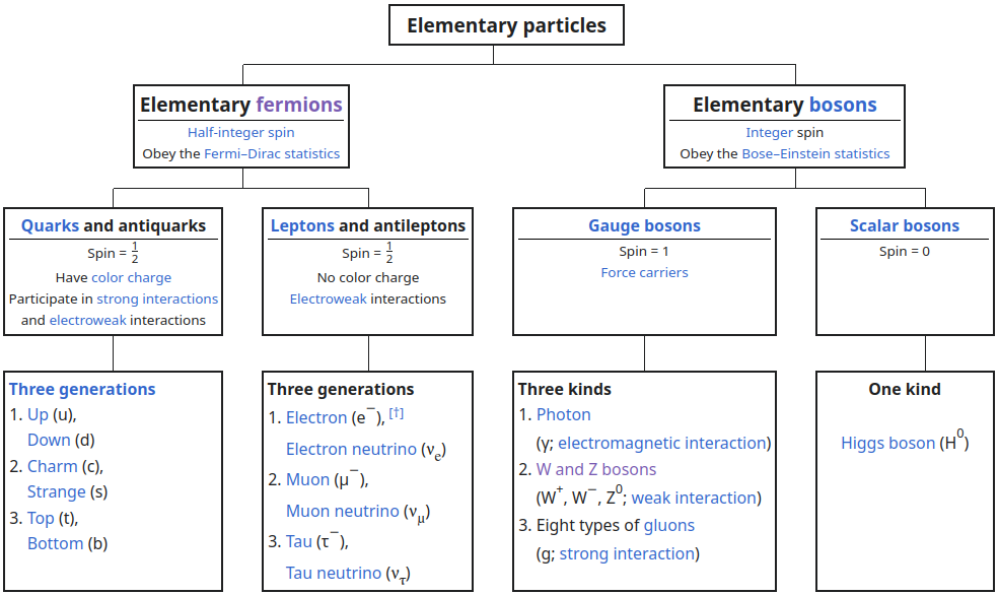


Figure 2.2: The fundamental elements of matter in the SM, Wikipedia

Every fermion is accompanied by an antiparticle with an opposite charge. These particles are classified into two groups: quarks and leptons. In each of these groups, the pair of particles is grouped into generations (see figure 2.2). The atoms in everyday life are made of first-generation particles: electrons orbit collections of up and down quarks, arranged into protons and neutrons. The second and third generations of fundamental particles are mostly observed in high-energy environments.

Boson particles can be classified as either gauge bosons or scalar bosons. Gauge bosons are defined as force carriers because they are responsible for mediating the fundamental interactions. There are four gauge bosons (photon, gluon, W± and Z) for electromagnetism, weak and strong interactions.

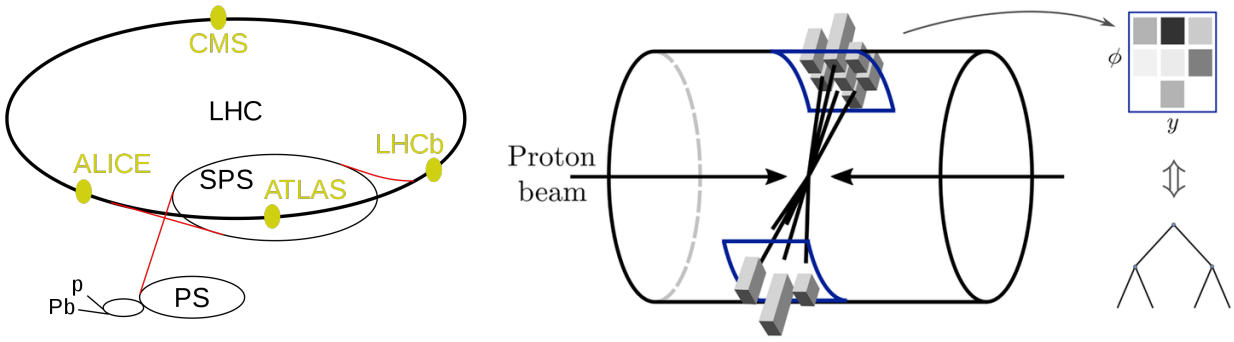
There is one fundamental scalar boson: Higgs Bosons, which play a unique role in the SM.Particles obtain their mass by interacting with Higgs fields. This phenomena is known as the Brout-Englert-Higgs mechanism proposed by Robert Brout, Francois Englert and Peter Higgs.

There are two reasons why general relativity is not compatible with the SM. Firstly, at the microscopic level, the force of gravity is weak and does not have an effect on a single subatomic particle. The second reason is that incorporating general relativity is challenging because it is “a classical theory in a quantum world”.

2.2 Overview of the LHC

The LHC is the world’s largest and most powerful particle accelerator, situated at the CERN near to Geneva, Switzerland. The facility was designed to collide particles at close to the speed of light in order to explore the fundamental nature of matter and the basic forces that shape our universe. The accelerator operates by accelerating two protons in opposite directions around a circular tunnel measuring approximately 27 kilometres in circumference (Evans and Bryant, 2008). The protons are directed to collide at four principal points, where they produce conditions analogous to those that prevailed immediately following the Big Bang (figure 2.3b). This allows scientists to observe

and study rare particles and phenomena.



(a) CERN accelerator complex (b) Illustration of ATLAS detector

Figure 2.3: The LHC experiments

There are four main experiments at the LHC including(figure 2.3a):

- ATLAS and CMS, designed to cover a wide range of physics, including the search for Higgs Boson and signs of new physics beyond the SM.
- ALICE, to study the quark-gluon plasma, a state of matter thought to have existed shortly after the Big Bang.
- LHCb, with a focus on understanding the differences between matter and antimatter, specifically looking at particles containing a type of quark called the "beauty" or "bottom" quark.

2.3 Some past implementations of anomaly detection in HEP

The anomaly detection task can be formulated as a classification problem, which has been the subject of extensive research in the literature. Existing works in this domain have primarily employed two main approaches: supervised learning and unsupervised learning. In the supervised learning approach, the anomaly detection task is treated as a classification problem. Conversely, the unsupervised learning approach to anomaly detection does not rely on labeled data, but rather aims to identify patterns and deviations within the data without the guidance of pre-existing classifications.

The most popular supervised ML algorithm used is the Boosted Decision Tree (BDT). They are used to separate rare signals from a large set of background particles or to identify physical objects in the detectors (Coadou, 2022). They happened to perform better than the matrix element calculation and Bayesian neural networks. Other ML models, such as Neural Network (NN) and Deep Neural Network (DNN), are also used in the literature to separate signal from background (Coadou, 2022).

A comprehensive review of the LHC Olympics 2020, a community challenge focused on anomaly detection in high energy physics, has been produced (Kasieczka et al., 2021). It details a competition where participants developed methods for anomaly detection using provided datasets. Various approaches, including unsupervised, weakly supervised, and (semi)-supervised methods, are discussed. For example, the Variational Recurrent Neural Network (VRNN) is used for anomaly detection of jets by Kasieczka et al. (2021). Basically, jets are considered as sequence of constituents. The input is a sequence of jets, comprising p_T, η and ϕ. A preprocessing step has been applied so as to preserve the structure of inputs. As for a standard Variational Auto-Encoder (VAE), the

loss is composed of a Mean Square Error (MSE) between the input and output, and a weighted Kullback–Leibler Divergence (KLD) (See equation 2.3.1).

$$\mathcal{L}(t) = MSE + 0.1 \times \bar{p_T}(t) D_{KL} \tag{2.3.1}$$

Each jet is then assigned an anomaly score defined by $1 - e^{-\bar{D}_{KL}}$, where D_{KL} is the KLD between the learned prior distribution and the encoded posterior distribution. So, a score closer to 1 indicate more anomalous events.

Anomaly detection through estimating the density of data points is also applied. Know as ANOmaly detection with Density Estimation (ANODE) proposal, it consist of learning two densities using conditional neural density estimation: $p_{\text{data}}(x|m)$ and $p_{bkg}(x|m)$ for m in the Signal Region (SR), the classification is then done by the likelihood ratio with $\alpha \in [0, 1]$:

$$R(x|m) = \frac{p_{\text{data}}(x|m)}{p_{\text{bkg}}(x|m)} = \frac{\alpha p_{\text{bkg}}(x|m) + (1-\alpha) p_{\text{sig}}(x|m)}{p_{\text{bkg}}(x|m)} \tag{2.3.2}$$

Belis et al. (2024) also proposed this method as an anomaly detection technique ,called over-density estimation.

The technique known as the Bump Hunting in Latent Space (BuHuLaSpa) is used to detect anomalies in a transformed feature space. It assumes that events are generated by a stochastic process, which is described by a stochastic generative model. The likelihood and posteriors are approximated using NNs. In most of the classification-based ML models, the momentum-based optimizers (like Adam) tend to perform well. However, as demonstrated by Kasieczka et al. (2021), this is not the case with VAEs. Adagrad or Adadelta optimizers peform much better in learning effective latent representation of the data with a small size of anomalous signal events.

The Generative Adversarial Network-based Auto-Encoder (GAN-AE) approach, proposed by Vaslin et al. (2023), enables both anomaly detection and model-independent background modelling. In contrast to the approach proposed by Belis et al. (2024), which utilises the reconstruction loss (MSE) of a VAE as an anomaly detection score, Vaslin et al. (2023) propose an alternative metric based on a supervised discriminator network trained to classify reconstructed events (labelled 0) and original events (labelled 1) (Figure 2.4). The anomaly score is then defined as the following Euclidean distance:

$$D(y, \hat{y}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{2.3.3}$$

Where y is the input, $\hat{y}$ the output and N their dimension.

Particle Graph Autoencoders (PGAE), introduced in the literature by Tsan et al. (2021), can be used for unsupervised detection of new physics. According to Kasieczka et al. (2021), this architecture can make Auto-Encoder (AE)s learn a good compressed representation of a jet and then identify anomaliess Beyond the Standard Model (BSM) signal events from LHC. As shown in figure 2.5, each input jet is a graph containing particles as nodes. Each particle (node) has 4 components representing the components of momentum in play (E, p_x, p_y, p_z).

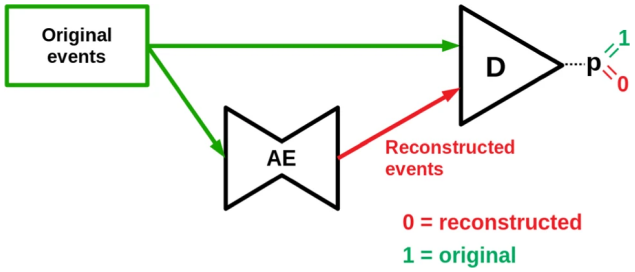


Figure 2.4: Architecture of the GAN-AE. Firstly, an Auto-Encoder (AE) is trained to reconstruct the original events. At the same time,a discriminator D is trained to distinguish reconstructed events from original events.

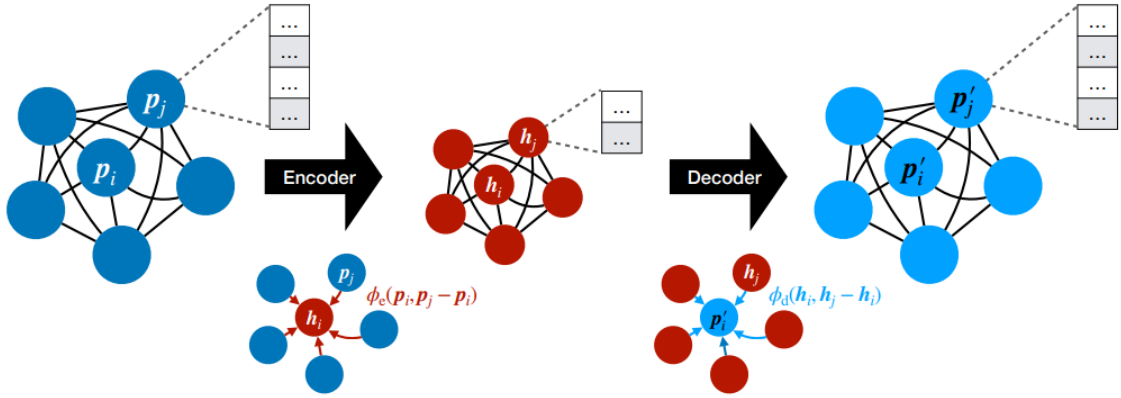


Figure 2.5: Architecture of the PGAE proposed by Kasieczka et al. (2021).

The loss function optimized in the process of learning the PGAE has 2 components: MSE between input jet and output jet and the Chamfer distance loss. The last is introduced to consider the order of reconstruction. The general loss function is then given by the following equation:

$$D^{\mathrm{NN}}(\mathcal{M},\mathcal{N}) = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}}\min_{j\in\mathcal{N}}\left(\left\|\boldsymbol{p}_i-\boldsymbol{p}_j\right\|\right)^2 + \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}}\min_{i\in\mathcal{M}}\left(\left\|\boldsymbol{p}_i-\boldsymbol{p}_j\right\|\right)^2 \tag{2.3.4}$$

Where $\mathcal{M}$ and $\mathcal{N}$ represent the input jets and the reconstructed jets, respectively.

3. Methodological approach for anomaly detection in HEP experiments

3.1 Data simulation, software and hardware configurations

The data was simulated using the framework Madgraph (Alwall et al., 2014). Track parameters were simulated, including: azimuthal angle(ϕ), transverse (p_T), pseudo-rapidity (η), transverse impact parameter (d_0), and longitudinal impact parameter (d_z).

- The transverse momentum(p_T) is the component that is perpendicular to the beam axis(figure 3.1) defined by:

$$p_T = P \sin(\theta_{cm}) \tag{3.1.1}$$

  Its invariance under Lorentz Boosts[1] along the beam axis make it a crucial component. A high value of p_T often indicates interesting physics events, since low values of p_T can come from background events. p_T is measured in units of mass multiplied by speed ($kg \cdot m/s$).

- The azimuthal angle(ϕ) around the beam axis (figure 3.1) is measured in the plane perpendicular to the beam. It provides the angular position of the particle in the plane and is used with p_T to describe the motion of the particle(figure 3.1)

- The peudo-rapidity(η) describes the angle of a particle relative to the beam axis. It is defined as:

$$\eta = -\ln\left[\tan\left(\frac{\theta_{cm}}{2}\right)\right] \tag{3.1.2}$$

  The differences in pseudo-rapidity are Lorentz invariant under boosts along the beam axis. It's then used for characterizing particle production in hadron colliders. High values of η are produced close to the beam axis.(figure 3.1)

- Transverse impact parameter (d_0) is the distance at which the trajectory of a particle comes closest to the primary vertex in the transverse plane. It indicates the extent to which the particle's trajectory deviates from the primary interaction point.

- Longitudinal impact parameter(d_z) is the distance along the z-axis from the primary vertex to the point where the particle crosses the transverse plane. As d_0, d_z provides information about the particle's production point relative to the primary vertex along the beam direction.

In the present work, a set of ML models are implemented in Python 3.10 (Section 3.2). The deep learning library utilized is PyTorch version 2.1. The computational infrastructure employed comprised a Google Cloud Virtual Machine with a 200GB hard disk, 60GB of RAM, and a single NVIDIA T4 GPU with 15GB of dedicated memory.

[1] A Lorentz boost specifically refers to transformations involving relative motion along one of the spatial axes (typically the x-axis).

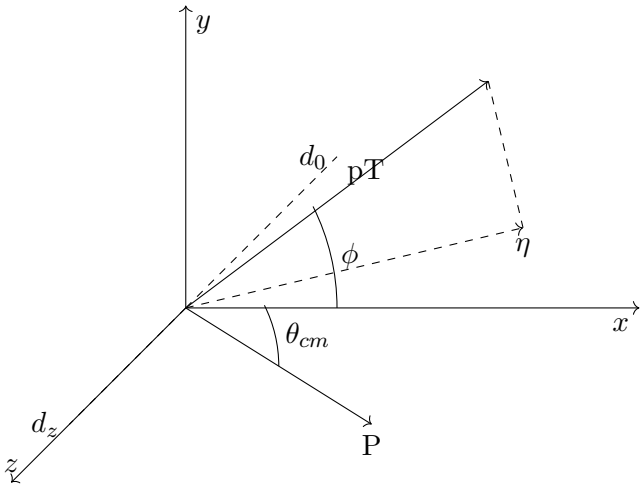


Figure 3.1: Components of momentum

3.2 ML-based anomaly detection in HEP

The existing methods for AD are very diverse as presented in the previous chapter. In the supervised learning approach, the anomaly detection task is treated as a classification problem, where labeled data are utilized to train a predictive model. In this scenario, the available dataset comprises events that have been previously identified and categorized as either normal or anomalous. The trained model can then be employed to classify new, unlabeled data points into these predefined classes, effectively detecting anomalies within the dataset. However, the unsupervised learning approach to anomaly detection does not rely on labeled data. In this case, the task is to identify patterns and deviations within the data without the benefit of pre-existing classifications. This approach is particularly useful when only a small portion of the data is labeled, or when the labeling process is resource-intensive or impractical. The unsupervised methods aim to discover anomalies by exploiting the inherent structure and relationships within the data, without the guidance of labeled examples.

3.2.1 Supervised learning approach

With labeled events, the AD task become a simple classification task. Since, many algorithms can be applied for this task. In this study, we will be focus on Decision Tree and Multi-Layer Perceptron (MLP).

Let us consider a binary classification problem, with two classes $\mathcal{K} = \{0, 1\}$, defined in a set of features $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_n\}$ and each feature $\mathcal{F}_i$ takes values from a categorical domain D^2. The feature space is defined by $\mathbb{F} = D_1 \times \cdots \times D_n$ representing the range of categorical values for each feature. An arbitrary point in the feature space is then represented by $x = (x_1, \ldots, x_n)$. A ML model then computes a function μ that maps the feature space into the set of classes $\mu : \mathbb{F} \leftarrow \mathcal{K}$.

A DT $\mathcal{T}$ is a directed acyclic graph having at most one path between every pair of nodes (Izza et al., 2020). All edges have one incoming edge (figure 3.2). It consists of three (03) types of nodes:

1. Root node - the decision process start with a test over the entire dataset
2. Internal nodes - a box evaluating a subset of the dataset on a given feature, and have only two outcomes(branchs)

3. Leaf nodes - the nodes that don't split further, indicating the final classification of data points .

To estimate the mapping function μ, the DT uses an algorithm called Classification And Regression Tree (CART). Intuitively, it uses the divide and conquer strategy via a greddy search to find the optimal split points within the tree. When considering optimality concept, we need to optimize a cost function. This cost function is defined by:

$$J(k, t_k) = \frac{N_{left}}{N_{node}} G_{left} + \frac{N_{right}}{N_{node}} G_{right}. \tag{3.2.1}$$

Where the index "left" and "right" represent, repectively, data points whose x_k i such that $x_k < t_k$ and $x_k \geq t_j$. G represents the Gini's diversity index defined for each node k by:

$$G_k = \sum_{i=1}^{N_{\text{classes}}} \left(\frac{N_{ki,k}}{N_k} \right)^2 = 1 - \sum_{i=1}^{N_{\text{classes}}} (p_k)^2 . \tag{3.2.2}$$

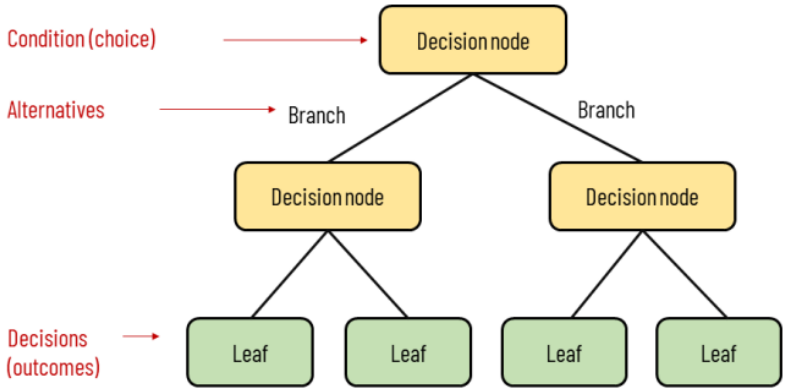


Figure 3.2: Illustration of a Decision Tree (Kosarenko, 2021).

The CART algorithm is a sequential algorithm where:

1. Inputs are training data $X = (x_1, \ldots, x_n)$ of n samples associated with their target y
2. Hyperparameters are maximum number of depth, minimum sample split, minimum sample per leaf, maximum leaf nodes
3. Outputs are collection of decision boundaries segmenting the k feature space.
4. Initialization take place at the root node.
5. Steps of the algorithm are:
   (a) For each feature k and for a value of t_k scanning the range $x_k^{\min}$ to $x_k^{\max}$:
      i. Compute the cost function $J(k, t_k)$
      ii. Find the pair (k, t_k) that produce the purest subsets (minimizing the cost)
   (b) Split the data set along the feature k using the threshold t_k into two new nodes
   (c) Repeat (5.a) at each new node

6. Exit conditions require at least one of the conditions of the hyper-parameters is satisfied.

The CART algorithm posses some limitations.DT choose the thresholds only on the available dataset. So without any constraints, it will continue cutting through the data noise and then leading to over-fitting. In addition, the boundaries of a DT are always perpendicular to an axis and doesn't correspond to certain configurations of the data. Finally, DT are very unstable because it can change drastically with only minimal modification of the data.

In order to produce a robust AI model for AD, a more flexible supervised learning model, MLP, is also explored. It is a type of feedforward neural network consisting of fully connected neurons with a non linear activation function (figure 3.3).

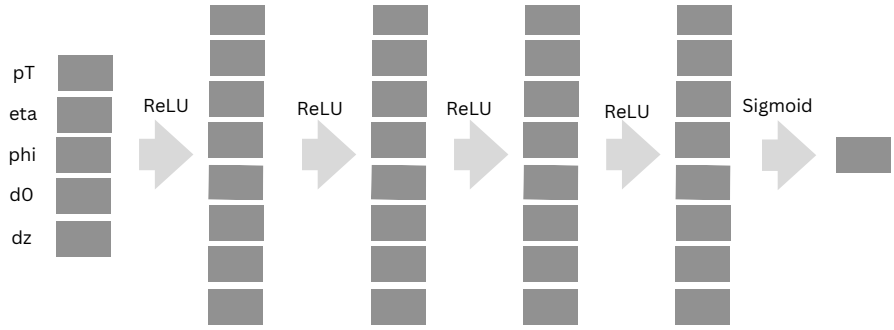


Figure 3.3: Architecture of the MLP containing four hidden layers of eight nodes. The five components of the momemtum represents the input of the network. The output represent the probability $P(\text{Signal}|pT, \eta, \phi, d_0, d_z)$ for the event to be signal. For the nodes of hidden layers, Rectified Linear Unit is used as activation function and the output layer use the sigmoid function as activation function.

In the other hands, having the knowledge of events being signal or background is mostly met in simulated datasets. Most of the cases, only few information are available about the anomaly of events. Since, supervised learning become not a very good choice for AD. A technique called unsupervised learning allow using both labeled and unlabeled events to build a model and will be crucial in this study.

3.2.2 Unsupervised learning approach: auto-encoders

Self-supervised learning (Gui et al., 2023) is a ML technique where the data provide supervision. One part of the data is used for training a ML model and the task is to reproduce this data. Since, the model learn the distribution and produce a meaningful representation of the data. The auto-encoder architecture is widely used for this task in HEP.

An AE network (Bank et al., 2021) consist of two networks: encoder network and decoder network. The encoder is designed to encode the input ($\mathbb{R}^5$) into a meaningful representation and a small space ($\mathbb{R}^4$) called latent space. The decoder, in the other hand, is responsible to decode back the latent space such that the reconstructed input is close as possible to the input (fig. 3.4).

The AE model is trained on background events so that it learn the distribution of the background events. The loss used is the MSE. Since then, any events from the same distribution (background) is expected to have a lowest error compared to the events coming from a different distribution

(signal). An anomaly score is then defined for every event i as following:

$$S_i = -\log\left[\frac{1}{5}\sum_{j=1}^{5}(x_j - \hat{x}_j)^2\right] \tag{3.2.3}$$

Then, the signal events are expected to give a small score compared to background events. In practice, it's common in ML to define a threshold with which, the score is compared to decide either signal or background event.

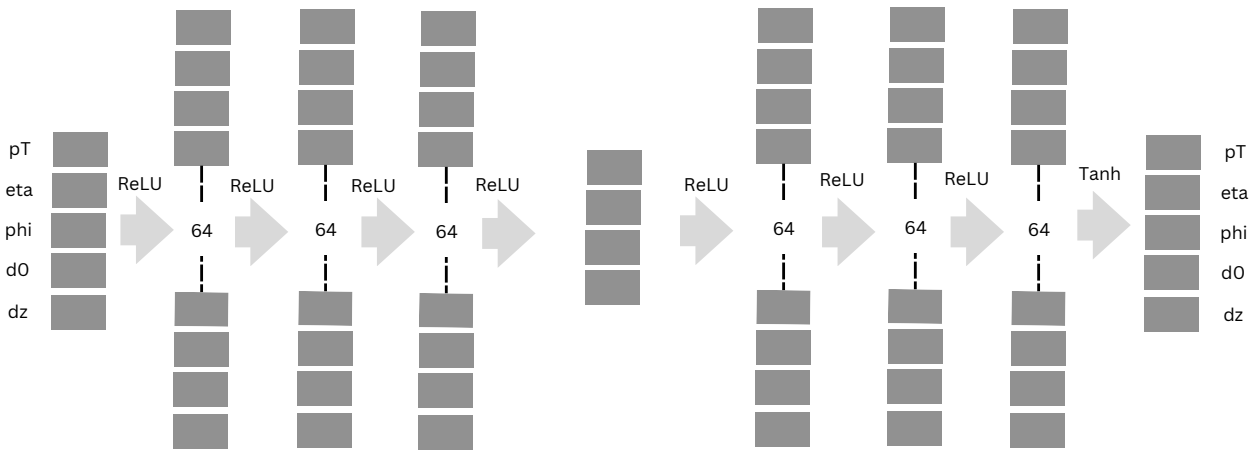


Figure 3.4: Architecture of the AE model. The input layer is constituted by the 5 components of the momentum $(pT, \eta, \phi, d_0, d_z)$. The encoder and decoder networks are constituted by 3 layers of 64 neurons using ReLU as activation function. As input layer, the output layer contains 5 neurons and is activated by a tanh function ensuring that outputs are between -1 and +1 as inputs.

Knowing the limits of the AE architecture, the one of bias-variance trade-off [2] , a VAE architecture is then implemented. The architecture is similar to the AE (figure 3.4). The particularity of VAE is that it models the latent representation to approximate a given probability distribution (Gaussian in the case of this study). This distribution is described by two parameters: mean and variance. Since then, a latent vector is sampled from the latent distribution and decoded every time that an input is encoded. The decision process is similar to the one for AE model.

[2]The bias-variance trade-off of AE representation is about three type of limits: firstly, it is not possible de sample a point for generation for an AE model; secondly, some labels can be represented over small areas and then lack of diversity; and finally, some generated events might be poor because of gaps between type of events in the latent space.

4. Empirical findings on anomaly detection techniques applied to HEP data

4.1 Dataset

The dataset is constituted of 12.5 million events where 50% are background events and the other half are signal events.

4.1.1 Data description

The statistical indicators of central tendency and dispersion, as shown in the table 4.1, provide an overview of the information contained in the data. These are mainly the range of the data, the mean and the standard deviation.

	Entire Dataset				Signal Dataset				Background Dataset			
	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std
p_T	0.06	881.06	1.16	2.97	0.23	217.70	1.18	2.47	0.06	881.06	1.13	3.40
η	-9.32	9.40	-0.03	2.25	-9.32	9.40	-0.07	2.85	-2.68	2.92	-0.00	1.42
ϕ	-3.14	3.14	0.00	1.81	-3.14	3.14	-0.00	1.81	-3.14	3.14	0.00	1.81
d_0	-1310.36	898.29	-0.04	8.94	-605.89	582.19	-0.08	10.26	-1310.36	898.29	-0.00	7.40
d_z	-5733.48	1973.11	-0.12	51.01	-654.34	661.48	-0.22	45.49	-5733.48	1973.11	-0.02	55.99

Table 4.1: Central tendency and dispersion of the data

The transverse (p_T) vary from 0 to 881.06 showing the average value of 1.16 and a standard deviation of 2.97. The distribution of its density is similar to Log-normal distribution (see figure 4.1) because the density is high for lower values of p_T and smoothly low for high values of p_T. The disparity of signal transverses (standard deviation: 2.47) is slightly lower than the background ones (standard deviation: 3.40).

The distribution of the pseudo-rapidity (η) seems almost symmetric around 0 and vary from -9.3 to +9.4 (see figure 4.1). The signal pseudo-rapidities seems follow a normal distribution since the background ones seems follow uniform distribution between -2.9 and +2.9. The background particles seems to moves close to transverse to the beam axis because the values of p_T are close to 0, since the background particles can moves far from the transverse as well but mostly close to transverse.

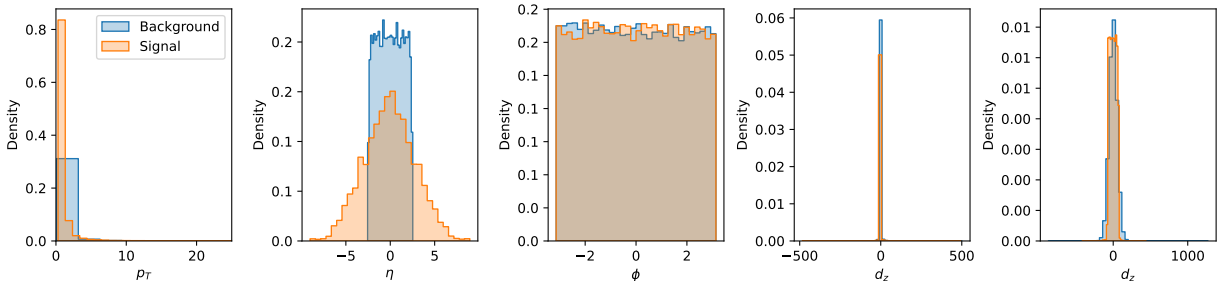


Figure 4.1: Density distributions of pT, η, ϕ, d_0 & d_z

The azimuthal angle (ϕ) seems follow a uniform distribution between $-\pi$ and $+\pi$. Whether the particles are signal or background, the azimuthal angles are almost the same in their structure (see figure 4.1). On the other hand, the parameter of transverse impact (d_0) is very close to 0 for most of the cases. Its distribution is more homogeneous when it comes from background (std:7.4) than signal (std: 10.26). Finally, the parameter of longitudinal impact (d_z) seems follow normal distribution of mean -0.12 and standard deviation 51. From background or not, the particles show almost the same values of d_z.

4.1.2 Data processing

The processing step involves transforming distributions, normalizing data, and splitting the data for training, validation and testing. The distributions of p_T, d_0 and d_z initially seem challenging for a machine learning model to learn due to their shapes. However, as shown in Figure 4.2, the histograms of these feature p_T in logarithmic scale appear much closer to normal distributions compared to its original form (see figure 4.1). Therefore, applying a log-transformation to p_T is appropriate.

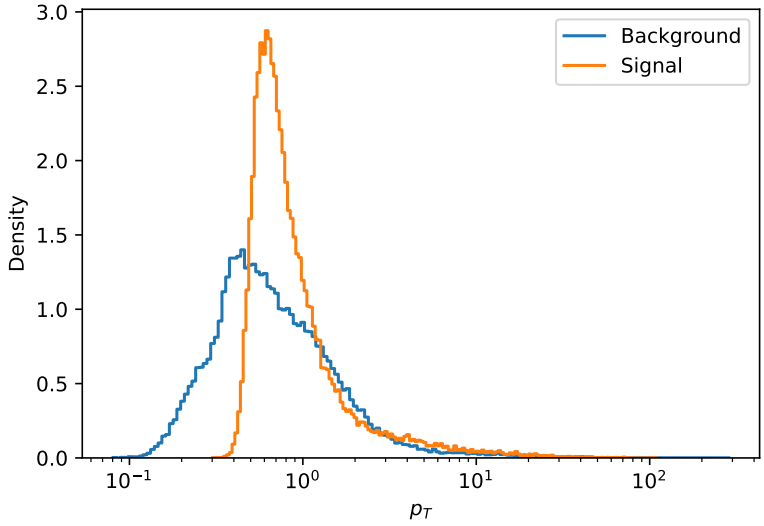


Figure 4.2: Density distribution of p_T in log-scale

Among the most popular machine learning models, such as neural networks, the scale of the input data can significantly affect convergence. To address this, a normalization is applied to all features to ensure their values range between -1 and +1. This is achieved using the following equation for a feature x_i:

$$x_i' = -1 + 2 \times \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}. \tag{4.1.1}$$

On the other hand, the dataset is randomly split into three groups: training set, validation set, and testing set using the framework Scikit-learn. The training set contains 80% of the data (10 million events), with a composition of 51% background events and 49% signal events. Both the validation and testing sets each represent 10% of the data, maintaining almost a similar distribution of events as the training set.

4.2 Training details

4.2.1 DT classifier

The training process of DT models requires careful tuning of hyperparameters to avoid overfitting. Figure 4.3 demonstrates the impact of varying the maximum depth hyperparameter on the model's learning and generalization performance. For a maximum depth less than 4, the DT model achieved an accuracy of approximately 70% and an AUC of around 81%. This suggests that the model was underfitting, with insufficient complexity to capture the underlying patterns in the data. When the maximum depth increase, both the accuracy and AUC improve, stabilizing at around 77% accuracy and 88% AUC after a maximum depth of 14. This indicates that a maximum depth of 14 represents the optimal trade-off between model complexity and generalization, enabling the DT to learn and generalize as effectively as possible.

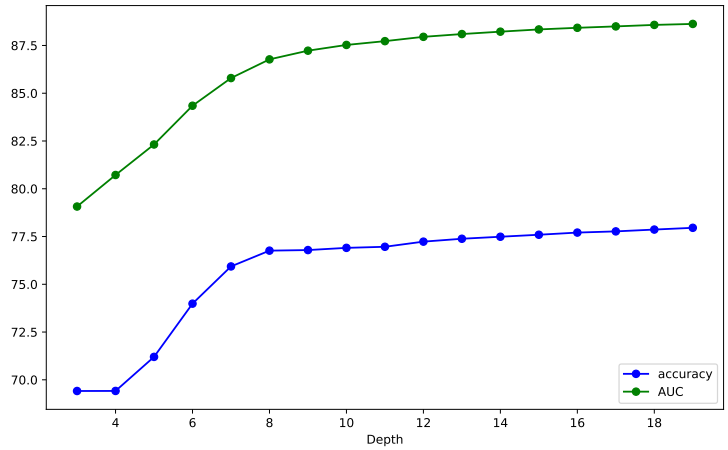


Figure 4.3: DT performance vs. depth

The feature p_T is the most important in the process of learning because it contribute at almost 58% to this process. The features d_z, d_0 and d_z contribute weakly respectively at 23%, 9%, 9.7%. However, ϕ contribute very weakly at 0.03% to the process of learning because it doesn't discriminate really signal from background events.

The learning process of the decision tree appears to be successful. As more data is used to train the classifier, the training score decreases while the validation score increases. With over 3,000,000 data points, both curves stabilize and converge (see Appendix A.1.2).

4.2.2 MLP classifier

The learning process of a MLP is not easy and require further analysis to find the hyper-parameters of the model that fit well the data.

- The learning rate is set to 1e-4 ;
- The batch size set to 50,000;
- Number of epochs set to 100 because after this epoch, both training and validation loss do not change anymore;

- Optimization algorithm: the algorithm used to update the weights of the network is Adam optimizer;
- Initialization method: for each linear layer, the weights are initialized with values drawn from a uniform distribution with a mean of 0 and a standard deviation calculated based on the number of input and output units.

Meanwhile, Binary Cross-Entropy (BCE) loss was employed during the training process. As shown in figure 4.4, both the training and validation loss curves exhibit a downward trend. The model achieved a training and validation loss of 0.45 (figure 4.4).

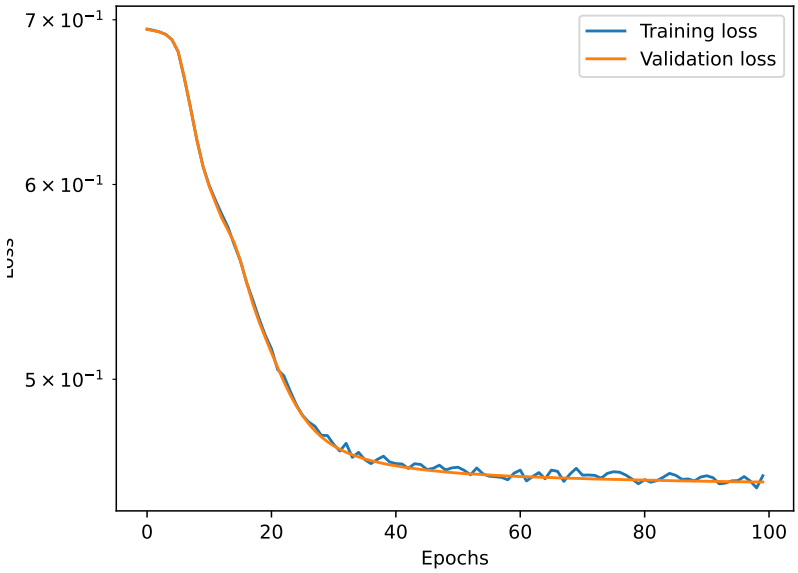


Figure 4.4: MLP loss

4.2.3 Auto-encoder model

The AE moel is trained on 5 million background events and validated on 400k background events. The configurations of the training are as following:

- Latent dimension: 4;
- learning rate: 1e-4;
- number of epochs: 100;
- batch size: 50,000;
- optimizer: Adam.

As demonstrated in the figure 4.5, the training and validation loss values both exhibit a decreasing trend, ultimately converging to values of 9.44e-6 and 1.14e-5, respectively. This indicates that the model is successfully learning the underlying patterns within the data during the training process.

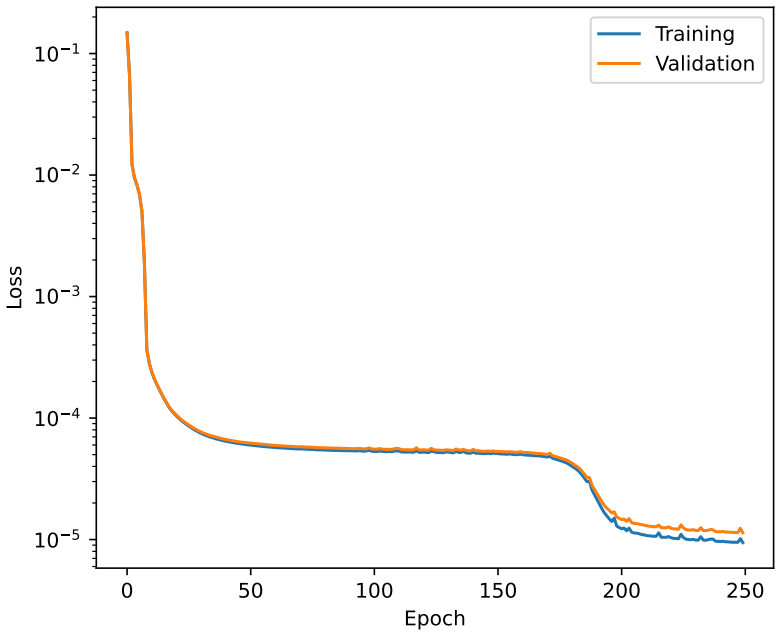


Figure 4.5: AE loss

The analysis reveals that the model has learned the distribution of the background events quite well. This can be ascertained through a comparative examination of the original data distributions and the reconstructed distributions generated by the model. The comparison between the original data distributions and the reconstructed distributions, as presented in Figure 4.6 and the supplementary appendix (Appendix A.3.3), demonstrates that the features p_T, η, ϕ and d_z have been quite well learned by the AE model. However, the model appears to have encountered difficulties in accurately learning the distributions of d_0.

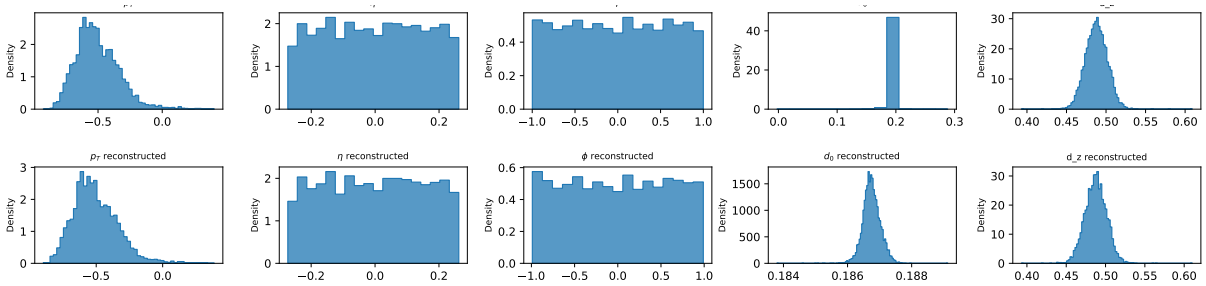


Figure 4.6: Original and learnt distributions by the AE model

The AE model constructs an anomaly score, as defined by equation 3.2.3, for the validation events, which include both background and signal events. The distributions of these anomaly scores exhibit a discriminative pattern, with a discernible threshold value at a score of 13 (figure 4.7). Events with an anomaly score below this value are considered to be outliers, or anomalies, within the dataset. Conversely, events with an anomaly score above this threshold are classified as normal, or background, events. This distinction provides a basis for differentiating between the signal and

background event populations based on the anomaly scores generated by the AE model.

The delineation of the anomaly score distribution, with a well-defined threshold, suggests that the AE model has successfully learned to discriminate between the background and signal event characteristics. This capability enables the effective identification of anomalous or outlier events within the validation dataset, which is a crucial step in the anomaly detection process.

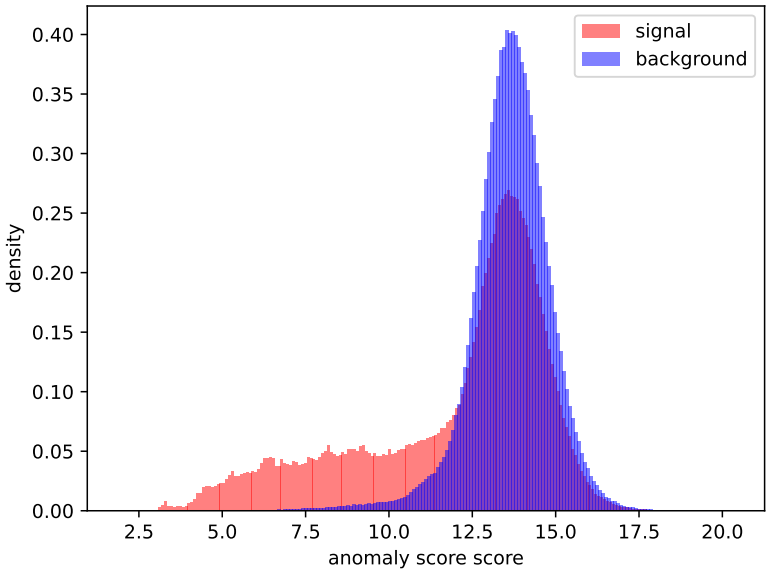


Figure 4.7: Distribution of the AE scores

4.2.4 VAE model

The architecture of the proposed VAE model is analogous to that of the previously described AE model, as depicted in figure 3.4. This structural similarity enables the reuse of certain hyper-parameters from the AE model, including the latent dimension, learning rate, number of epochs, batch size, and the optimizer.

In addition to the shared hyper-parameters, the VAE model introduces the notion of the β parameter, which represents the weighting of the KLD term in the total loss function. A range of β values is explored, as defined by the following set:

$$\mathcal{B} = \{0.001, 0.01, 0.05, 0.1, 0.5, 1\} \tag{4.2.1}$$

For each value of β, a corresponding VAE model is trained, and the learning curves are presented in the supplementary appendix A.4.1. The analysis reveals that the lowest reconstruction loss is achieved when the β parameter is set to 0.001. This suggests that, for the given dataset and task, emphasizing the reconstruction objective (i.e., a smaller β value) leads to superior performance in terms of minimizing the overall reconstruction error. In addition, this value of β allows the model to reconstruct the events better than the rest of the tested values(see appendix A.4.3 & A.4.4).

For the selected optimal value of the β parameter, specifically $\beta = 0.001$, the reconstruction loss exhibits a decreasing trend, ultimately converging to a value of 2e-3 after 30 epochs of training. This convergence behavior is also observed in the overall total loss of the VAE model (figure 4.8).

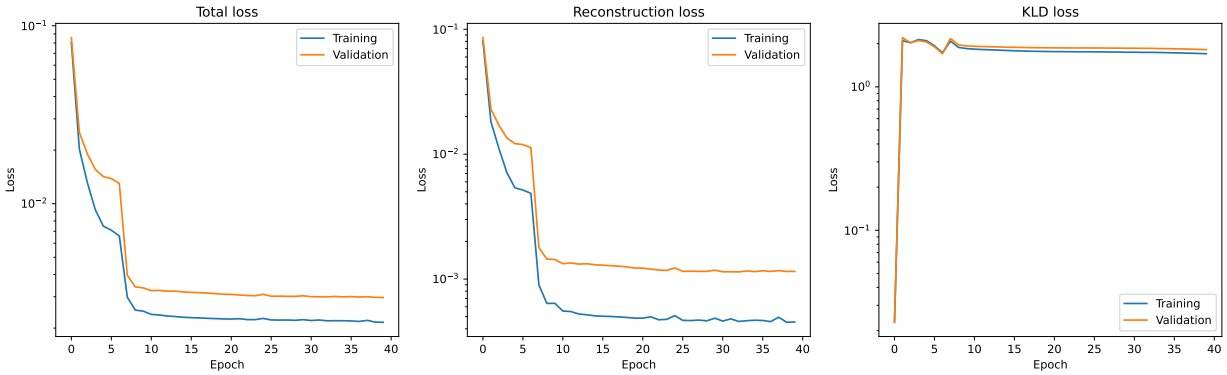


Figure 4.8: VAE loss

In contrast to the reconstruction and total loss profiles, the KLD loss term demonstrates an increasing tendency. During the initial 15 epochs, the KLD loss is observed to increase quickly. This is then followed by stabilization around a value of 1.63 after 80 epochs of training.

Similar to the AE model, the VAE generates an anomaly score, as defined by equation 3.2.3, for the validation dataset containing both signal and background events. The anomaly score distributions exhibit a discriminative pattern, revealing a threshold around a score of 7.0. This threshold serves as a decision boundary, with events below 7.0 considered anomalies and those above 6.0 classified as normal background events. The distinct separation of the distributions suggests the VAE model has effectively learned to differentiate between signal and background characteristics, enabling reliable anomaly detection within the validation set.

Let us note that VAEs do not directly optimize the likelihood of the data. Instead, they introduce a latent variable and learn an approximate posterior distribution over this latent variable. So, to estimate the true likelihood of the data under the VAE model, the ideal would be to integrate over the entire latent space. However, this integration is often intractable, as it involves high-dimensional marginalization. To address this, the likelihood is estimated using Monte Carlo sampling.

The likelihoods projected in 2-dimensional space allow us to visualize how well the VAE has learned the real distribution of the background events. As shown in Appendix A.4.6, the real events seem to be globally close to the simulated events. However, the maps obtained from combinations of the dimensions associated with p_T, ϕ, and η show some difficulties for the VAE to learn the distributions. This is because the real data are more widely dispersed than the likelihood projected in these 2D spaces. This visualization provides valuable insights into the performance of the VAE model. While the model appears to capture the overall distribution of the real events, the discrepancies observed in the p_T, ϕ, and η dimensions suggest that the VAE may struggle to fully learn the complex structure of the real data distribution.

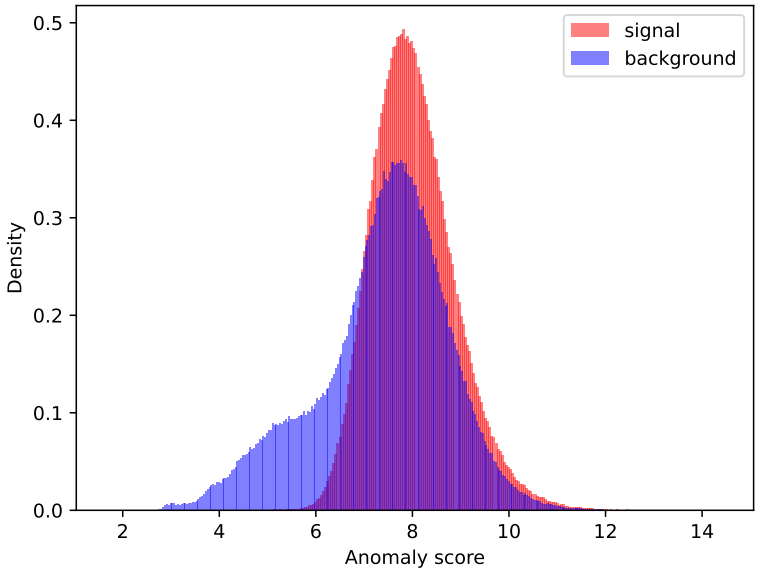


Figure 4.9: Distribution of the VAE scores

4.3 Performances analysis

After the training phase, the next step is to evaluate whether the proposed models generalize well to unseen data, typically performed using various performance indicators such as accuracy, False Positive Rate (FPR), False Negative Rate (FNR), F1-score, and area under the AUC curve, which provide a comprehensive evaluation of the model's performance on the validation or test dataset. Additionally, visualizations such as the confusion matrix and the Receiver Operating Characteristic (ROC) curve can offer valuable insights into the model's behavior, with the confusion matrix providing a breakdown of the model's predictions and the ROC curve plotting the true positive rate against the false positive rate to assess the model's trade-off between sensitivity and specificity. All of which are essential for understanding the strengths, weaknesses, and overall generalization capabilities of the trained models.

The table 4.2 reports the performance metrics of trained models evaluated on a dataset of 1,250,000 test events. The DT classifier achieved an accuracy of 77.47%. Regarding the anomalous (signal) events, the DT model correctly detected 75.52% of them, while failing to detect 19.59% (FNR). However, the DT model also incorrectly identified 25.48% of the events as anomalous (FPR). Despite these nuances, the DT model's F1-score of 78.12% indicates a good balance between precision and recall. This performance is further supported by the model's AUC of 77%, as shown in Appendix B.1a.

The MLP classifier exhibited similar yet slightly lower performance compared to the DT model. The MLP had an accuracy of 72.40%, which is 6 percentage points lower than the DT. Additionally, the MLP had a higher false negative rate of 31.48%, resulting in a lower F1-score of 71.30% compared to the DT.

Regarding the unsupervised learning models, the VAE showed the lowest performance, with an accuracy of 61.5% and high error rates (FPR of 9.88% and FNR of 67.07%), leading to an F1-score

of 46.12%. The AE show an AUC of 65.78%, which represents 2 points of percentages more higher than the VAE one (figure 4.10).

Model	Accuracy	False positive rate	False negative rate	F1-score	Test events
DT*	77.47%	25.48%	19.59%	78.12%	1,250,000
MLP	72.40%	23.71%	31.48%	71.30%	1,250,000
AE#	60.73%	35.00%	43.53%	59.00%	1,250,000
VAE	61.50%	9.88%	67.07%	46.12%	1,250,000

*: best model; #: best unsupervised model

Table 4.2: Performance metrics of the models

Overall, the DT model seems to be the best anomaly detector. It was able to achieve the highest accuracy, F1-score and AUC compared to others models.

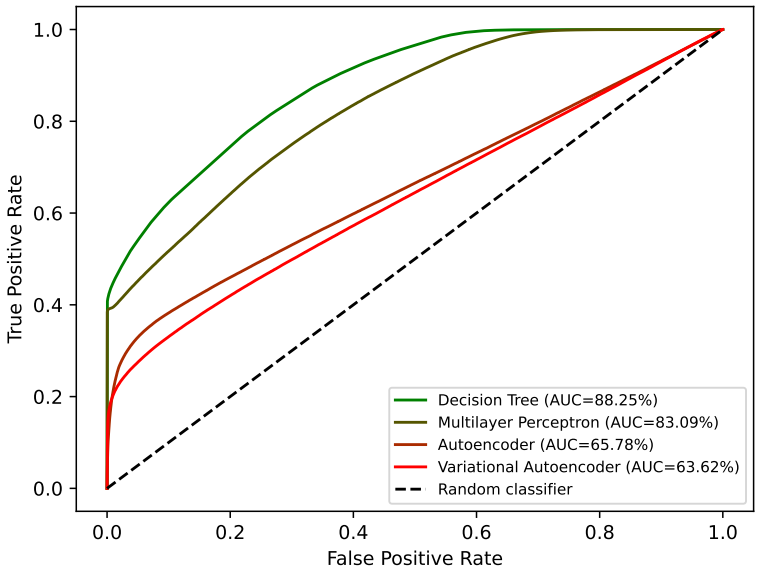


Figure 4.10: Comparison of ROC curves

On the other hand, a meticulous analysis of the distribution of the pseudo-rapidity (η) presented in the figure 4.1 reveal that the range of this feature depends if it is signal or background. This seems not realistic because the pseudo-rapidity (η) is a geometric quantity that does not depend on whether an event is signal or background. The pseudo-rapidity, as defined in the equation 3.1.2 in page 8, is a purely kinematic variable that characterizes the angular distribution of particles, and its distribution should not exhibit significant differences between signal and background events unless there are underlying physical differences in the production mechanisms or detector acceptance effects.

The difference in the range of the pseudo-rapidity (η) feature between signal and background events does not directly affect the learning process of the machine learning models, since the autoencoders models are trained on the background events. However, this discrepancy can have a significant

impact when evaluating the trained models, potentially leading to an increase in the true positive rate and, consequently, a distorted assessment of the model's performance.

To address the issue of model generalizability, the trained machine learning models can be assessed on a modified test dataset, where any events exhibiting pseudo-rapidity (η) values outside the range observed for the background class are excluded from the evaluation. The results are presented in the table 4.3 and figure 4.11. This targeted evaluation strategy helps to provide a more accurate and reliable assessment of the model's performance, focusing on its ability to accurately discriminate between signal and background based on the underlying patterns in the data, without being skewed by the biased feature distribution.

Model	Accuracy	False positive rate	False negative rate	F1-score	Test events
DT	72.43%	25.48%	30.84%	66.11%	1,021,805
MLP	66.25%	23.71%	49.55%	53.75%	1,021,805
AE	52.77%	35.00%	66.47%	35.57%	1,021,805
VAE	58.88%	9.84%	90.30%	15.50%	1,021,805

Table 4.3: Performance metrics of the models over the modified test set

In general, the performances of the models decreased considerably (around 10% lesser) especially for unsupervised learning models. But the best anomaly detector is still the DT classifier.

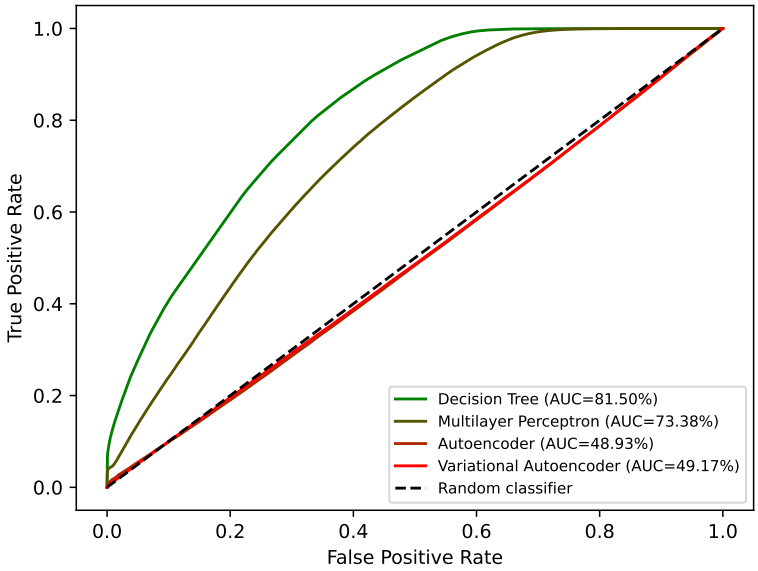


Figure 4.11: Comparison of ROC curves over the modified test set

5. Discussion and considerations for using ML in anomaly detection for HEP

In the search for new physics at the LHC, physicists try to find behaviors of the particles that don't satisfy the SM. This is known as anomaly detection, a crucial task in high energy physics research. The goal is to identify any deviations or discrepancies from the well-established predictions of the SM, which could be indicative of new fundamental particles, interactions, or phenomena beyond the current theoretical framework. The primary goal of this work was to develop a ML-based anomaly detection algorithm capable of identifying anomalous events in HEP data.

Every second, the LHC simulates over one billion particle collisions and stores the immense amount of data generated. Distinguishing which collisions are potentially interesting signals from the vast sea of expected SM background events has become an increasingly challenging task.

In recent years, the field of HEP has witnessed a remarkable shift towards the adoption of powerful ML models, which have demonstrated significant advantages over traditional statistical approaches in the context of anomaly detection. These advanced ML techniques have the ability to detect difficult patterns and deviations from the SM predictions that may be indicative of new physics phenomena.

5.1 Key findings and their implications

The results of this study demonstrate that the proposed ML-based anomaly detection approach was able to identify anomalous events with an accuracy of 77.5% and an AUC of 88%. The developed models still struggle to find and learn patterns in the data because the best model show an anomaly detection rate only of 80% and ignore 20% of signals. Tested over a more realistic data, the explored models show a very weak performances (see table 4.3) and figure 4.11).

The ability to detect anomalous events has important implications for HEP research. Anomalous events may indicate the presence of new physics phenomena or the malfunctioning of detector equipment. By quickly identifying these anomalies, physicists can focus their efforts on further investigating the most promising leads, potentially leading to new scientific discoveries or improved detector performance.

Additionally, the ML model was able to identify the most influential features contributing to anomaly detection (p_T and d_z), providing physicists with insights into which aspects of the event data are most indicative of anomalous behavior. This information can inform future detector design and data collection strategies.

The models proposed in this work occupy a remarkably small memory footprint. The DT model was observed to require 309.8kB of storage space and took 2.6 milliseconds to load. When performing a single prediction, the DT model exhibited an inference time of just 0.89 milliseconds. Conversely, the DT model demonstrated an even more compact memory usage of only 4.8kB, with a slightly longer loading time of 3.5 milliseconds. However, the DT model was able to generate predictions in 2.3 milliseconds per event. These findings indicate the exceptional efficiency and optimization potential of the proposed ML models. Their small memory footprints and rapid inference capabilities suggest they are well-suited for deployment in the demanding environment of LHC experiments, where high-throughput data processing is of paramount importance. The compact

nature and expeditious performance characteristics of these models highlight their suitability for integration into LHC research frameworks.

5.2 Limitations and future research directions

The developed machine learning-based anomaly detection models have demonstrated certain difficulties. Specifically, unsupervised anomaly detection remains the most challenging task. The autoencoders appear to be able to learn only simpler patterns in the data, and they have struggled when the difference between the signal and background is more complex.

For future research directions, we recommend the following:

- The latent space representation of the autoencoder models appears to exhibit some boundaries between signal events and background events. A supervised classifier can be trained on the latent space to detect anomalies with optimality.
- Exploration of more sophisticated architectures, such as GAN-AE or PGAE presented earlier, may yield improved performance in the anomaly detection task.

6. Conclusion

The current theory that explains the fundamental workings of the universe is the SM. This theory describes how 12 fundamental particles interact with each other through three fundamental forces: the strong, weak, and electromagnetic forces. However, the SM is known to be incompatible with the theory of general relativity, which describes the gravitational force. To address this issue, the LHC has been designed to simulate conditions similar to those just after the Big Bang. By colliding protons and observing the resulting particle behavior, physicists hope to discover new particles and incorporate the principles of general relativity into the SM.

Based on the concept of supersymmetry, researchers expect to detect anomalous events from the track parameters of the collisions. This has led to the development of various machine learning-based anomaly detection models, with the goal of accurately identifying signals beyond the SM.

The literature presents a wide range of supervised and unsupervised learning methods for anomaly detection, including DTs , NNs , and generative models such as autoencoders, GANs , and normalizing flows. This study explored two supervised models (DT and MLP) and two unsupervised models (AE and VAE) to compare their performance in detecting these anomalous events.

The results show that the DT model exhibits the best overall performance, with high accuracy and AUC scores. Among the unsupervised models, the AE (Autoencoder) model slightly outperforms the VAE model.

These findings contribute to the ongoing efforts to enhance our understanding of the fundamental nature of the universe by incorporating the principles of general relativity into the SM. The development and comparison of machine learning-based anomaly detection models provide valuable insights for future research in this field.

The limitations of the proposed anomaly detection models suggest the need to explore more sophisticated architectures. Future research could focus on generative models such as GAN-AE and PGAE, which may provide enhanced capabilities compared to the simpler models examined in this study. By expanding the investigation to include these more complex generative frameworks, researchers may uncover techniques that can more effectively identify the elusive signals beyond the SM, advancing our fundamental understanding of the universe.

Acknowledgements

I want to acknowledge AIMS and its funder Google DeepMind for supporting this work, as well as my supervisor, Dr. Daniel Murnane Thomas from University of Copenhagen.

References

Alwall, J., Frederix, R., Frixione, S., Hirschi, V., Maltoni, F., Mattelaer, O., Shao, H.-S., Stelzer, T., Torrielli, P., and Zaro, M. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), July 2014. ISSN 1029-8479. doi: 10.1007/jhep07(2014)079. URL http://dx.doi.org/10.1007/JHEP07(2014)079.

Bank, D., Koenigstein, N., and Giryes, R. Autoencoders, 2021. URL https://doi.org/10.48550/arXiv.2003.05991.

Barboianu, C. Galileo and the birth of modern science. *The Collector*, 2022. Last visited on 16.06.2024.

Belis, V., Odagiu, P., and Aarrestad, T. K. Machine learning for anomaly detection in particle physics. *Reviews in Physics*, 12:100091, Dec. 2024. ISSN 2405-4283. doi: 10.1016/j.revip.2024.100091. URL http://dx.doi.org/10.1016/j.revip.2024.100091.

Coadou, Y. *Boosted Decision Trees*, chapter Chapter 2, pages 9–58. 2022. doi: 10.1142/9789811234033_0002. URL https://www.worldscientific.com/doi/abs/10.1142/9789811234033_0002.

Collins, J. H., Howe, K., and Nachman, B. Anomaly detection for resonant new physics with machine learning, 2018. URL https://doi.org/10.48550/arXiv.1805.02664.

Evans, L. and Bryant, P. Lhc machine. *Journal of Instrumentation*, 3(08):S08001, aug 2008. doi: 10.1088/1748-0221/3/08/S08001. URL https://dx.doi.org/10.1088/1748-0221/3/08/S08001.

Gaillard, M. Cern data centre passes the 200-petabyte milestone. *CERN Accelerating science*, July 2017. URL https://www.home.cern/news/news/computing/cern-data-centre-passes-200-petabyte-milestone.

Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., and Tao, D. A survey on self-supervised learning: Algorithms, applications, and future trends, 2023. URL https://doi.org/10.48550/arXiv.2301.05712.

Izza, Y., Ignatiev, A., and Marques-Silva, J. On explaining decision trees, 2020.

Kasieczka, G., Nachman, B., Shih, D., Amram, O., Andreassen, A., Benkendorfer, K., Bortolato, B., Brooijmans, G., Canelli, F., Collins, J. H., Dai, B., De Freitas, F. F., Dillon, B. M., Dinu, I.-M., Dong, Z., Donini, J., Duarte, J., Faroughy, D. A., Gonski, J., Harris, P., Kahn, A., Kamenik, J. F., Khosa, C. K., Komiske, P., Le Pottier, L., Martín-Ramiro, P., Matevc, A., Metodiev, E., Mikuni, V., Murphy, C. W., Ochoa, I., Park, S. E., Pierini, M., Rankin, D., Sanz, V., Sarda, N., Seljak, U., Smolkovic, A., Stein, G., Suarez, C. M., Szewc, M., Thaler, J., Tsan, S., Udrescu, S.-M., Vaslin, L., Vlimant, J.-R., Williams, D., and Yunus, M. The lhc olympics 2020 a community challenge for anomaly detection in high energy physics. *Reports on Progress in Physics*, 84(12):124201, Dec. 2021. ISSN 1361-6633. doi: 10.1088/1361-6633/ac36b9. URL http://dx.doi.org/10.1088/1361-6633/ac36b9.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

Kosarenko, Y. How to create decision trees for business rules analysis. *Why change consulting*, 11 2021. URL https://why-change.com/2021/11/13/how-to-create-decision-trees-for-business-rules-analysis/. Last visited on 16.06.2024.

Metodiev, E. M., Nachman, B., and Thaler, J. Classification without labels: learning from mixed samples in high energy physics. *Journal of High Energy Physics*, 2017(10), Oct. 2017. ISSN 1029-8479. doi: 10.1007/jhep10(2017)174. URL http://dx.doi.org/10.1007/JHEP10(2017)174.

Ouali, Y., Hudelot, C., and Tami, M. An overview of deep semi-supervised learning, 2020.

Salam, A. *Weak and electromagnetic interactions*, pages 244–254. 1994. doi: 10.1142/9789812795915_0034. URL https://www.worldscientific.com/doi/abs/10.1142/9789812795915_0034.

Serway, R. A. and Jewett, J. W. *Physics for scientists and engineers*. 6th edition, 2004. URL https://faculty.ksu.edu.sa/sites/default/files/physics_serway.pdf.

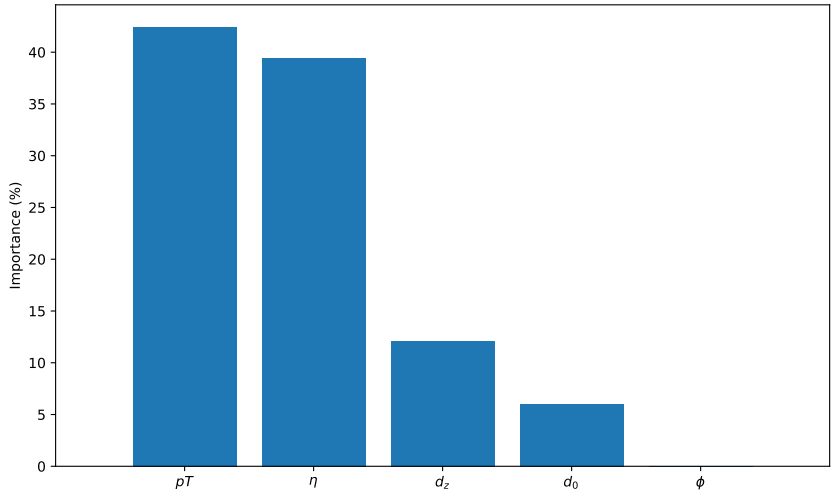
Tsan, S., Kansal, R., Aportela, A., Diaz, D., Duarte, J., Krishna, S., Mokhtar, F., Vlimant, J.-R., and Pierini, M. Particle graph autoencoders and differentiable, learned energy mover's distance, 2021. URL https://doi.org/10.48550/arXiv.2111.12849.

Vaslin, L., Barra, V., and Donini, J. Gan-ae: an anomaly detection algorithm for new physics search in lhc data. *The European Physical Journal C*, 83, Nov. 2023. ISSN 1434-6052. doi: 10.1140/epjc/s10052-023-12169-4. URL https://doi.org/10.1140/epjc/s10052-023-12169-4.

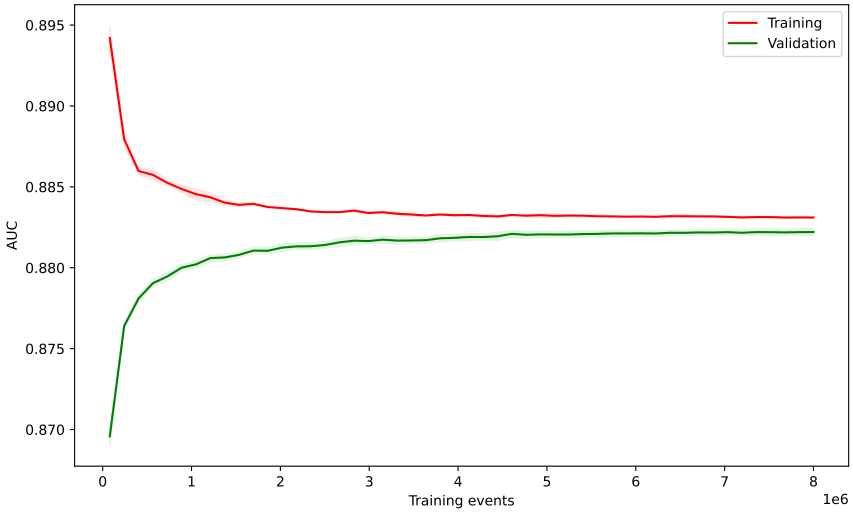
Appendix A. Training plots

A.1 Training plots of the DT

A.1.1 Feature importance plot

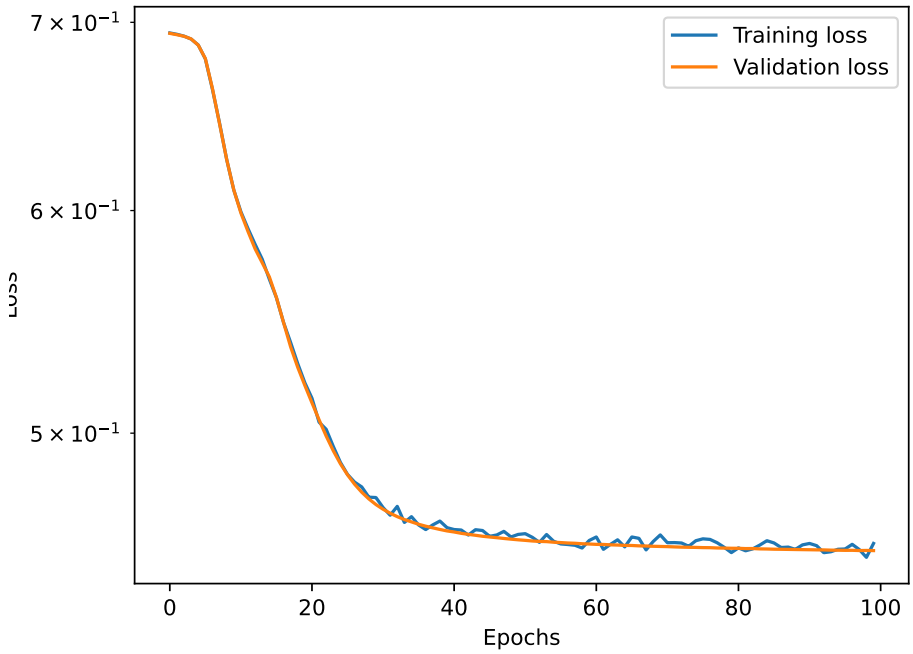


A.1.2 Learning curves



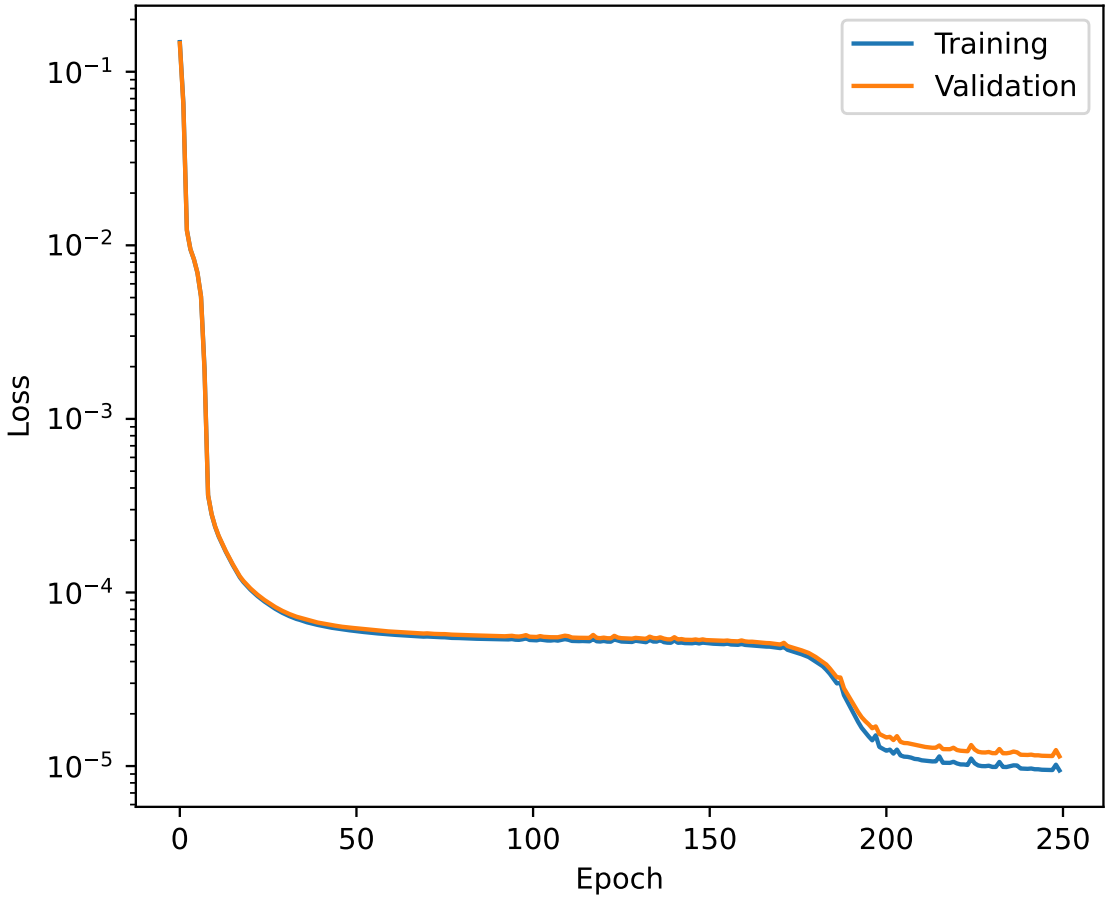
A.2 Training plots of the MLP

A.2.1 Training and validation loss

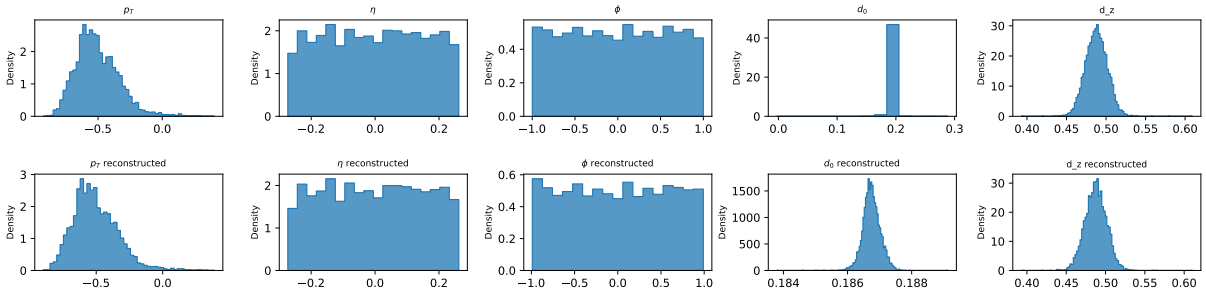


A.3 Training plots of the AE

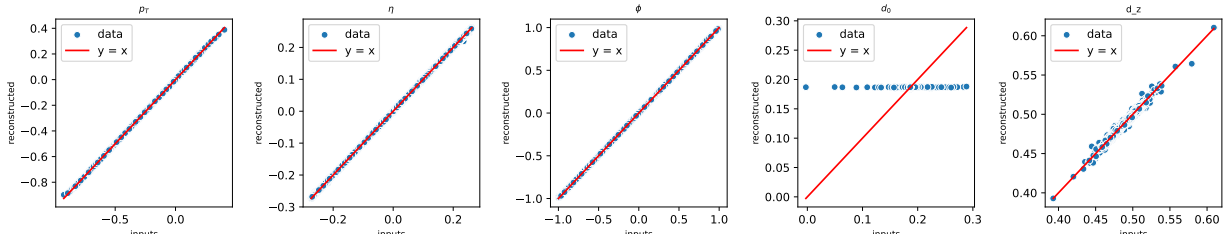
A.3.1 Training and validation loss of the AE



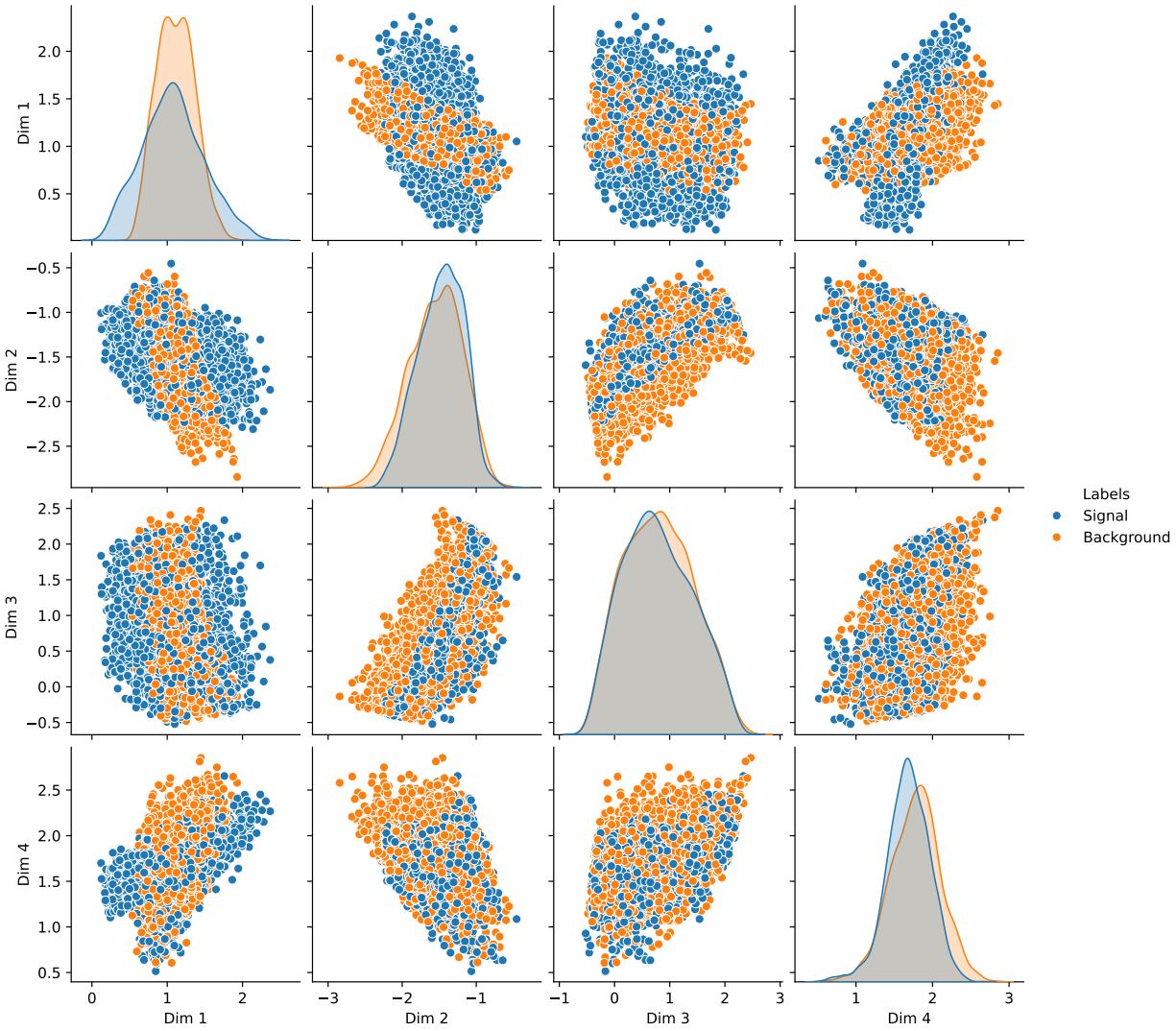
A.3.2 Distribution of reconstructed events by the AE



A.3.3 Comparison of inputs and reconstructed events by the AE

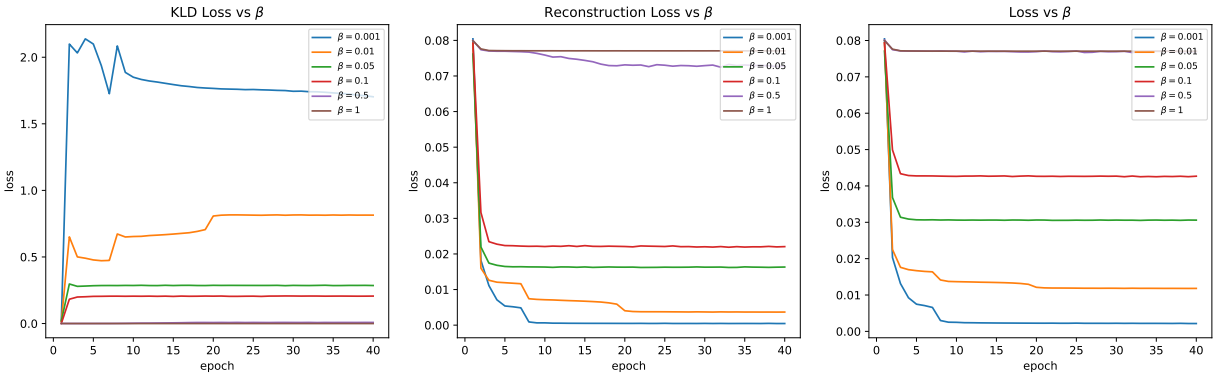


A.3.4 Latent space of the AE projected in 2D

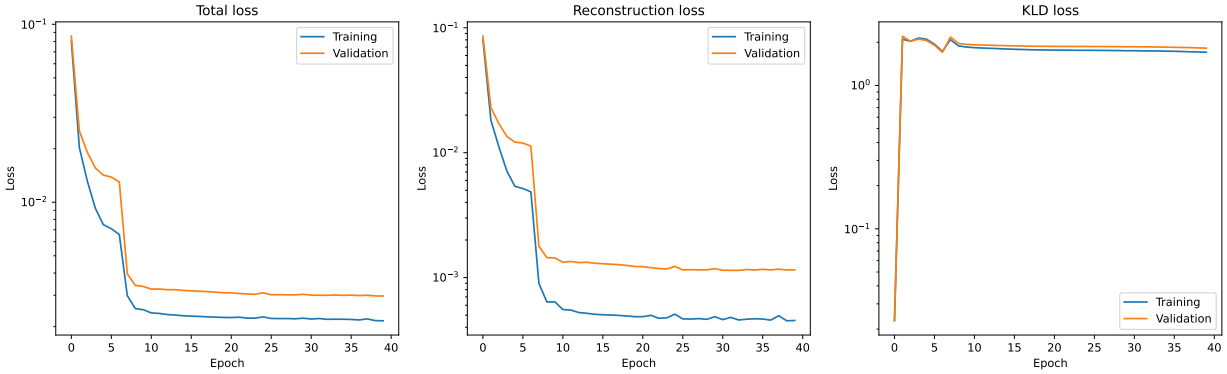


A.4 Training plots of the VAE

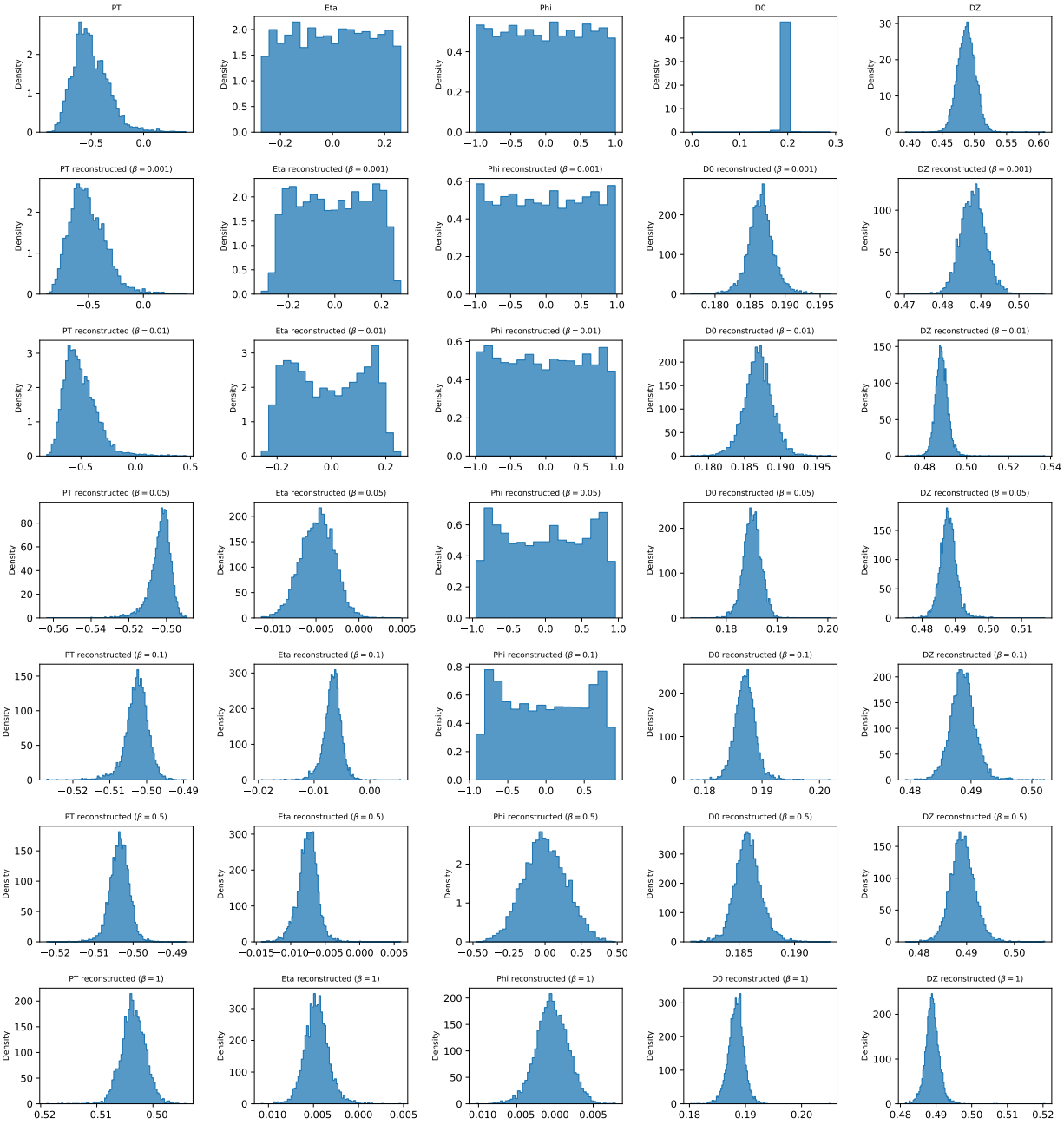
A.4.1 Losses vs β values



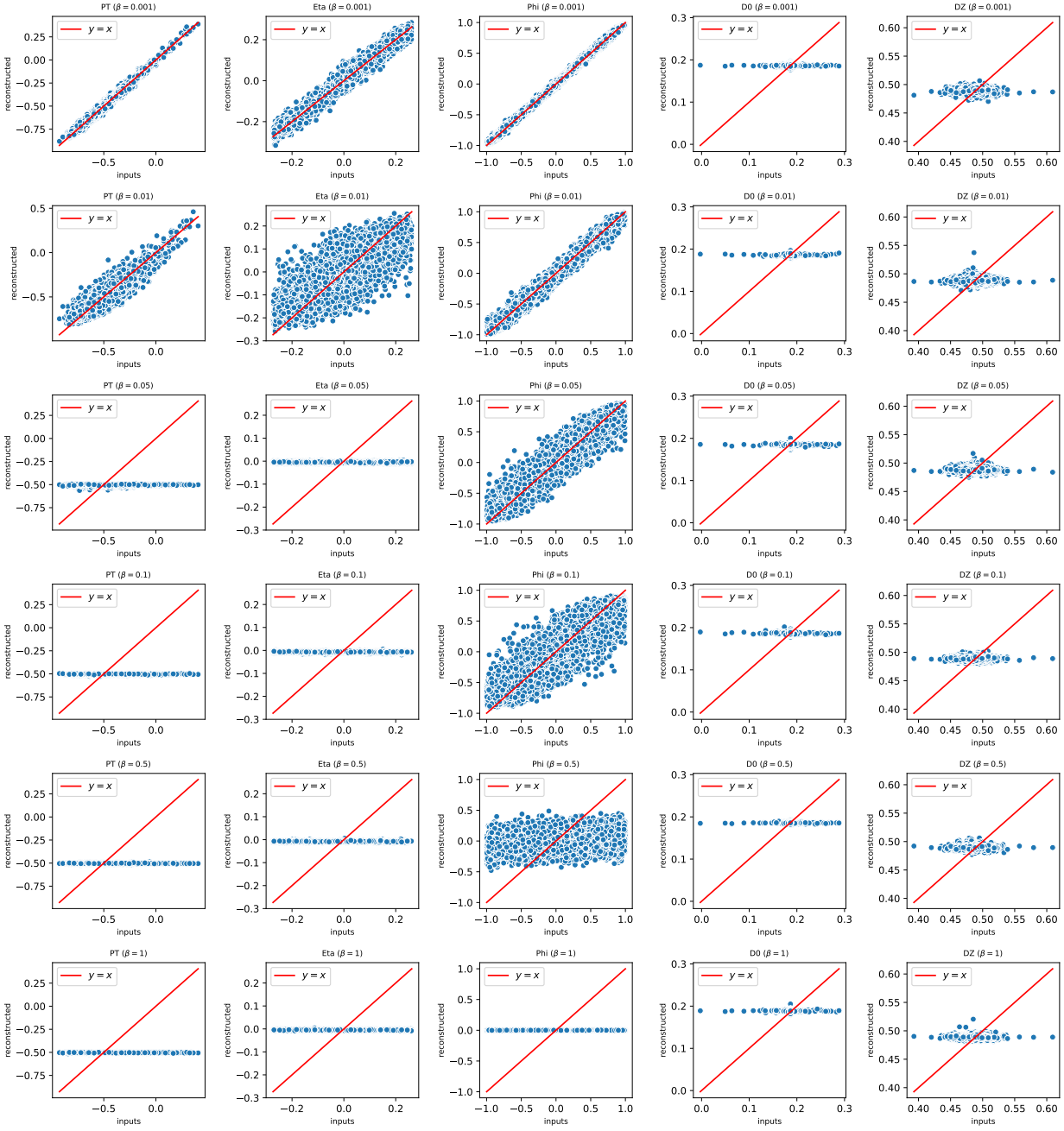
A.4.2 Training and validation loss of the VAE for $\beta = 0.001$



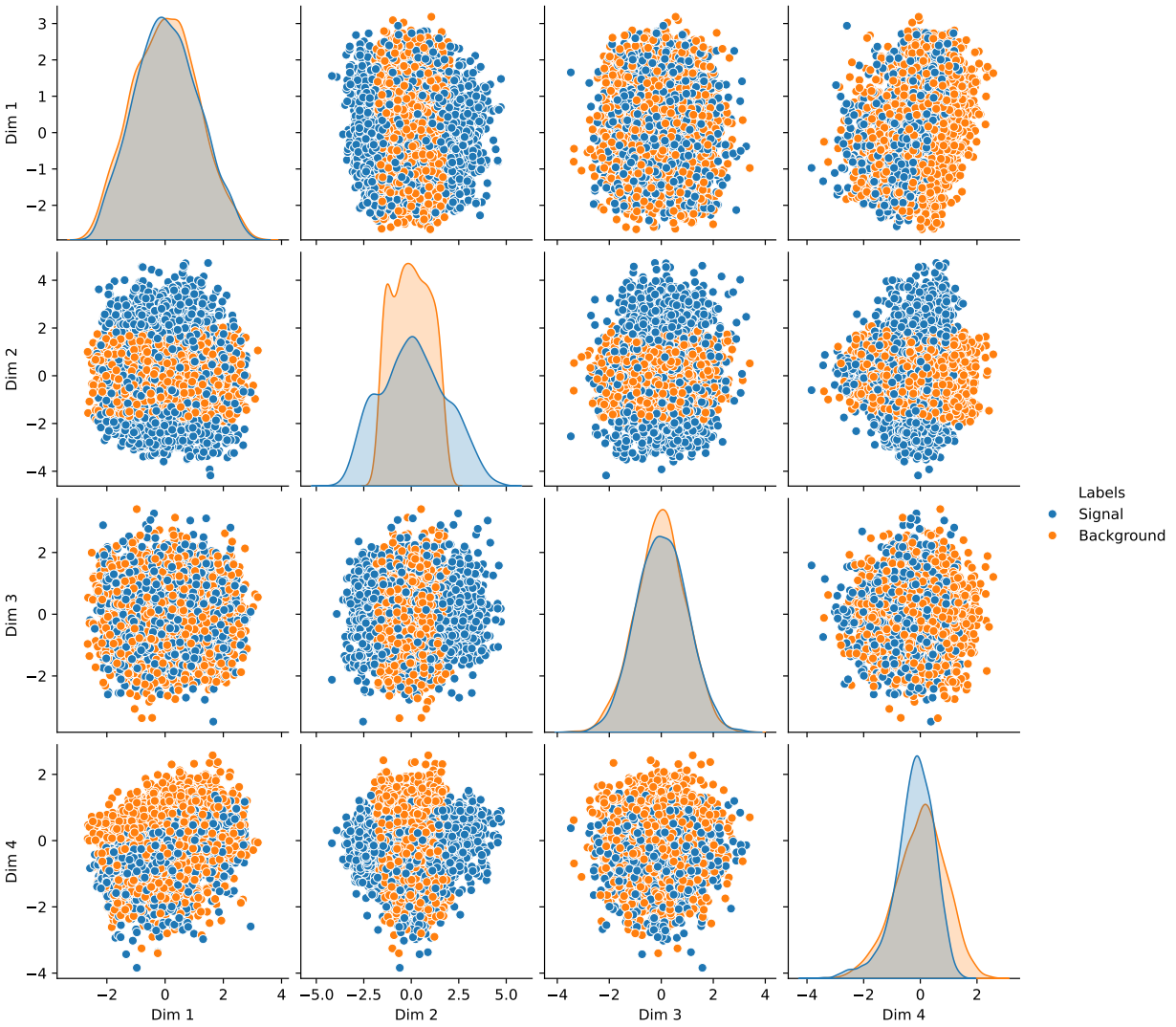
A.4.3 Distribution of reconstructed events by the VAE



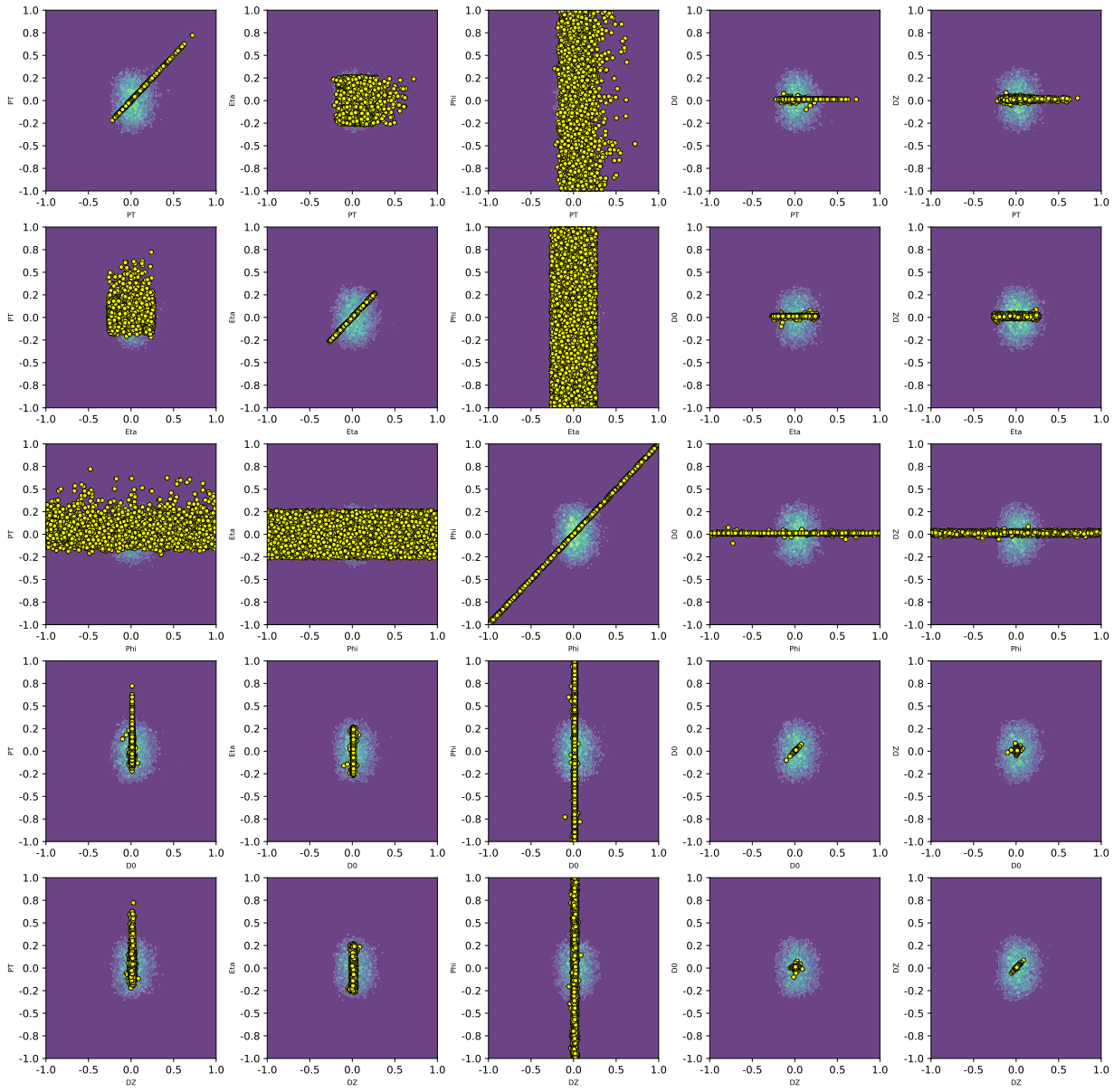
A.4.4 Comparison of inputs and reconstructed events by the VAE vs β values



A.4.5 Latent space of the VAE projected in 2D

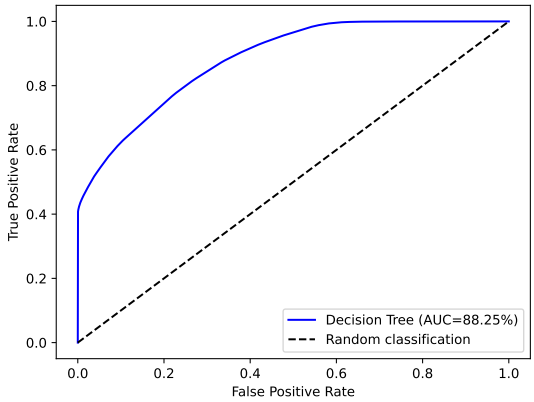


A.4.6 VAE likelihoods projected in 2D

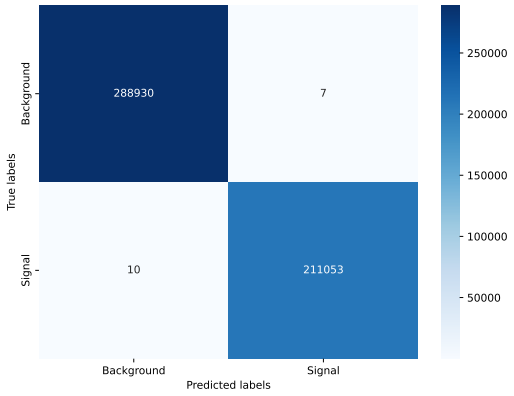


Appendix B. Performance plots

B.1 Performance plots of the DT classifier

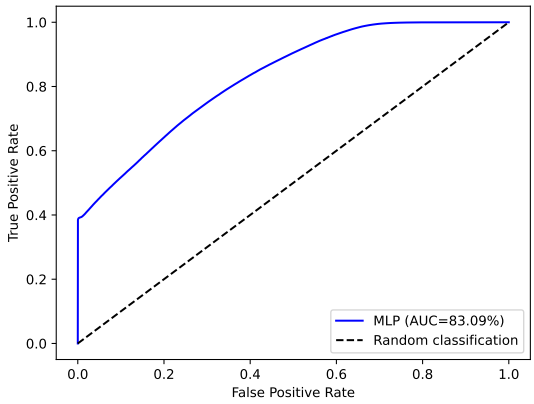


(a) ROC curve

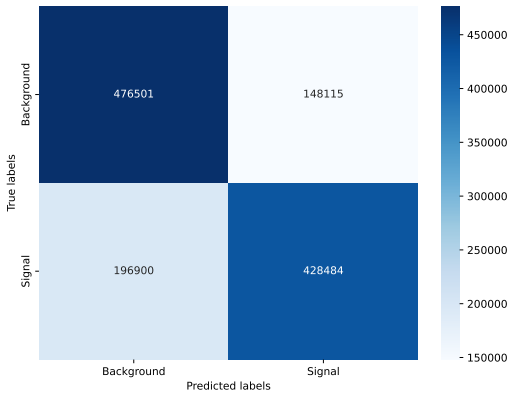


(b) Confusion matrix

B.2 Performances plots of the MLP classifier

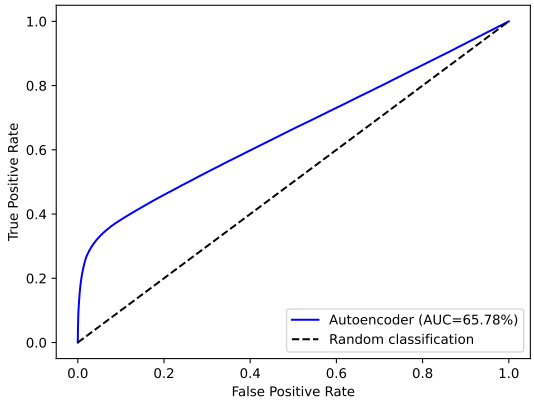


(a) ROC curve

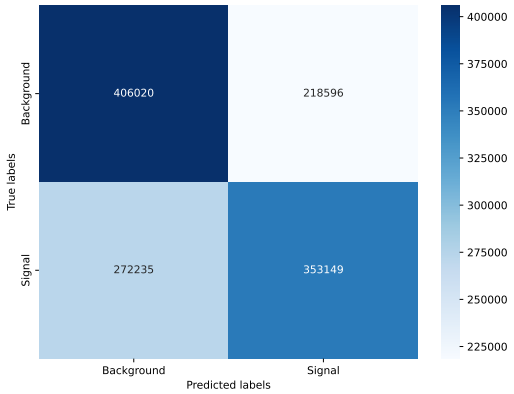


(b) Confusion matrix

B.3 Performances plots of the AE

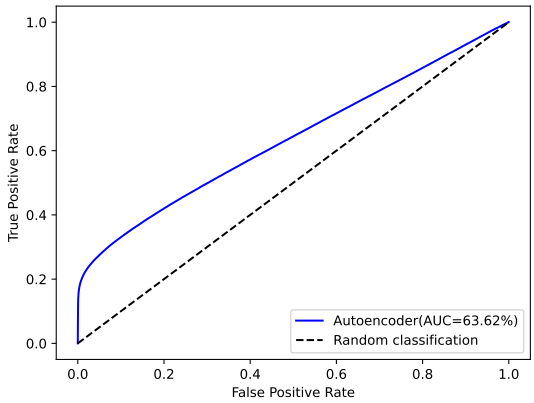


(a) ROC curve

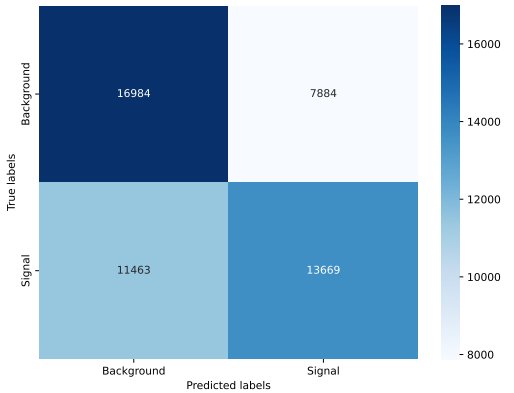


(b) Confusion matrix

B.4 Performances plots of the VAE



(a) ROC curve



(b) Confusion matrix

Figure