ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA
Dipartimento di Informatica - Scienza e Ingegneria
Master degree in Artificial Intelligence

Project work: Artificial Intelligence in Industry

SYNTHETIC DATA GENERATION USING GENERATIVE MODELLING

Student Name:	Faisal Ramzan
Matriculation ID:	0001005787
Presented To:	Professor Michele Lombardi

ABSTRACT

Synthetic data generation is a very complex and challenging task because the replicated data must be having similar statistical properties to the original data. To achieve such a type of statistical characteristics in newly generated data, we focus on the deep learning generative modeling techniques called Generative Adversarial Networks. Generative Adversarial Networks (GANs) are based on neural networks architecture which is having many efficient results in deep learning, initially, their implementation provides very good results in image translation and generations of artificial images. Later this will be used in the financial field to generate time series and other types of synthetic data. The purpose of this task is to generate the synthetic data using GANs networks and then evaluate the quality of synthetic data to the original data. As the domain of finance, there are many constraints and regulations on original data because of the privacy and confidentially of users' data. Organizations are unable to share the original data because of privacy issues and many others issues regarding the misuse, stealing of original data etc. Another issue is when we have a few amounts of data or low dimensional data, sometimes when the real data do not exist completely, so we only rely on the synthetic data or artificial data to fulfill the needs and requirement of the given task. Through generative modeling, we generate artificial data that is statistically very similar and highly correlated to the original distribution of data. In GANs model, two important components are called generator and discriminator. The generator is responsible to generate the fake samples and the discriminator is performing the classification task between the real and fake samples. The performance of both can be measured through the loss function which is divided into two types known as generator loss and discriminator loss. After the generation of the synthetic samples, we can measure the quality of generated data in terms of mean, similarities, and correlations between the original and synthetically generated data.

Keywords

Generative adversarial networks, Generator G, Discriminator D, Synthetic data, Latent Point, Artificial data, minmax loss function.

1. INTRODUCTION

In this era, many organizations use different machine learning techniques to process and organize a large amount of data. The data is very important for all organizations like medical, educational, and financial fields like stock markets, banking systems, and insurances, etc. The main concern is the privacy, security, and protection of the data because the enterprise's level of organizations having high confidential data about their clients, customers or about financial transactions, etc. Because of the privacy and protection of the data, they didn't share their original data with external consultants or with the organizations in order to fear about the tempering and stealing of the original data. Sometimes these organization needs to measure their financial losses, make their competitors business analysis, or doing some future observations and experiments on that business data.

In the past, some researcher uses statistical and mathematical techniques like the randomization-based methods or Bayesian network model to generate synthetic data or correlated data to fulfill their requirements but after the emergence of deep learning technology, it introduces many methodologies to deal with this type of problem. With the introduction of generative modeling, we use different neural network models like VAE's (Variational Auto-encoder's) and GAN's (Generative adversarial networks) to deal with this type of issue with great accuracy and performance.

As in the current work or in some situations when we do not have enough data for training the model and we require a lot of similar data or artificial data to training the model. In this work, we use GAN's (Generative adversarial network) to produce the artificial data that is having high correlation and probabilistically very similar distribution to the original dataset. The purpose of the synthetic or fake generated data is to increase the worth of small size of data, also for to doing business analysis or measuring the financial losses without having any privacy or confidentially issue. This task is specifically about to the financial domain because we are working on the generation of the synthetic data of stock market index as similar to the training data. It is unsupervised learning task; we have real dataset from stock market indexes that will act as training datasets but the size of the training dataset is very small. The structure of the given dataset contains 746 rows having four columns without any given labels. The task is to find an appropriate transformation that can produce such type of data that is similar to the given dataset.

2. Background

Generative Adversarial Networks Since GANs were proposed in 2014 by Ian Goodfellow. The main idea of GANs network is based on two Neural Network models called Generator G and the discriminator D, this is also called two-player minimax games. Overall abstract of generative modeling is that the generator of the model is responsible to produce the synthetic or fake sample using the latent noisy data, while on the other end the discriminator D of the model is to classify between the real and fake samples. In the whole scenario the generator and discriminator compete each other in order to maximize their performance. During the initial training phase, generator G is not performing well to generate the synthetic sample and the discriminator D can easily catch that it was a fake sample. While the training of the generator, the generator performs progressively better in producing the fake sample, and it fooled the discriminator to make mistakes during classification, until the discriminator is unable to recognize that it was a real or fake sample. The main purpose of the generator is to produce a fake sample that is similar to the real dataset and the discriminator is to classify between real and fake samples. Both of these models are defined by multi-layer perceptron and the whole network is trained through the backpropagation technique.

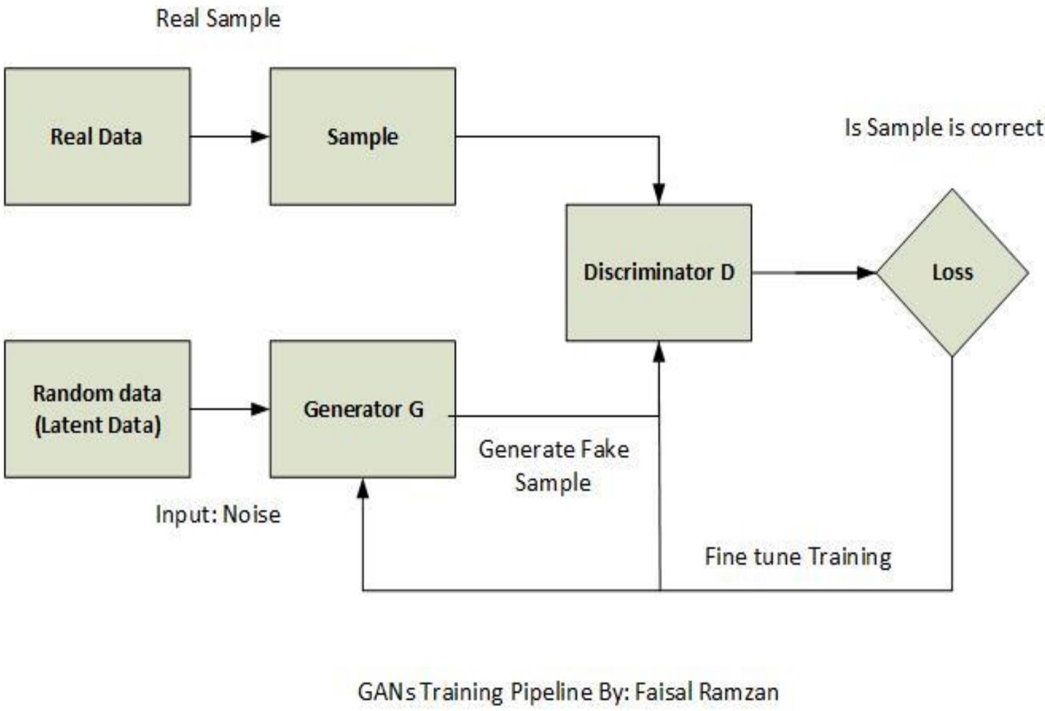
Generative modeling has an initial implementation on image data to produce such type of images or fake images that is having no existence in the real word. Here I give some reference to the site (https://thispersondoesnotexist.com/), this site is responsible to generate the fake sample that is having no existence in the world. While with the emergence of the deep learning modeling techniques GANs and VAEs can also deal with other type of problem which are related to different fields like financial, educational or medical.

3. System Description

Modelling in finance is a very complex and challenging task because of statistical properties and very complicated internal working are largely unknown. When we are facing such type issue or we have not sufficient data to train our model then we select the deep learning generative modeling technique to deal with that type of issue. With the help of Generative adversarial networks (GANs) and Variational Autoencoders (VAEs) techniques can curb with this type of problems. Both of these having a very good result in image generation and later these are also successfully applied in the other fields like synthetic time series data generation in the field of finances.

Architecture:

Generative adversarial networks (GANs) consist of the two parts, the Generator G that takes a latent noisy point from latent space and produce the synthetic or fake data. The second model discriminator D called a classifier that classify the samples that came from real data or from the fake dataset. The task of G is to generate the fake sample that is very similar to the real distribution while the discriminator identifies the difference between real and fake data like police who can identify the difference between the real and fake currency. Generator try to produce optimized samples that fool the discriminator while D is optimized in case of find the real and fake samples.



See Figure (1): architecture of GAN’s

Latent space:

Latent space is responsible to generate random noise points in the space and reshaped them into such dimension that generator take those point as an input. Generator G take takes latent point as an input and produce the fake sample, the label of those fake point will be 0. From the real data we select some random real sample and label of those point is 1.

Discriminator:

The GANs model work as the two-player game, the discriminator D is a classifier that receive the real and synthetic point from the generator G and start to classify between the real and fake samples. Discriminator D is simple sequential keras model including dense layers, it depends on the designer of the model that how many numbers of layers used in the model, the purpose of these layer to extract the input features and the output of these layers are activated with “Relu” function. The activation function is responsible to transform the feature into the next layer. Rectified linear activation function can work on only positive number otherwise it will return with 0 (if x>0 return 1 else 0). The final or output layer in the discriminator model can be activated by the “sigmoid” activation function because this layer is used for the classification purpose to discriminate the input sample are real (1) or fake (0).

Discriminator model connect with two loss function (Generator Loss, Discriminator loss) which is used in different part of training the model. After the classification between the real and fake samples, the loss function penalizes the discriminator for misclassification of the points. we use backpropagation technique to optimize the network model and minimize the loss function, by this technique we update the network weights and then predict again with updated network parameters.

Generator:

The generator G can use the outcome from the discriminator, the generator can generate the data that can be classified by the discriminator as the real data or real samples. The neural network architecture of the generator model is similar to the discriminator. The generator takes the random latent noisy point as an input and then those point can be classified from the discriminator and result in a generator loss which is penalize the generator because of non-fooling the discriminator model. By using the noisy point, Generator can potentially produce a wide range of output which is used as input to the discriminator.

Loss Function:

The purpose of the loss function is to measure and evaluate the performance of the model, GANs loss function measure the distance between the real and synthetic data by assessing their similarities, how both of these distributions look like in term of mean and standard deviations, etc. The GANs loss function called “Vanilla” which is also called the minimax loss function. The evaluation of the minimax loss can be used from the cross-entropy loss function to evaluated the difference between the real and generated data distribution by the JS(Jensen-Shannon) divergence when the discriminator is optimal. The JS divergence measures the similarities between two probability distributions. For the generator G loss, minimize loss is equal to the minimizing,

Log (1 -D(G(z)), so it can’t directly affect the Log D(x).

Generator and discriminator measure the distance between the probability distribution,

$$\mathbf{G\ min\ D\ max\ V\ (D, G) = Ex\ [\ log\ D(x)] + Ez\ [\ log\ (1 - D(G(z)))]}$$

Meaning of the terms used in the equations:

- D(x) discriminator which estimate the probability of real instance X.
- Ex Expected value of all real instances.
- G(z) Output of generator when given noise or latent point as an input for generating the fake or synthetic samples
- D(G(z)) Classification of discriminator about the real or fake instance.
- Ez expected value for all random inputs to the generator.

Parameter Optimization:

Through the generator and discriminator losses functions we evaluate the performance of the generative model. In supervised learning task when we aren’t successfully generating or compute the output which is similar or close to the actual output then we need to update the parameters of the model like bias and wights using the gradient decent techniques. After updating the model parameter, we train the model again with updated weight matrixes and predict the model output. The whole process is called backpropagation which is used for the minimization of the loss function and finding the regulating values of the parameters. The another important hyperparameter which control the updating speed of model called learning rate.

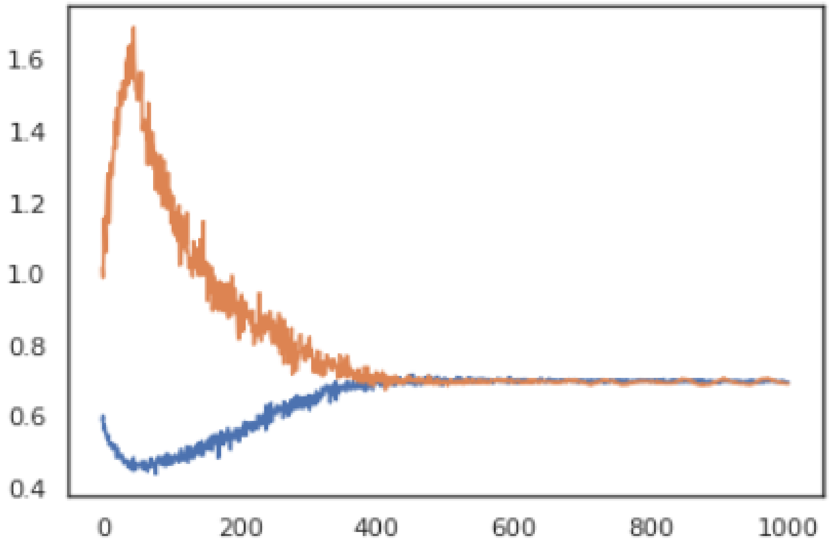
For the given unsupervised learning task, we don’t know the actual output of the given problem but for this unsupervised we learn data and generate the output that is similar to the given input feature. Our model used gradient decent optimizer to find the local minimum of differentiable function by small number of iterations. We use most popular “Adam” (Adaptive moment estimation) optimizer which is used to update the parameter of the network iteratively and it is also the extension of gradient decent.

Model Training:

Both Discriminator (D) and Generator (G) network models are trained separately, by different alternating processes. The training flow of the model is that the generator G take a random noise or latent point z as an input and the G generator produce a new artificial sample. Discriminator can classify the given sample is “real” or “fake” then D calculated a loss for misclassification. By using backpropagation technique for minimizing the loss function, update the parameters of model through gradient decent and re-calculated the loss function with the new or updated parameters. We also need to set some important hyper parameter like the number of epochs, batch size, early stopping and the learning rate etc. Finally, we will compute the average loss by combing the half batch of real data and half batch of fake data.

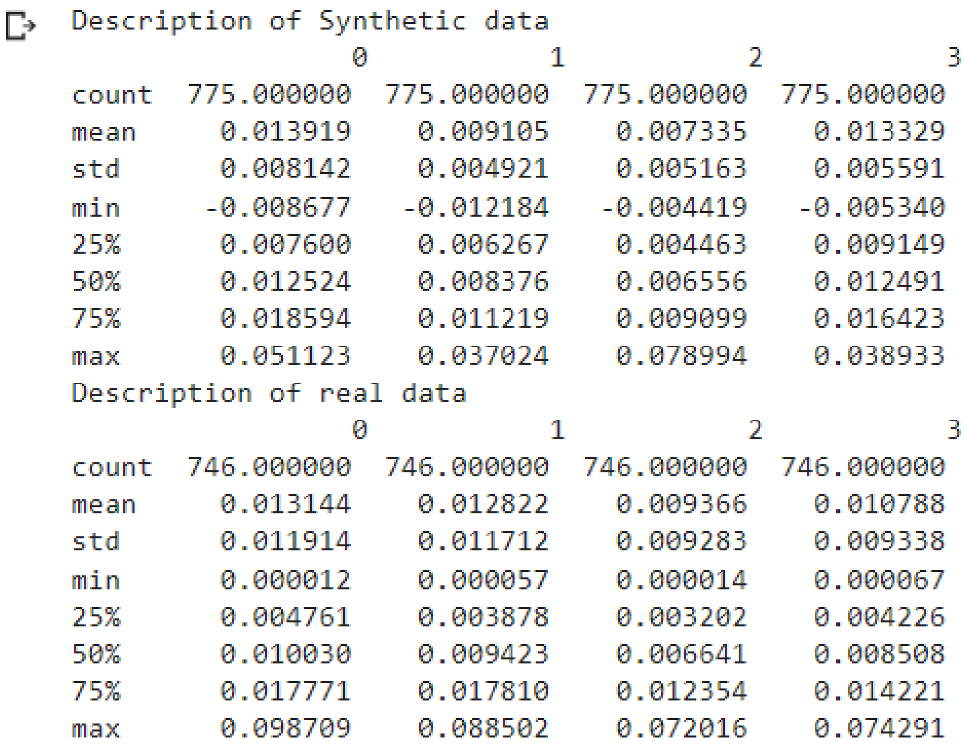
4. Experimental setup and results

After training the model, the losses of the generative model which is given below as shown in Fig (2), the blue color indicates the loss of real sample while the orange shows the loss of synthetic data. The given graph shows the loss of the real and fake samples, initially the loss of synthetic generated data increase but after some of the epoch’s loss diverge and continuously decreases with the increasing rate of epochs. After the 400 epochs the loss of real and fake is stable until the total number of epochs as shown below,



See Figure (2): Losses of real (blue) and synthetic samples (orange).

After training we are going to evaluate the result of the given GAN model regarding the generation of the synthetic data. The given task is unsupervised so our main focus is on the similarity and the quality of the generated. In the term mathematically and statistically, we describe the mean standard deviation, minimum, maximum and the percentile of the real and fake date by using NumPy methods. By observing the mean, std of both the data we drawn as a conclusion that both of real and fake that are very similar and close to each other as show in Fig (3).

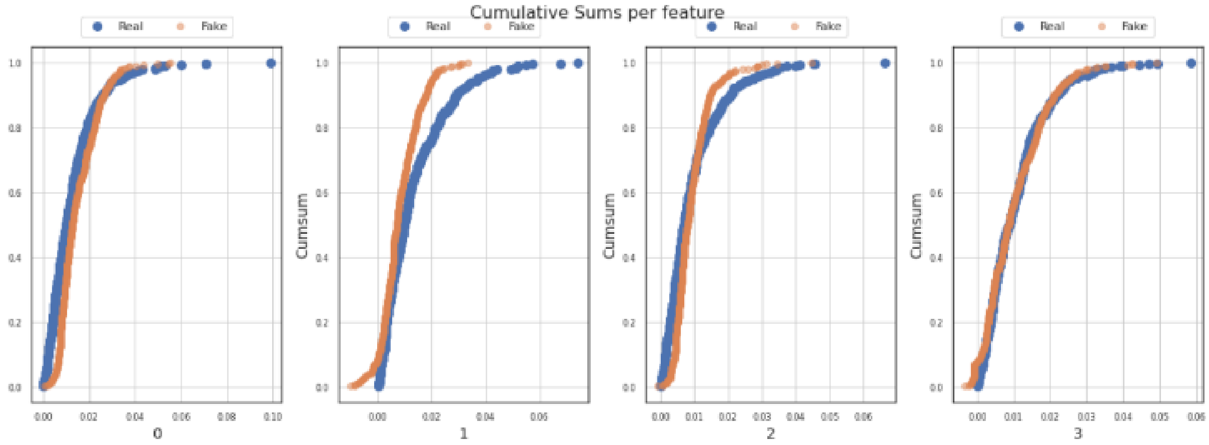


```
Description of Synthetic data
                 0           1           2           3
count   775.000000  775.000000  775.000000  775.000000
mean      0.013919    0.009105    0.007335    0.013329
std       0.008142    0.004921    0.005163    0.005591
min      -0.008677   -0.012184   -0.004419   -0.005340
25%       0.007600    0.006267    0.004463    0.009149
50%       0.012524    0.008376    0.006556    0.012491
75%       0.018594    0.011219    0.009099    0.016423
max       0.051123    0.037024    0.078994    0.038933
Description of real data
                 0           1           2           3
count   746.000000  746.000000  746.000000  746.000000
mean      0.013144    0.012822    0.009366    0.010788
std       0.011914    0.011712    0.009283    0.009338
min       0.000012    0.000057    0.000014    0.000067
25%       0.004761    0.003878    0.003202    0.004226
50%       0.010030    0.009423    0.006641    0.008508
75%       0.017771    0.017810    0.012354    0.014221
max       0.098709    0.088502    0.072016    0.074291
```

See Figure (3): Description of real and synthetic data.

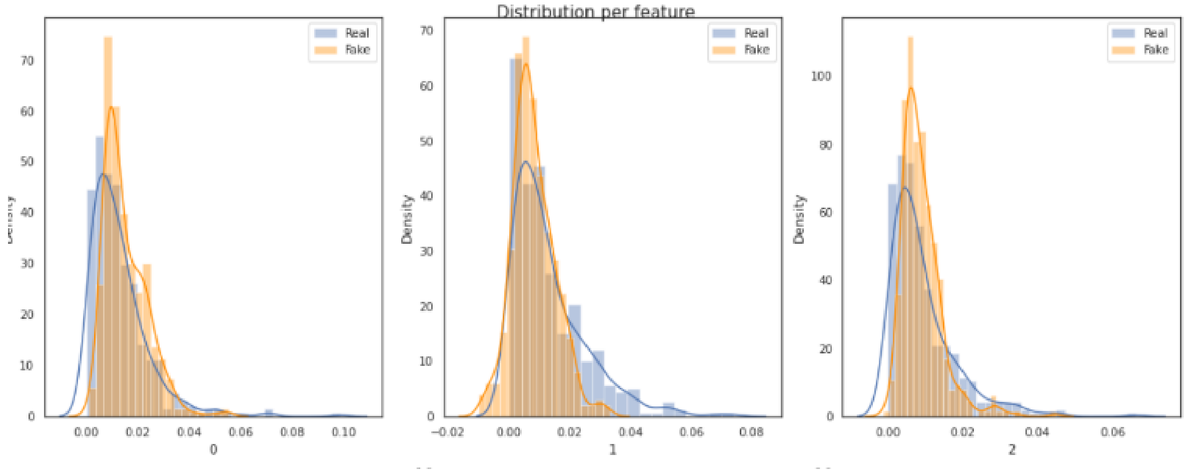
As in the above table, mean of both distributions represent the average of all data that is very similar to each other vice versa the standard deviation and other. For better understanding and checking the quality of data we use python tabular evaluator method which shows the similarity between both distribution in term of cumulative frequency and correlation. This task is practically implemented in python and showing all result regarding the similarities and quality of the generated data.

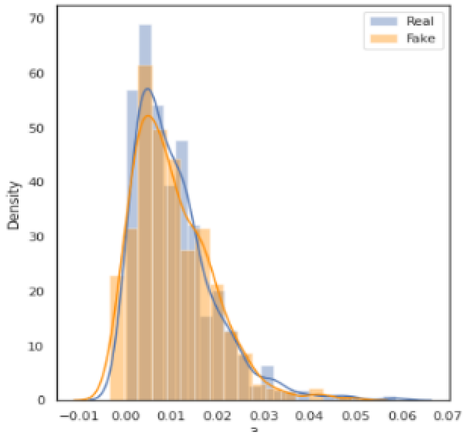
The cumulative frequency distribution showing the frequency of points that belong to which class limits or range, it also represents in the categorical or ordinal representation of the features. The graph of cumulative sum of frequency distribution is given below in Fig (4),



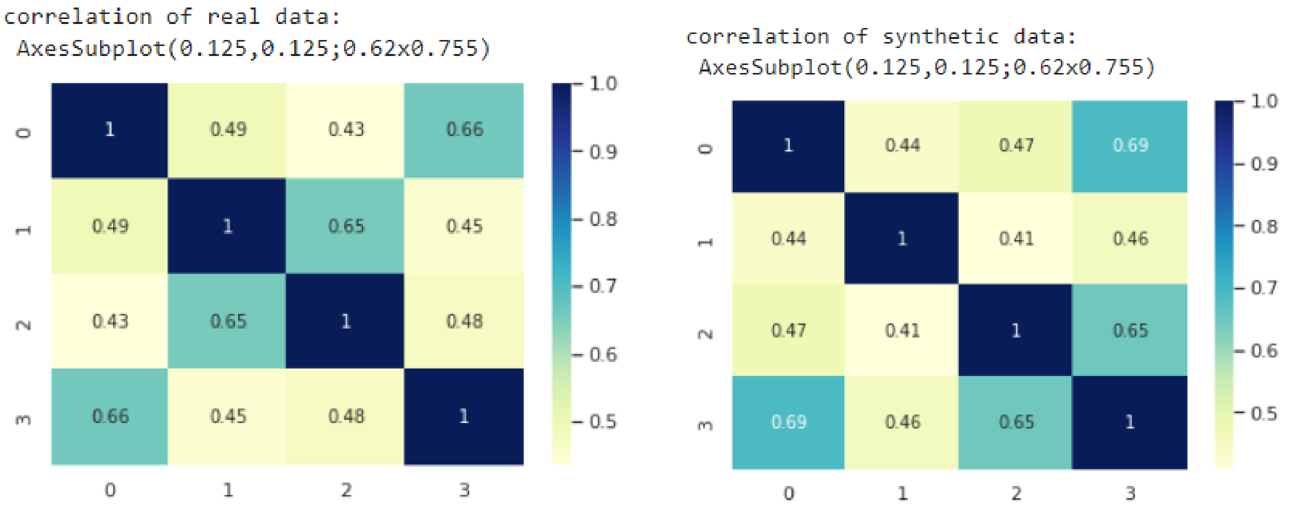
See Figure (4): Cumulative sum frequency per feature.

Tabular evaluator in python is very popular tool for representing the graphical distribution of data, by using these methods we can measure the quality of the data. Here, I will show the correlation between the real and fake generated data using GANs model. The below graph shows the data distribution per feature, we have four feature available in the training dataset. The real data distribution is represented in blue color and the color orange is representing the fake data distribution. There is high similarities and correlation between the real and fake generated data distribution as shown below in Fig (5), Fig (6), and Fig (7)





See Figure (5): Real and Fake data distribution



```
correlation of real data:
 AxesSubplot(0.125,0.125;0.62x0.755)
```

	0	1	2	3
0	1	0.49	0.43	0.66
1	0.49	1	0.65	0.45
2	0.43	0.65	1	0.48
3	0.66	0.45	0.48	1

```
correlation of synthetic data:
 AxesSubplot(0.125,0.125;0.62x0.755)
```

	0	1	2	3
0	1	0.44	0.47	0.69
1	0.44	1	0.41	0.46
2	0.47	0.41	1	0.65
3	0.69	0.46	0.65	1

Fig (6): Correlation Matrix of real and synthetic data

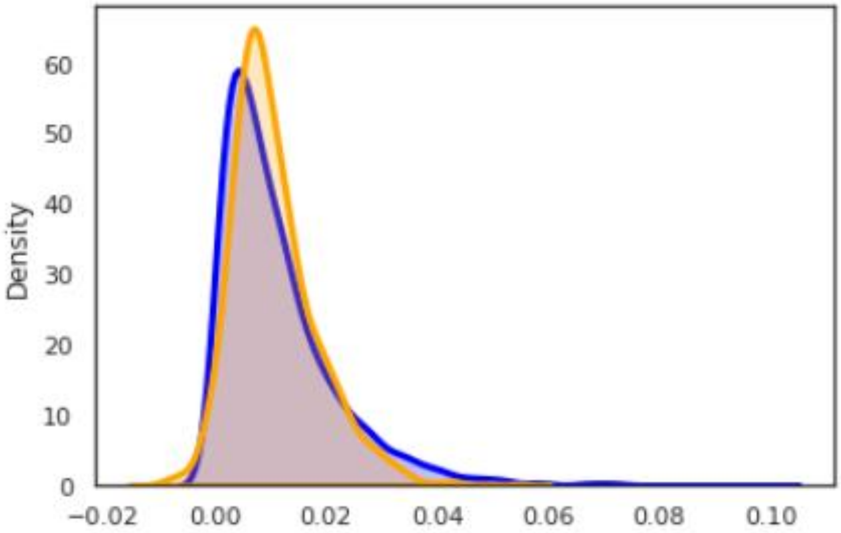
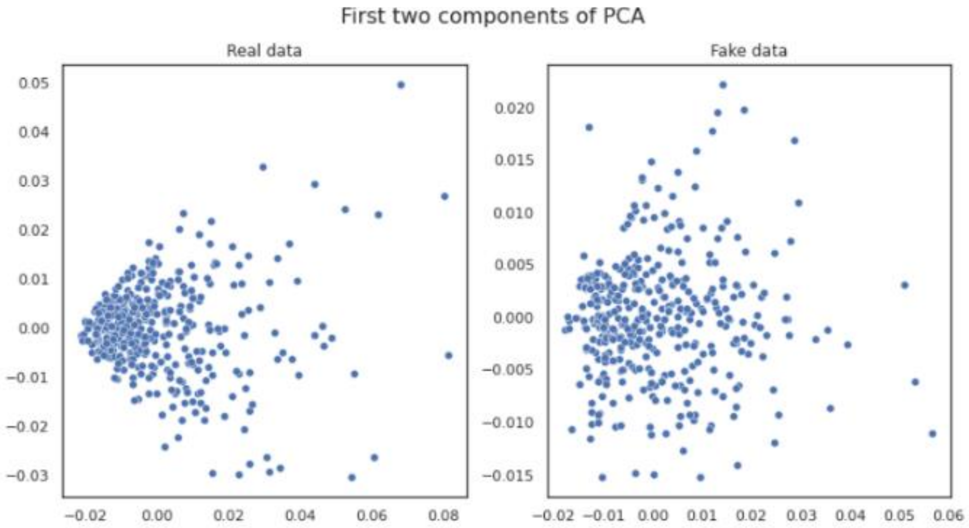


Fig (7): Common plot of both distributions



See Figure (8): PCA of real and fake data

5. Analysis of result

Our task is about "Generative Modeling in Financial Losses" which uses the deep learning generative modeling technique called Generative Adversarial Network for the implementation of this task. Our main focuses are on the generation of synthetic data because in the financial field or other field where we need security and protection of data, also when we have a small amount of data for training the model or for performing some experiments on data. There are many Generative Adversarial Network (GANs) applications which is gaining more popularity day by day in many fields especially in finance, medical and educational sectors.

As in the financial aspect, our problem is to generate the synthetic data that is very similar to the original data. As in the above experimental section, it's a clear view about the efficiency and quality of the generated data. The generated data having a strong correlation with the original data as to see the Fig (H) where the common plot show the distribution of both real and fake data. Principle Component Analysis (PCA) also demonstrate the same distribution of the real and synthetic data. Here we have successfully generated synthetic data in this task, the quality and nature of generated data as shown in the experiment that show the clear difference between the real and fake data.

6. Conclusion

Generative modeling (GANs & VAEs) has a very effective result especially in the generation of fake images, synthetic tabular data generation or time series data generation. The task is about to generate a synthetic data that contain a few extreme values from some given input noise. The GANs model learn from the real data and generate the synthetic data which is similar to the original distribution of data. The purpose of the synthetic data is to enhance the size of training data or sometime when the real data is small or does not exist then synthetic data is only the solution to curb with that type of problems. Another is the protection of confidential data because the real data having many constraints due to privacy or some other regulations. So, the synthetic data can adapt all important statistical properties of real data without exposing the original dataset.

7. References

1. **[Generative Adversarial Network by Ian J. Goodfellow]**

https://arxiv.org/abs/1406.2661

2. **[Privacy of the synthetic data]**

https://www.ahajournals.org/doi/10.1161/CIRCOUTCOMES.118.005122

3. **[Methodology of artificial data]**

https://diyago.github.io/2020/03/26/gans-tabular.html

4. **[Generating Realistic Stock Market Order Streams]**

https://arxiv.org/abs/2006.04212

5. **[Synthesizing Tabular Data using Generative Adversarial Networks]**

https://arxiv.org/abs/1811.11264