KERNEL DENSITY ESTIMATION-BASED SIMULATION OF MONTE-CARLO EVENTS AT LHC

KRUGER 2022

NIDHI TRIPATHI

SCHOOL OF PHYSICS

UNIVERSITY OF THE WITWATERSRAND

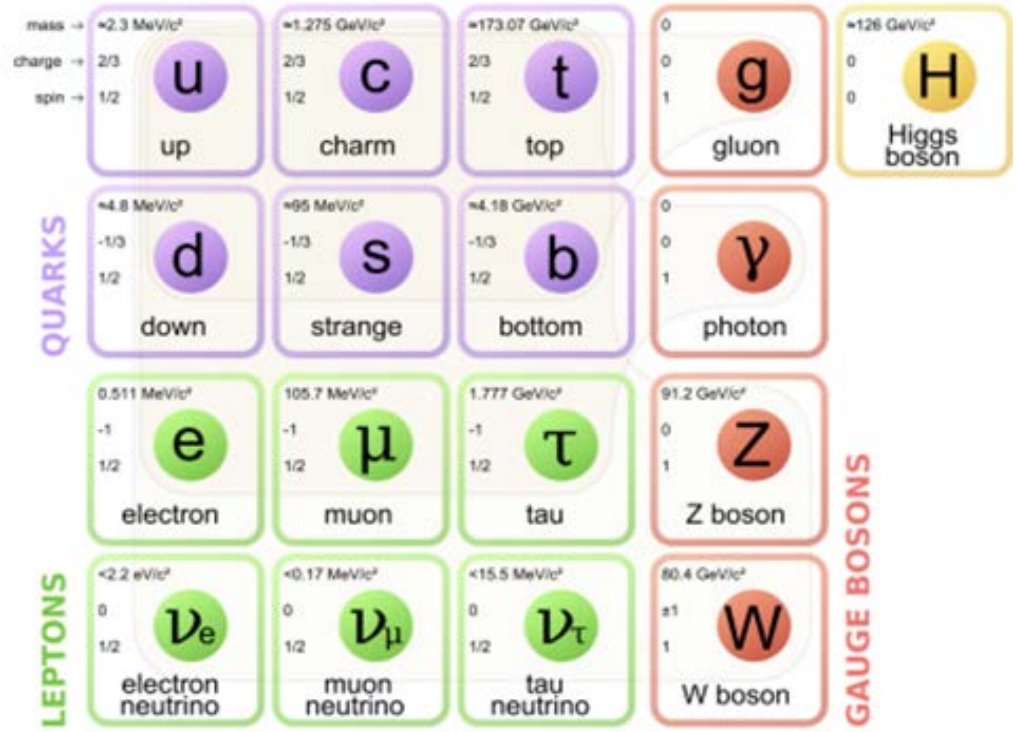
Outline

1. Introduction
2. Methodology
3. Results
4. Conclusion

Introduction

Objective

- After the discovery of the Higgs boson new opportunity opens for the search of direct evidence of physics beyond the Standard Model (SM).
- Searches for new bosons are completed by looking for Zgamma resonances in Zγ (pp → H → Zγ) fast simulation events.
- Semi-supervised machine learning can play a significant role to reduce model dependencies.
- To quantify the fake signal generated in the training of semi-supervised DNN classifier



Introduction

Machine Learning

- Machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, algorithm will produce an estimate about a pattern in the data and try imitate the way that humans learn, gradually improving its accuracy.

- Machine learning algorithms are generally used for different applications such as weather forecasting, stock trading, facial recognition, medical prediction, spam detection and commodity sales among others.

- Machine learning is a sub-fields of artificial intelligence.

Ref: https://www.ibm.com/za-en/cloud/learn/machine-learning#toc-machine-le-SzgJbkmk

Introduction

Machine Learning

Supervised machine learning

Supervised machine learning, is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.

Unsupervised machine learning

Unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention.

Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set.

Ref: https://www.ibm.com/za-en/cloud/learn/machine-learning#toc-machine-le-SzgJbkmk

Introduction

Why Machine Learning?

- Synthetic data generated using machine learning based on Kernel density estimation.
- Quantify fake signals in generated data trained on weak supervised DNN classifiers.
- A Deep Neural Network (DNN) model based on weak supervision trained on generated Zγ background data to perform binary classification.
- MC simulation at LHC is CPU intensive, machine learning reduce CPU pressure.
- To reduce model dependencies and for frequentist study.

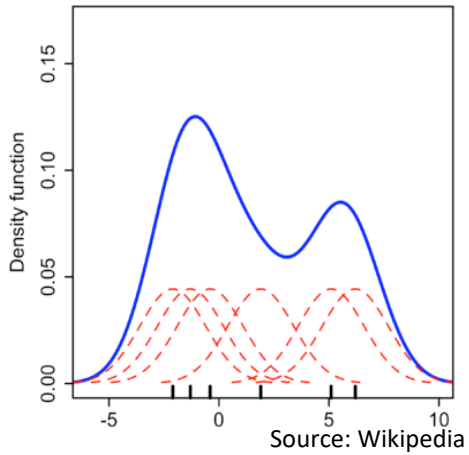
Methodology

Kernel Density Estimation

Kernel density estimation or KDE is a non-parametric way to estimate the probability density function of a random variable. In other words, the aim of KDE is to find probability density function (PDF) for a given dataset. With this generative model, new samples can be drawn.

For a sample of $(x_1, x_2, \ldots, x_n)$ the kernel density estimate, is given by:

$$p(x)=\frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{x-xj}{h}\right)$$



Source: Wikipedia

where K(a) is the kernel function and h is the smoothing parameter, also called the bandwidth.

Ref:
https://en.wikipedia.org/wiki/Kernel_density_estimation#:~:text=In%20statistics%2C%20kernel%20density%20estimation,on%20a%20finite%20data%20sample.

Methodology

Kernel Density Estimation

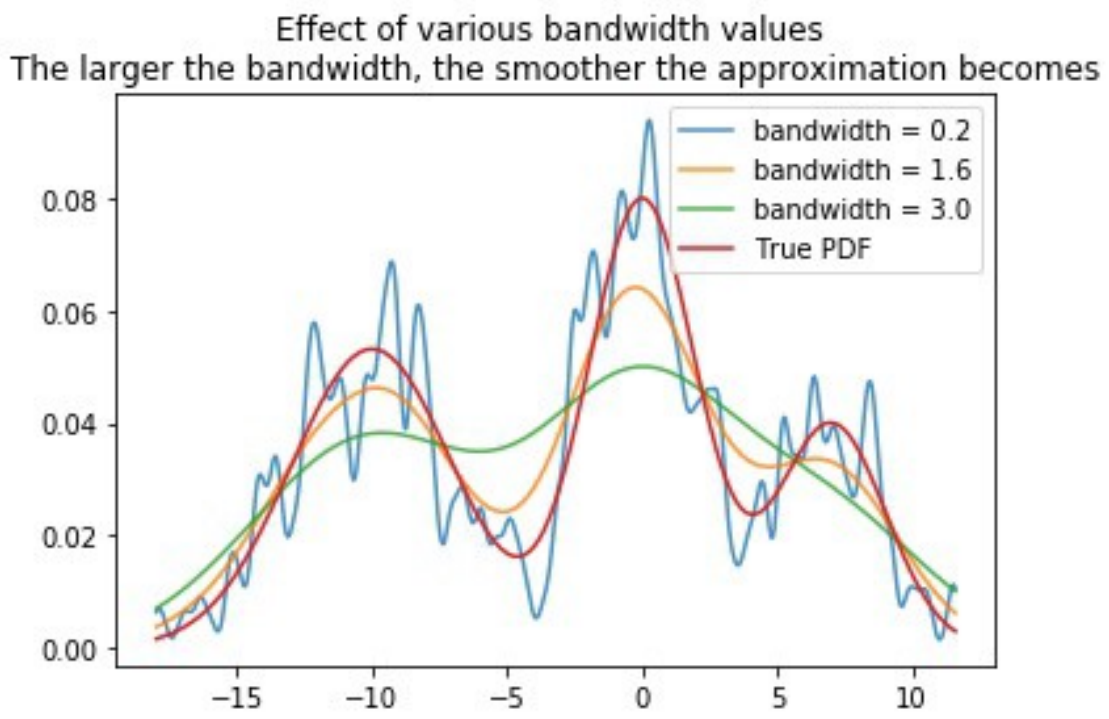
- Kernel Density Estimation (KDE) uses the Ball Tree or KD Tree algorithms for efficient queries.
- KDE are some of the most popular and useful density estimation techniques.
- The general idea of machine learning is to get a model to learn distribution of real data and be able to reproduce synthetic data with similar distribution.

Methodology

Tuning of the bandwidth parameter

- The scikit-learn library allows the tuning of the bandwidth parameter via cross-validation and returns the parameter value that maximizes the log-likelihood of data.

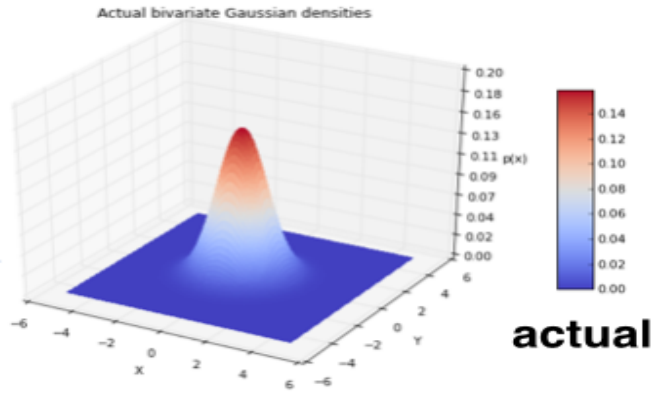
- The function we can use to achieve this is GridSearchCV(), which requires different values of the bandwidth parameter.



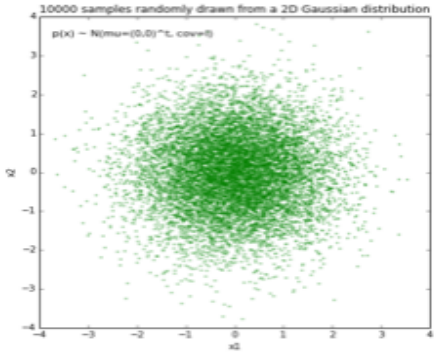
Source: Wikipedia

Methodology

Our goal:

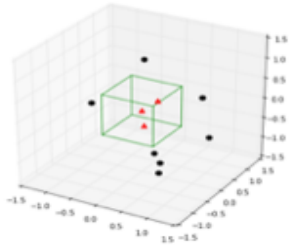


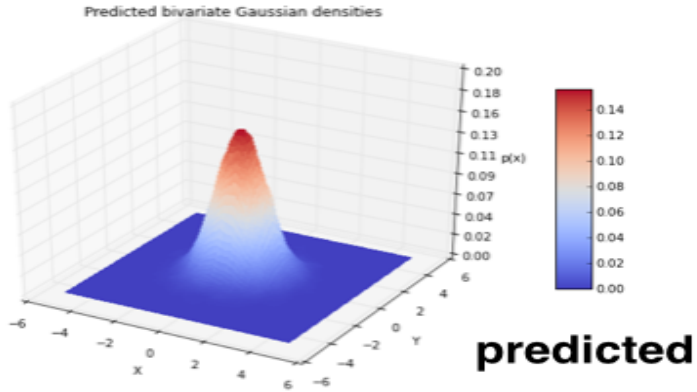
actual



Assuming we have samples drawn from a **unknown** distribution (here: bivariate Gaussian)

We want to **estimate** probability densities at certain points



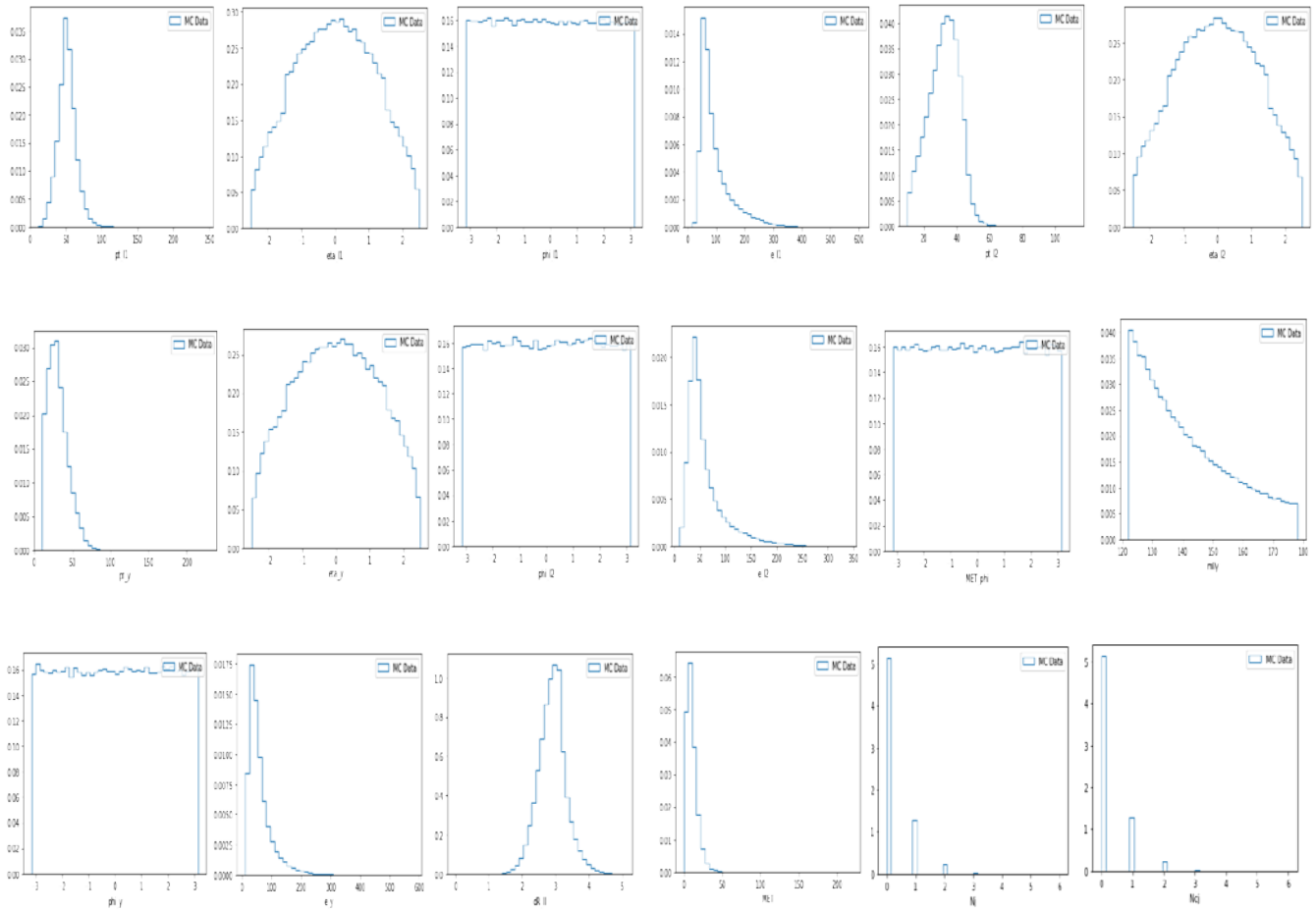


predicted

Methodology

Dataset

In the search for new bosons, the Zγ final state data is used as a pure background Monte Carlo(MC) sample. Using Scikit-learn and NumPy libraries the KDE generative model is constructed to take the pre-processed Zγ data and generate a sample dataset.



dataset:

['mlly', 'phi_zy', 'eta_zy', 'pt_zy', 'e_zy', 'mll', 'phi_ll', 'eta_ll', 'pt_ll', 'e_ll', 'dR_ll', 'MET', 'MET_phi', 'Nj', 'Ncj', 'dPhi_ll', 'dPhi_METZy', 'llpt_mlly']

Methodology

Parameters:

algorithm{‘kd_tree’, ‘ball_tree’, ‘auto’}: = auto

bandwidth: = 0.001

breadth_first’: = True

kernel: = gaussian

leaf_size: = 40

metric: = 'euclidean',

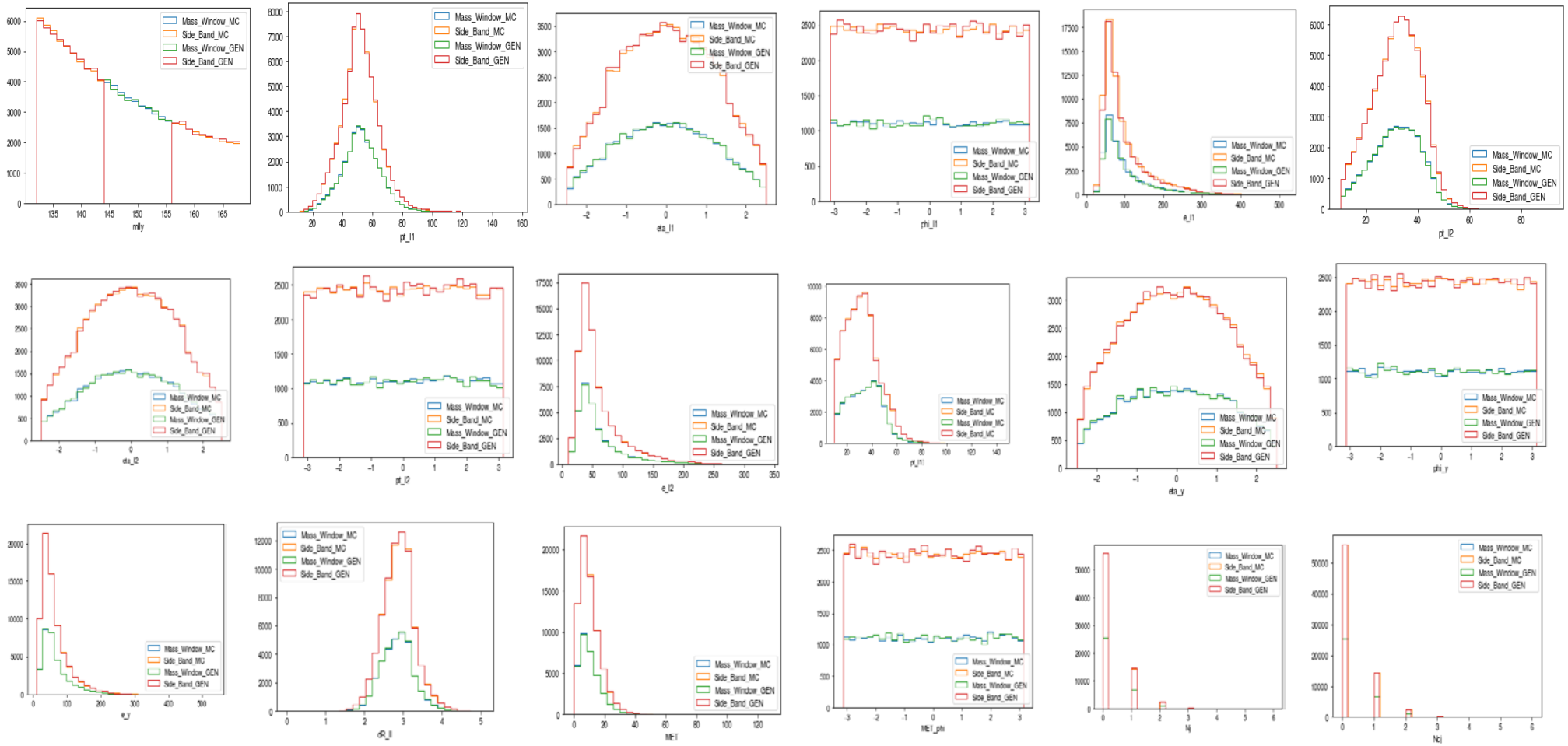
metric_params: = None,

rtol: = 0

Results

Comparison of for MC and Generated datasets

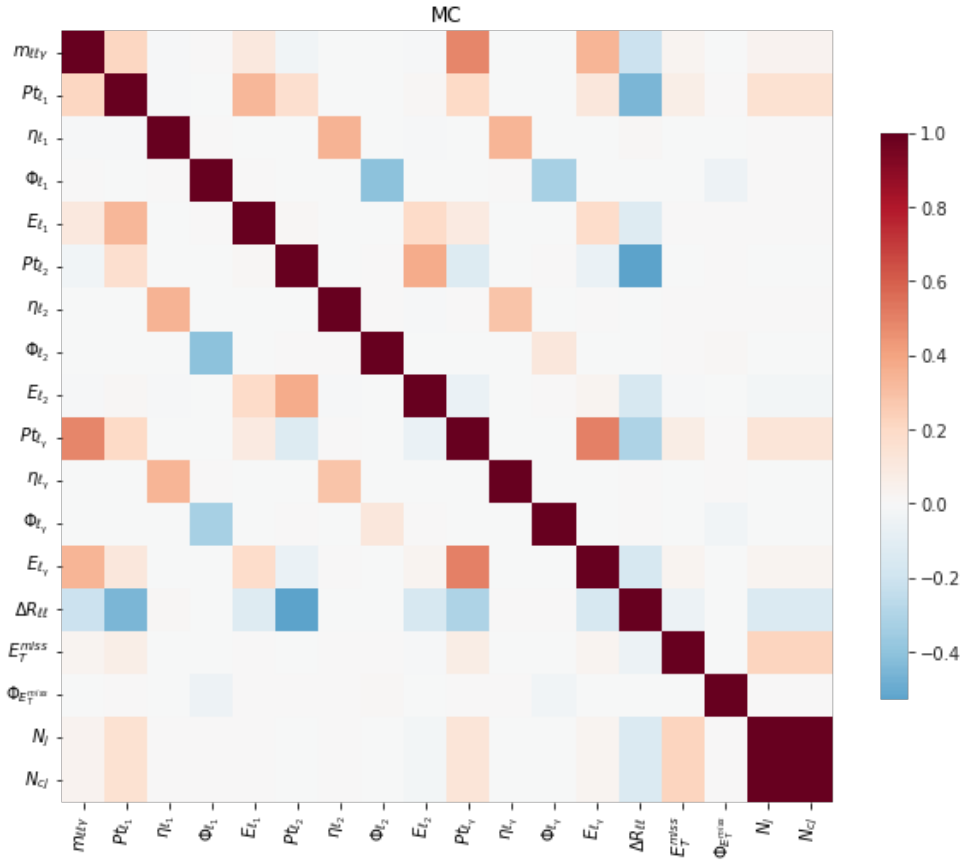
- Centre of Mass: 150GeV
- Side Band: (132 ≤ mℓℓ < 144) and (156 < mℓℓ ≤ 168)
- Mass Window: 144 ≤ mℓℓ ≤ 156

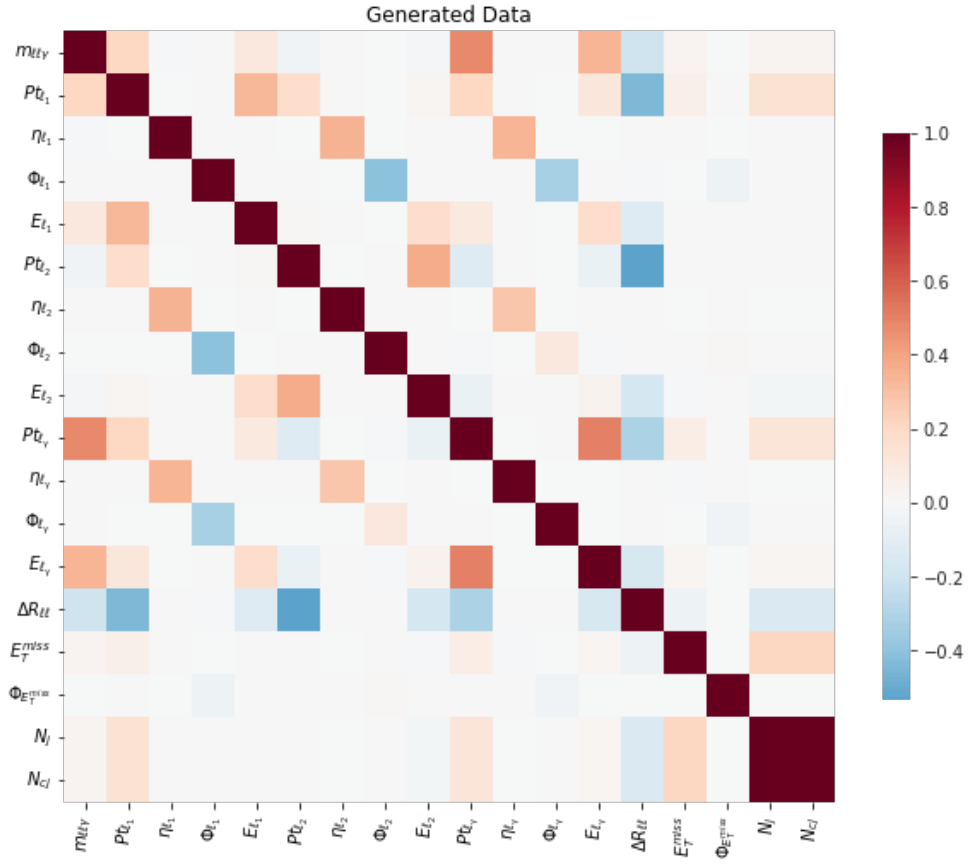


Results

Correlation Matrix

- Each cell in the figure shows the correlation between two variables.

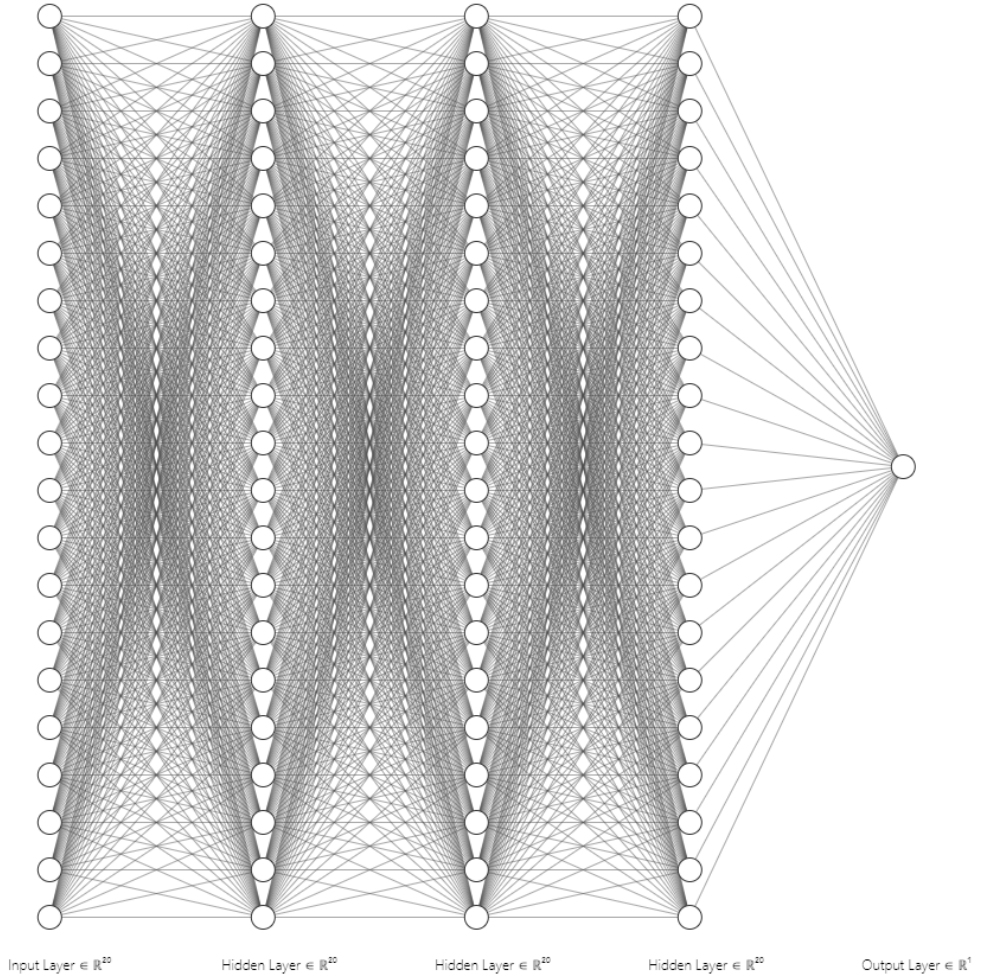




Results

DNN structure on weakly supervised learning

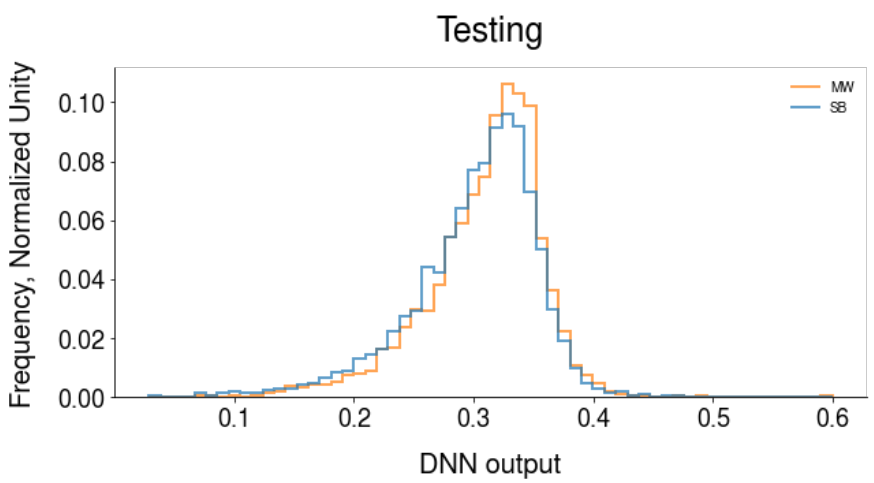
- DNN with four hidden layers of 20 nodes.
- Hidden Layers of the DNN used ReLu for an activation function and sigmoid for the output.
- DNN Model is trained on Zγ final state data as pure background.
- During the weak supervised learning study, the generated data set is divided into mass window and side band :
  - Sideband is labeled as 1
  - Mass window is labeled as 0

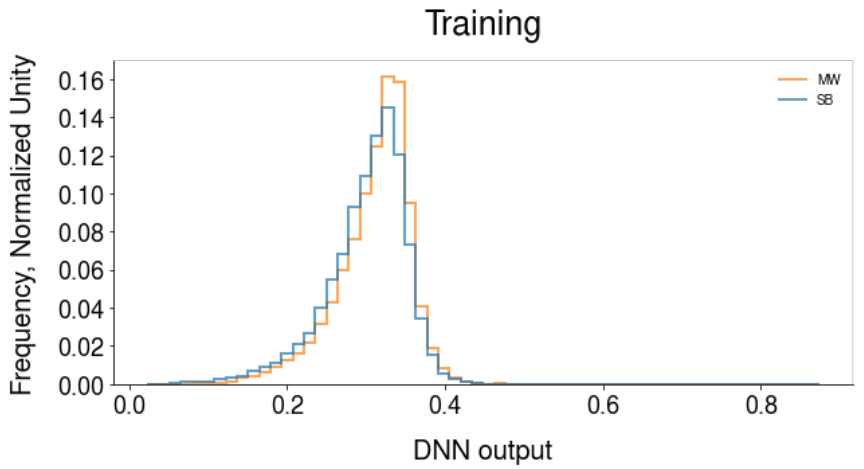


Results

DNN Response

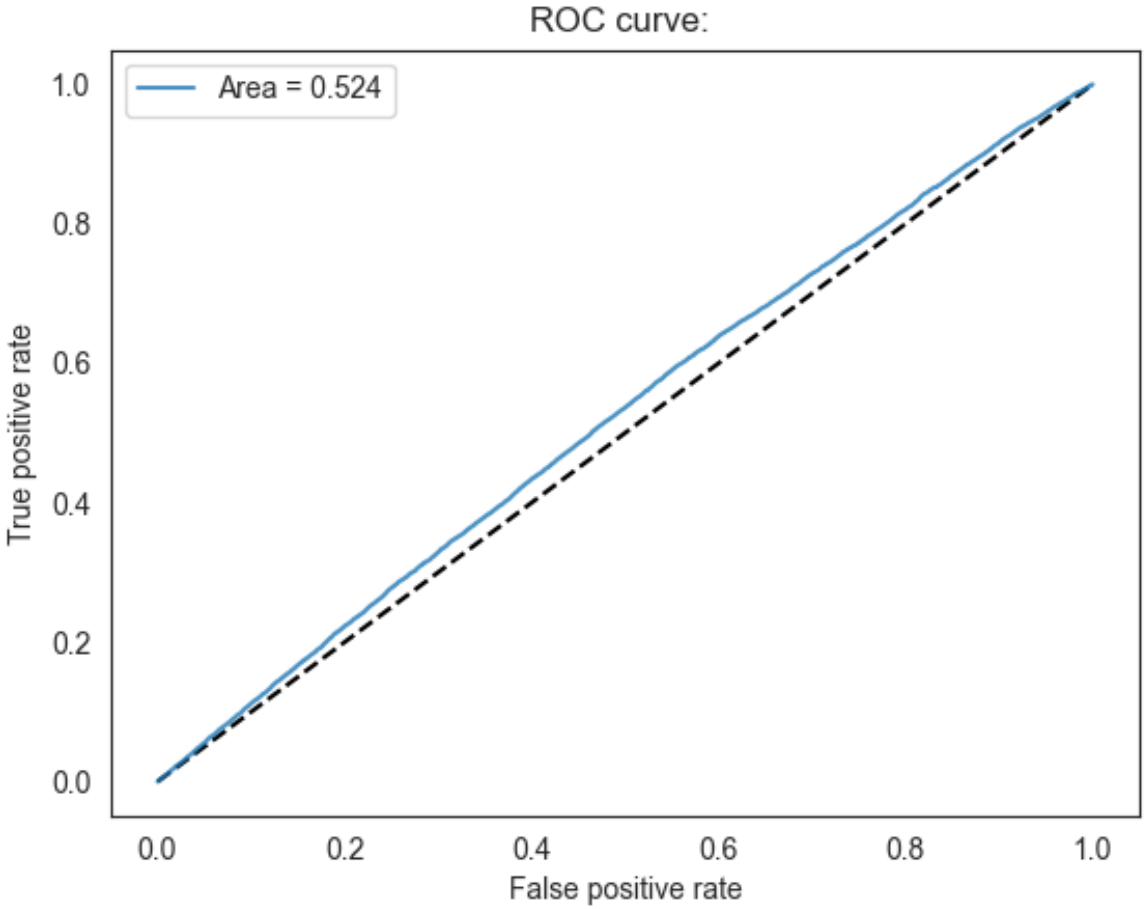
- DNN classifier based on weak supervision trained on generated dataset which is pure background.
- The DNN should not find any separation between samples as it has no signals in the samples.
- Quantify the fake signal generated.

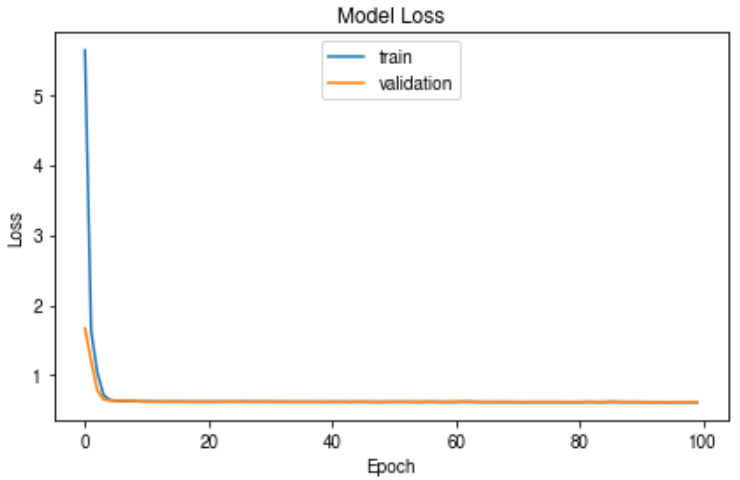


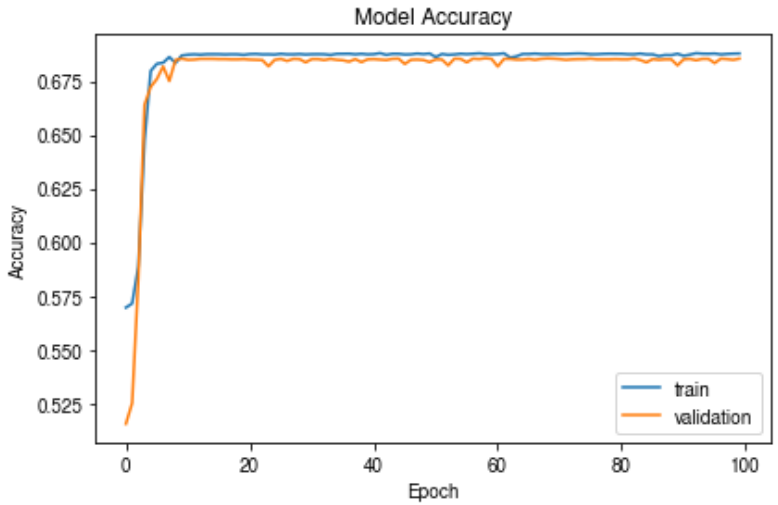


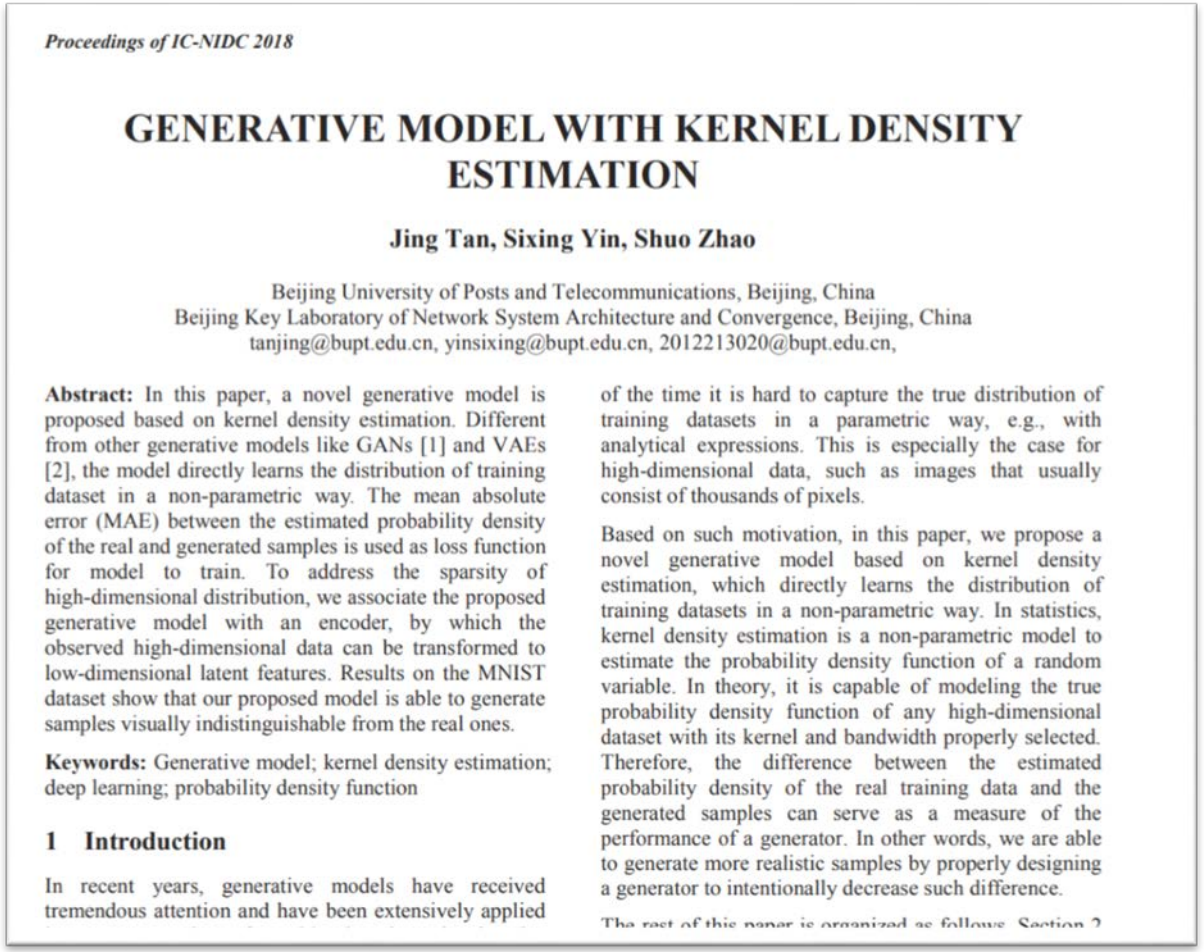
Results

DNN Response









Proceedings of IC-NIDC 2018

GENERATIVE MODEL WITH KERNEL DENSITY ESTIMATION

Jing Tan, Sixing Yin, Shuo Zhao

Beijing University of Posts and Telecommunications, Beijing, China
Beijing Key Laboratory of Network System Architecture and Convergence, Beijing, China
tanjing@bupt.edu.cn, yinsixing@bupt.edu.cn, 2012213020@bupt.edu.cn,

Abstract: In this paper, a novel generative model is proposed based on kernel density estimation. Different from other generative models like GANs [1] and VAEs [2], the model directly learns the distribution of training dataset in a non-parametric way. The mean absolute error (MAE) between the estimated probability density of the real and generated samples is used as loss function for model to train. To address the sparsity of high-dimensional distribution, we associate the proposed generative model with an encoder, by which the observed high-dimensional data can be transformed to low-dimensional latent features. Results on the MNIST dataset show that our proposed model is able to generate samples visually indistinguishable from the real ones.

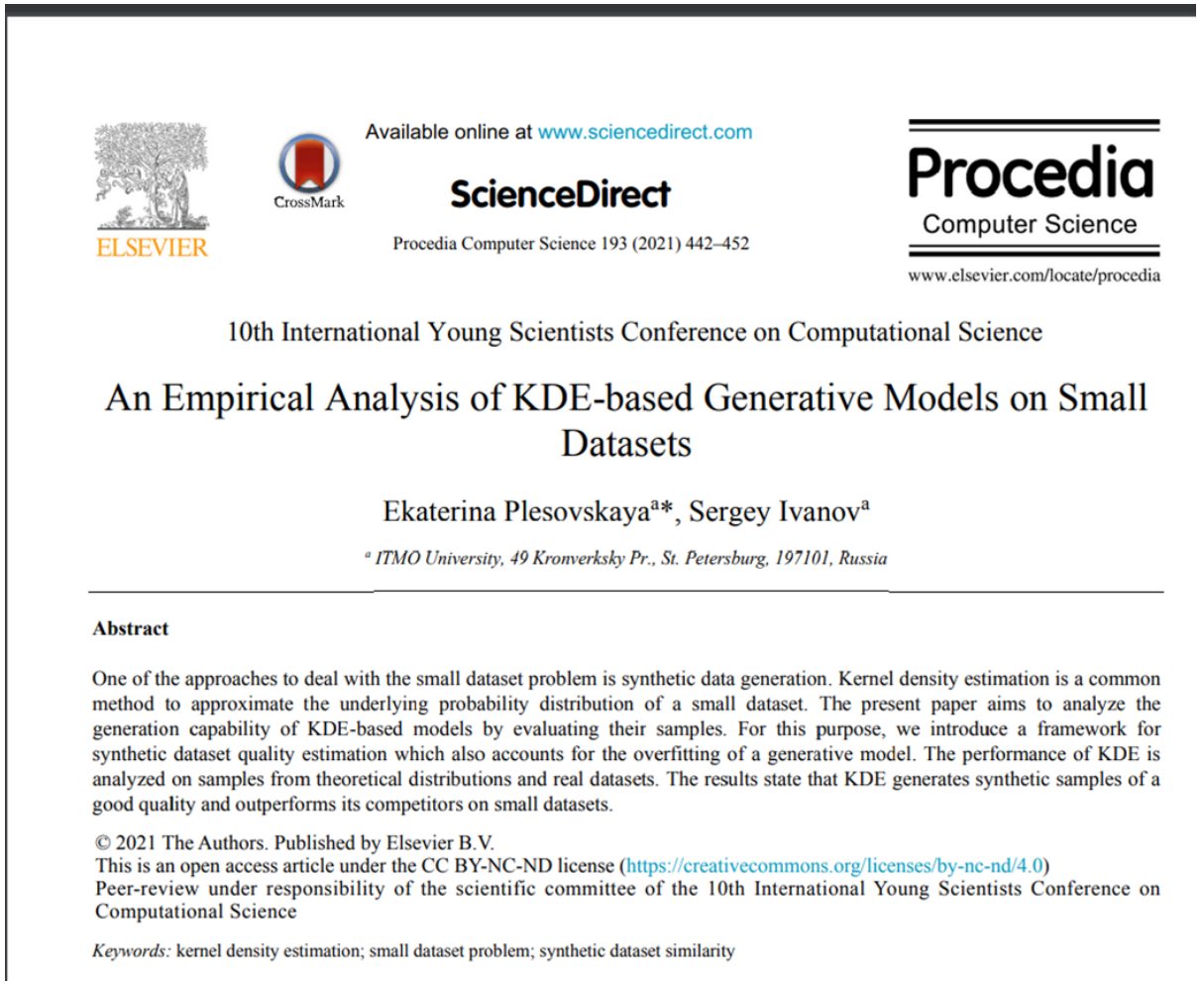
Keywords: Generative model; kernel density estimation; deep learning; probability density function

1 Introduction

In recent years, generative models have received tremendous attention and have been extensively applied of the time it is hard to capture the true distribution of training datasets in a parametric way, e.g., with analytical expressions. This is especially the case for high-dimensional data, such as images that usually consist of thousands of pixels.

Based on such motivation, in this paper, we propose a novel generative model based on kernel density estimation, which directly learns the distribution of training datasets in a non-parametric way. In statistics, kernel density estimation is a non-parametric model to estimate the probability density function of a random variable. In theory, it is capable of modeling the true probability density function of any high-dimensional dataset with its kernel and bandwidth properly selected. Therefore, the difference between the estimated probability density of the real training data and the generated samples can serve as a measure of the performance of a generator. In other words, we are able to generate more realistic samples by properly designing a generator to intentionally decrease such difference.

The rest of this paper is organized as follows. Section 2



ELSEVIER

CrossMark

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 193 (2021) 442–452

Procedia Computer Science

www.elsevier.com/locate/procedia

10th International Young Scientists Conference on Computational Science

An Empirical Analysis of KDE-based Generative Models on Small Datasets

Ekaterina Plesovskaya[a]*, Sergey Ivanov[a]

[a] *ITMO University, 49 Kronverksky Pr., St. Petersburg, 197101, Russia*

Abstract

One of the approaches to deal with the small dataset problem is synthetic data generation. Kernel density estimation is a common method to approximate the underlying probability distribution of a small dataset. The present paper aims to analyze the generation capability of KDE-based models by evaluating their samples. For this purpose, we introduce a framework for synthetic dataset quality estimation which also accounts for the overfitting of a generative model. The performance of KDE is analyzed on samples from theoretical distributions and real datasets. The results state that KDE generates synthetic samples of a good quality and outperforms its competitors on small datasets.

Keywords: kernel density estimation; small dataset problem; synthetic dataset similarity

[1] J. Tan, S. Yin and S. Zhao, "Generative Model with Kernel Density Estimation," 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), 2018, pp. 304-308, doi: 10.1109/ICNIDC.2018.8525628.
[2] E. Plesovskaya, S. Ivanov, “An Empirical Analysis of KDE-based Generative Models on Small Datasets,” Procedia Computer Science, 2021, pp. 442-452

Conclusion

- Kernel density estimation (KDE) sampling model performs well.
- The model able to generate synthetic dataset similar distribution as Monte Carlo(MC) dataset.
- Used semi-supervised leaning to quantify fake signal and reduce model dependencies.
- In KDE performance worsens exponentially with high dimensional data sets, this phenomenon is called “curse of dimensionality”.
- Continue investigation for better KDE performance on new datasets.