Information Processing and Management 59 (2022) 102889

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Data augmentation of credit default swap transactions based on a sequence GAN

Xi Fan [*], Xin Guo, Qi Chen, Yishuang Chen, Tongyao Wang, Yuxin Zhang

University of California, Berkeley, USA

ARTICLE INFO

Keywords:
Generative adversarial network
Synthetic ranking agreement index
Credit default swap transaction

ABSTRACT

Credit default swap transaction data repositories are frequently applied with credit default swap spread estimation and financial market risk assessment. However, in practical applications, there is poor liquidity, some missing data, and inaccurate definitions. Small samples tend to lead to poor prediction accuracy and poor adaptability of the statistical algorithm. Data generation can effectively increase the sample size and improve the effect of the risk assessment model. In this paper, a credit default swap data generation algorithm based on a sequence generative adversarial network (SeqGAN) is proposed, and the policy gradient algorithm in reinforcement learning is introduced to optimize the traditional generative adversarial network (GAN) algorithm to solve the gradient disappearance and poor data adaptability problems in the traditional algorithm. Gradient disappearance is due to the generator network in GAN being designed to be able to adjust the output continuously, which does not work on discrete data generation. The optimization algorithm proposed in this paper is used to train randomly distributed sequence data and generate credit default swap transactions with diversity and good model applicability. The credit default swap data generated in this paper are verified by the synthetic ranking agreement (SRA) index. The results show that SeqGAN can effectively synthesize various simulation samples, which can provide support for the risk discrimination model.

1. Introduction

The credit default swap market, one of the world's most important financial markets, can be regarded as insurance against financial asset default. Creditors sell their debt risk through CDS contracts, the price of which is a premium. The party who purchases credit default insurance is called the buyer, and the party who bears the risk is called the seller. Both parties agree that if there is no default event defined in the contract, the buyer will pay the "insurance premium" to the seller regularly, and the seller will bear the asset loss of the buyer in case of default (Xinjiang and Wenting, 2014). Usually, the contract includes various default situations, such as the financial asset debtor going bankrupt for liquidation, the debtor failing to pay the interest on schedule, the creditor violating the rules requiring the creditor to recall the debt principal and requiring advance repayment, and debt restructuring. Speaking of the CDS insurance premium depends on the estimation of the creditor's risk by various financial organizations. In the CDS market, the risk estimation officials are called the credit default swap spread (CDS).

CDSs are traditionally estimated by analyzing the accounting information provided in financial reports with each company(Tang and Yan, 2016). There is a positive relationship between accounting information and CDSs. There is more risk exposed in accounting

[*] Corresponding author.
E-mail address: xifan96@berkeley.edu (X. Fan).

https://doi.org/10.1016/j.ipm.2022.102889
Received 3 August 2021; Received in revised form 24 November 2021; Accepted 28 January 2022
Available online 14 February 2022

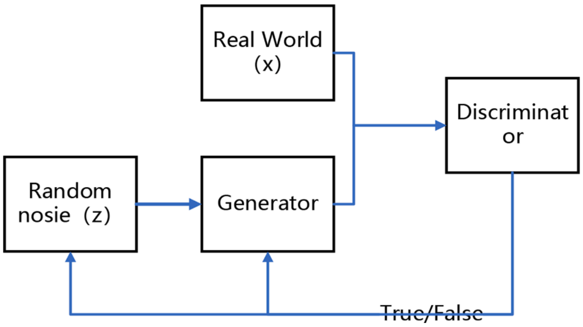


Fig. 1. The original GAN structure.

information the lower CDSs are. Credit default swap contracts have grown in prominence in 2018, with a notional value outstanding of $11 trillion today. As the CDS market grows maturely across the world, researchers discover that imbalanced supply and demand situations can affect CDS estimation. An increasing buying demand can raise the CDS price, and a sufficient supply of sellers might reduce prices. Currently, CDS estimation depends not only on accounting information but also on the CDS transaction data repository. In other words, traders are potent to exploit CDS transaction data mining to help them make more profits and bear fewer risks. The CDS transaction data repository (SDR) operation is similar to stock transaction data. The price and frequency of bid-ask behaviors are composed of transactions. CDS products are widely used by enterprises to hedge risks in developed countries such as Europe and the United States. They have set up mature market mechanisms and analysis algorithms.

However, CDS is difficult, and the CDS market developed slowly due to more bond defaults (Abad et al., 2016; Augustin, Subrahmanyam, Tang, & Wang, 2014). Moreover, there is not much financial or political information in the CDS open market. Even considering unpublished accounting information, CDS products have not been popular in recent years. The core reason may be that there are few institutions actively involved in providing credit protection to sellers, and the concept of rigid payment for bond investment is still deep. Furthermore, transaction market activity is poor, and CDS application is still constrained by various factors, such as low default rates relative to the market volume, a fundamental bond market with little liquidity, and scarce proper reference marks. These difficulties eventually reflect that unreasonable price estimation based on barren risk hedging tools prevents investors from being incentivized to hedge credit risk. Even worse, sellers have loose interests in providing credit default insurance. The pricing of the CDS spread relies on the risk discrimination model to evaluate the risk of sellers' accounting information and the transaction market data repository (Lantao, Weinan, & Jun, 2017). In addition to policy factors, the scarcity of CDS market data and opacity hinder the further improvement of the CDS risk discrimination model. This paper argues that the optimized CDS risk discrimination model through transaction data augmentation can help sellers make accurate pricing and promote the application of CDS in the global financial market.

With the rapid development of deep learning, many data augmentation problems find solutions. Deep learning is a mature and advanced machine learning implementation(Guo et al., 2019). Due to its powerful feature extraction ability, deep learning can reconstruct the structural features of trained images and generate several branching algorithms. At present, the known generated adversarial network (GAN) has superior innovation and accuracy in image data reconstruction(Shao, Huang, & Gao, 2020; Yang, Hui, & Chen, 2019; Zhong, Li, & Clausi, 2019). Generative adversarial networks include two neural networks playing against each other. One is the generator and the other is discriminatory. It is characterized by the confrontation between the internal discriminator and the generator. Initially, the implicit equation extracts the original data features of the dataset, and then, the summarized data features are fed back to the user in the form of target-generated data through the generator. In the training, the discriminator network and the generation network play against each other. The goal of the generator is to generate high-quality images that can deceive the discriminator. Emulative data augmentation is based on such a deceived discriminator(Han et al., 2018).

The original GAN algorithm has good prospects in the field of computer vision(Gong et al., 2017). However, unlike continuous image data, financial transaction data are discrete and random. In this paper, SeqGAN is adopted as the model framework to help model training iterations with policy gradients to generate new CDS data(Liang and Tang, 2019). To verify the CDS data quality, a synthetic ranking agreement was used to evaluate the discrete data according to the diversity and model applicability requirements. SRA indicators are obtained based on five machine learning models: random forest, logistic regression, linear support vector regression, polynomial naive Bayes and Gaussian naive Bayes. The experimental results show that with training and iteration, SeqGAN can generate sufficient and diverse synthetic CDS transaction data to help select a suitable algorithm.

2. Generative adversarial network

2.1. Original generative adversarial network (GAN)

We choose generative adversarial networks (GANs) as our primary model due to their impressive performance and creativity in generating. To be more detailed, the GAN model is one of the most promising approaches to generating synthetic data that mimics the real data after understanding and analyzing the tremendous amount of real information. Generative adversarial networks include two

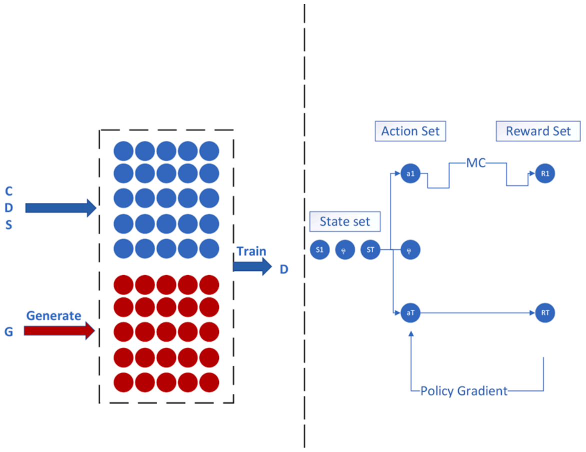


Fig. 2. SeqGAN structure.

neural networks playing against each other(Goodfellow et al., 2014). One is the generator (G), and the other is the discriminator (D). The architecture of the original generated confrontation model is shown in Fig. 1. The input data of G are random noise (z), and the output is G (z). The input data of D are the real data x and generated synthetic data G (z). The output of D is a binary classified value. If the output is 1, D thinks that all the inputs are real data. If the output is 0, D considers the input G(z) to be generated data. G is optimized with feedback from D, and the goal of G is to make the distribution p_z of G(z) close to the distribution p_{data} of the real data x.

The game between the generator and discriminator can be regarded as a minimax problem, and the objective function is shown in Eq. (1).

$$\underset{G}{min}\underset{D}{max}E_{x\sim p_{data}}\left[\log D_{(x)}\right] + E_{z\sim p_z}[\log(1-D(G(z)] \tag{1}$$

The original GAN uses an unsaturated formal function to train the model, and Eq. (2) is obtained.

$$\begin{aligned}&\underset{G}{\min}\underset{D}{\max}V(D,G) = \int p_{data}(x)\times\log D(x)+\\&\int p_z(z)\times\log(1-D(G(z)))\end{aligned} \tag{2}$$

According to Goodfellow, the loss function of minimizing G is equivalent to minimizing the Jensen–Shannon divergence between p_z and p_{data} (Kasem, Hung, & Jiang, 2019; Salimans et al., 2016). Simultaneously, Eq. (3) is set up.

$$C(G) = \underset{D}{\max}V(D,G) \tag{3}$$

If and only if p_z becomes the same as p_{data}, $C(G)$ reaches the minimum value$-log4$ so that we obtain Eq. (4).

$$C(G) = -\log 4 + 2\times JS\left(p_{data}\parallel p_g\right) \tag{4}$$

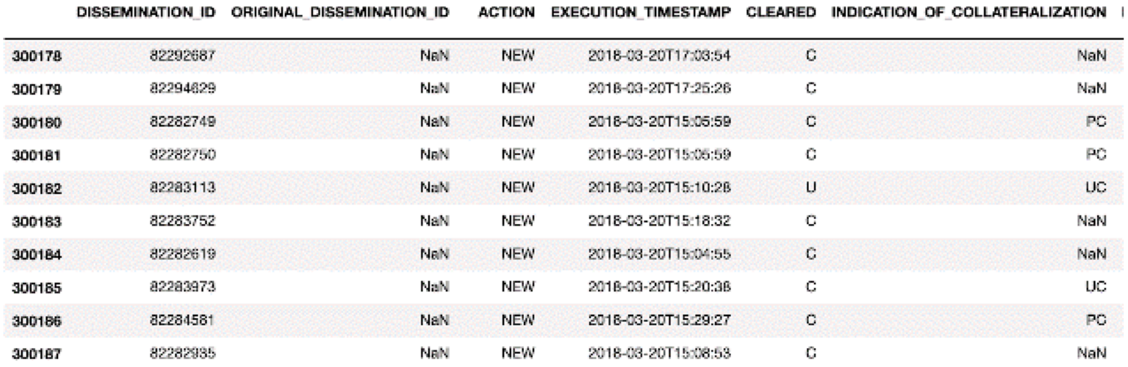
$JS(P_{data}\parallel P_g)$ indicates the Jensen–Shannon difference of the distribution of real data p_{data} and the distribution of generated data p_z. Since the loss function based on JS divergence deduction tends to cause the gradient to disappear and is not suitable for discrete data, this paper uses the policy gradient algorithm to facilitate training.

2.2. *Sequence generative adversarial network (Goodfellow et al., 2014) (SeqGAN)*

The policy function is the probability density function. When the policy is applied, it is selected according to the probability distribution in the action set () $A_{1:T}$ $=(a_1,a_2,a_3,\dots,a_t,\dots a_T)$. The generator generates sets $G(a_t|A_{1:T-1})$ based on selected actions. Then, both real CDS transaction data and generated data are input into D_φ, and the output reflects the probability of confidence $D_\varphi(A_{1:T})$. It utilizes Monte Carlo methods to obtain an optimized action and then increases the similarity between P_{data} and P_g through a policy gradient. Starting from the initial state, each time a choice is made, the model advances to the next state and is accordingly rewarded. In the process, we can obtain state set $S_{0:T}=(s_0,s_1,s_2,\dots,s_t,\dots s_T)$ and reward set $R_{0:T}$ $=(R_1,R_2,R_3,\dots,R_t,\dots R_T)$. The maximum cumulative reward leads to the optimal solution and we obtain Eq. (5).

$$\begin{aligned}J(\theta) &= E[R_T|s_0,\theta]\\&=\sum_{a_1\in A}G_\theta(a_1|s_0)Q_{D_\varphi}^{G_\theta}(s_0,a_1)\end{aligned} \tag{5}$$

R_T is the final reward; $Q_{D_\varphi}^{G_\theta}$ is a reward function while staying at state $S_{1:T-1}$ selecting the next action among the action sets to



	DISSEMINATION_ID	ORIGINAL_DISSEMINATION_ID	ACTION	EXECUTION_TIMESTAMP	CLEARED	INDICATION_OF_COLLATERALIZATION	I
300178	82292687	NaN	NEW	2018-03-20T17:03:54	C	NaN	
300179	82294629	NaN	NEW	2018-03-20T17:25:26	C	NaN	
300180	82282749	NaN	NEW	2018-03-20T15:05:59	C	PC	
300181	82282750	NaN	NEW	2018-03-20T15:05:59	C	PC	
300182	82283113	NaN	NEW	2018-03-20T15:10:28	U	UC	
300183	82283752	NaN	NEW	2018-03-20T15:18:32	C	NaN	
300184	82282619	NaN	NEW	2018-03-20T15:04:55	C	NaN	
300185	82283973	NaN	NEW	2018-03-20T15:20:38	C	UC	
300186	82284581	NaN	NEW	2018-03-20T15:29:27	C	PC	
300187	82282935	NaN	NEW	2018-03-20T15:08:53	C	NaN	

Fig. 3. Some of the CDS transaction data.

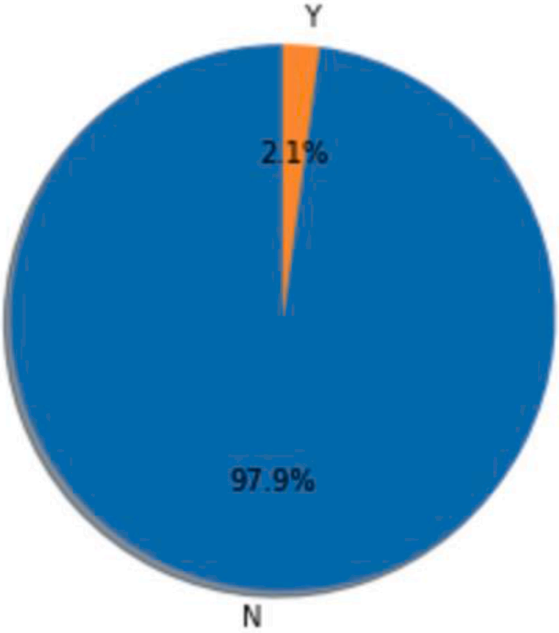


Fig. 4. Example of feature distribution.

generate dataG_θ $(a_t|A_{1:T-1})$. Monte Carlo simulates the reward on each move before obtaining the final reward.

$$Q_{D_\varphi}^{G_\theta}(S = S_{1:t-1}, a_t) = \begin{cases} \frac{1}{N} \sum\limits_{A_{1:T} \in MC_{G_\theta}(A_{1:T})} D_\varphi(A_{1:T}) t < T \\ D_\varphi(A_{1:T}) t = T \end{cases} \tag{6}$$

The algorithm architecture used in this article is shown in Fig. 2. The Monte Carlo search provides a trajectory from the initial state to the results so that the total rewards of the discriminator can be calculated and the policy gradient updated. The generator iterates through the policy gradient.

The generator can be optimized in each confrontation using $J(\theta)$ as the objective discriminator function. Each time the generator generates a complete CDS transaction dataset, the discriminator is retrained in the following formula.

$$\underset{\varphi}{min} - E_{A \sim P_{data}}[\log D_\varphi(A)] - E_{A \sim G_\theta}[\log(1 - D_\varphi(A))] \tag{7}$$

After the discriminator is optimized each time, the generator G_θ can be optimized by the strategy gradient parameter θ.

$$\nabla_\theta J(\theta) = \sum\nolimits_{t=1}^{T} E_{A_{1,t-1 \sim G\theta}} \left[\sum_{a_t \epsilon A} \nabla_\theta G_\theta(a_t|A_{1:t-1}) \cdot Q_{D_\varphi}^{G_\theta}(A_{1:t-1}, a_t) \right] \tag{8}$$

Table 1
Network parameter setting and structure.

Model	Type	Channel	Step Size	Active Function
D	Conv x2	64–32	3 × 3/2	ReLU
	Fc x3	256–128–64	–	ReLU
	Fc x3	64–128–256	–	Softmax
	(Pool+Conv) x2	32–64	3 × 3/2	ReLU
G	(Pool+Conv) x2	32–64	3 × 3/2	ReLU
	Conv		3 × 3/2	Tanh

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta) \tag{9}$$

α represents learning rate.

3. Credit default swap transactions

We obtained trading data from the Depository Trust & Clearing Corporation website (SDR Services, 2019))by using HTTP requests. The data were stored daily, from 01/01/2014 to 02/27/2019, covering all trading actions of each product reported to the management agency within this time range. Because of the over-the-counter market structure and the lack of a central clearing mechanism, there is no single transparent or comprehensive data source for CDS transactions. Therefore, our findings and conclusions can be undermined by this data constraint. It is clear that our conclusions may be affected by enterprise account information and that we focus only on the market level, ignoring the firm level (Yin, 2019).

The original dataset has 1103,307 records and 44 features. However, there is considerable null data in our original datasets. Some of the categorical data are very imbalanced. For example, 98% of the data in “INDICATION_OF_END_USER_EXCEPTION” feature is N. This means that most of the data have the same value. Here, only 2% of the data is ‘Y’, which are outliers of normal observations. In statistics, such imbalanced feature cannot provide useful information while analyzing and building models that we should remove it.

We exclude those data consisting of 80% of null data and 7 features most relevant to financial information are selected as the final features including action, taxonomy, collateralization, cleared, price, amount, and the type of the order. In particular, the features “cleared” and “collateralization” describe the financial product risk. In all kinds of financial products, there are potential credit risks. The risk problem becomes even worse in the case of the CDS index (CDX) since both buyers’ and sellers’ sides may have the possibility of failing to achieve their obligations (Aldasoro and Ehlers, 2018). To mitigate the default risk from both sides, there is always a central counterparty clearing (CCP) to serve as the third party to guarantee the transaction and examine the indices. Both sides focus on the CCP to cover the cost if they default. This procedure is called clearing. Additionally, clearing is truly important to both buyers and sellers and to the whole market for controlling systematic risk.

Otherwise, taking taxonomy as a data cleaning example, taxonomy indicates the CDS product category.(Guo et al., 2019) This feature includes 66 category types. Among all of these, we focus on two: CDX.IG (Investment Grade), which includes 125 liquid North American entities with investment-grade credit ratings and CDX.HY (high yield), which includes 100 liquid North American entities with high yield credit ratings. In finance, high yield always corresponds to high risk. IG denotes CDS firms with credit ratings of BBB- and higher, and HY is for BB- and lower. In general, the companies in the CDX. The HY index has a lower credit rating than those in CDX.IG. Additionally, we want to filter the products by their lifetime. The reason is that most products last 3–6 years, and most of their lifetime is centered at approximately 5 years. To reduce unnecessary information, we only keep the products whose lifetime is between 4.5 years and 5.5 years. After that, we sorted our data based on the timestamp and found that some of the orders were canceled immediately after they entered the market, which means that the product no longer existed, so we removed only the two orders. Some of the orders were corrected, and we had to link the order to the original order and update the new information for this product, which can be easily solved by removing the original data. Additionally, we converted price and amount into discrete data. Since we selected a sequence generated model - SeqGAN, it is necessary to ensure that all input data are category type. For price and amount, we manually set bins for each variable and make the number of values in each bin equal. In price, we set [−0.01, 1, 2, 3, 5, 10, 20, 50, 100] to divide the continuous price value into eight bins. For example, (−0.01,1) is the first bin, which was labeled ‘a’. Therefore, we split the amount variable into 14 categories and labeled it from ‘a’ to ‘n’ following alphabetical order.

Considering the redundancy and efficiency of the model calculation, we use a simpler indicator to represent the categories of each variable. In "ACTION", we replace "NEW" with ’NE’, "CORRECT" with ’CR’ and “CANCEL" with ’CC’. Similarly, in "TAXONOMY" and "PRICE_FORMING_CONTINUATION_DATA". {’Credit:Index:CDX: CDXIG’ to ’IG’, ’Credit:Index:CDX: CDXHY’ to ’HY’, ’trade’ to ’T’, ’novation’ to ’N’, ’partial termination’ to ’P’, ’Amendment’ and ’A’}. Adding “Price” and “Amount”, one sequence of one row can be displayed as NE,U,UC,IG,T,a,m, which represents (‘NEW’, ‘uncleared’, ‘UC’, ‘Credit:Index:CDX: CDXIG’, ‘trade’, ‘1.00′, ‘100.0′). Finally, we have 27,196 rows of prepared CDS transaction input.

4. Model training and performance

The network structure is shown in Table 1. The network training adopted a small-batch training method with a batch number of 50 (Liao et al., 2020). At the beginning of the training, the generator and discriminator were trained separately, generator and

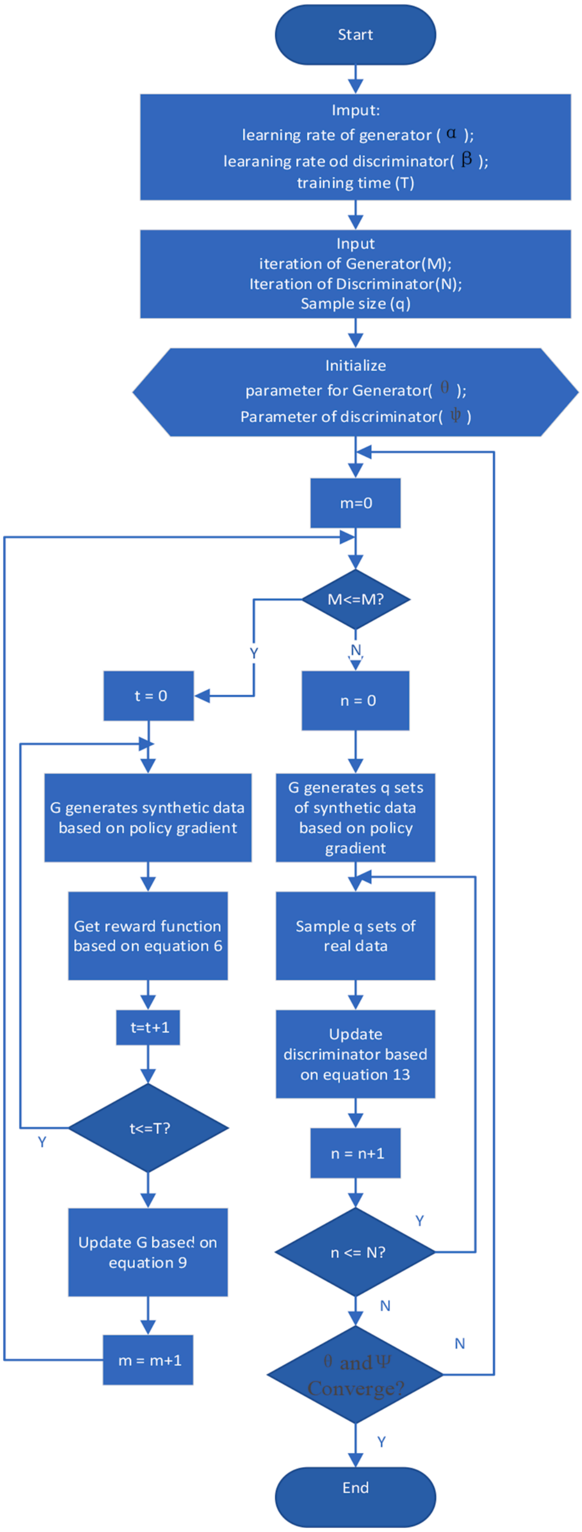

Start
Imput:
learning rate of generator (α);
learaning rate od discriminator(β);
training time (T)
Input
iteration of Generator(M);
Iteration of Discriminator(N);
Sample size (q)
Initialize
parameter for Generator(θ);
Parameter of discriminator(ψ)
m=0
M<=M?
Y
N
t = 0
n = 0
G generates synthetic data based on policy gradient
G generates q sets of synthetic data based on policy gradient
Get reward function based on equation 6
Sample q sets of real data
t=t+1
Update discriminator based on equation 13
t<=T?
Y
n = n+1
Update G based on equation 9
n <= N?
Y
N
N
m = m+1
θ and Ψ Converge?
Y
End

Fig. 5. Model structure.

discriminator iterations were conducted during the training, and the number of epochs was 100.

As shown in Figs. 5 and 6, the generator and discriminator pretraining loss functions gradually approach 0.5, and then the generator and discriminator are alternately trained. The two models continue to train until the seqGAN model converges. As shown in

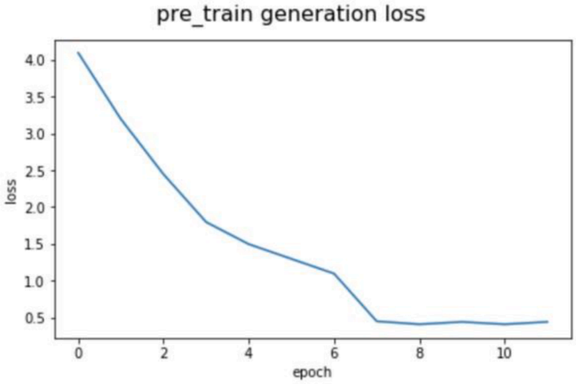


Fig. 6. Pretrained generator loss.

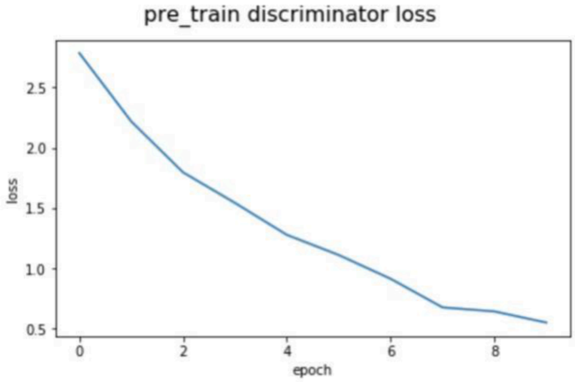


Fig. 7. Pretrained discriminator loss.

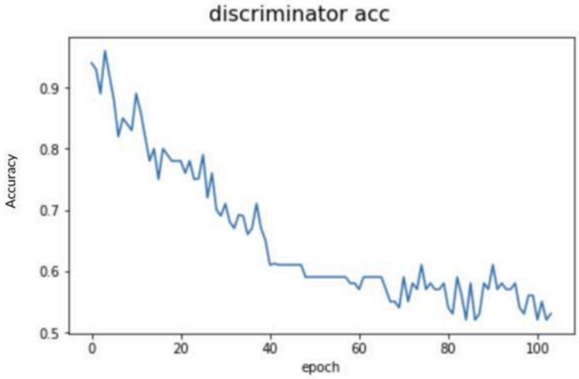


Fig. 8. Discriminator accuracy of the trained discriminator.

Fig. 7, after 100 iterations, the discriminator's accuracy gradually converges to 50%. This means that the discriminator cannot distinguish between real and simulated data, and the generated data are very similar to the real data(De et al., 2020).

A well-trained generator is used to generate credit default swap transaction data and decode the alphabetic sequence. We need to decode our letter-based sequence data (NE,U,UC,IG,T,a,m) back to the readable financial data ('NEW', 'uncleared','UC', 'Credit:Index: CDX:CDXIG', 'trade', '1.00′, '100.0′). The next step is to determine whether the data we generate have high quality. We evaluate our data in 2 aspects: whether the numerical proportion holds and calculate the SRA score to determine whether the relative performance of different algorithms holds in synthetic data.

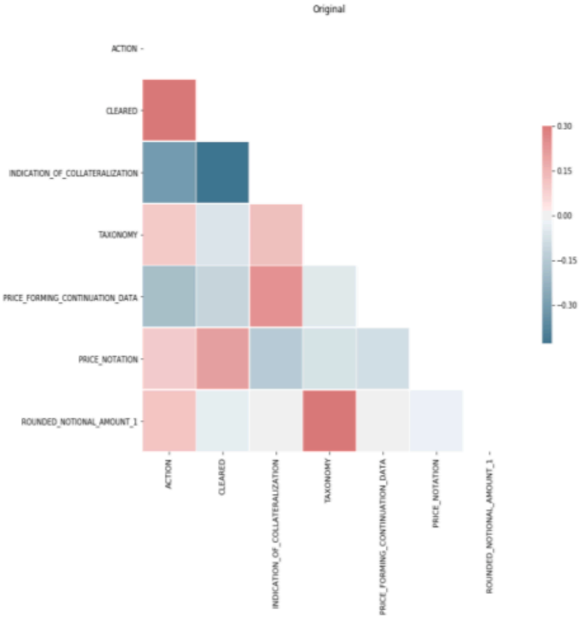


Fig. 9. Original data correlation matrix.

5. Synthetic data evaluation criteria

5.1. Correlation

As we said, we explored the correlation between every two features. Here, we plot a correlation matrix between two different variables with our original data that contains every current order. We applied our training model to a completely simulated, new dataset. By analyzing our new data, we compared similarities and differences between the new data and the original data. From the original data, we see the correlation matrix had very small correlations.

In a more mathematical view, we see their correlations numerically below.

With our simulation, we put the data into a new graphical correlation matrix and a new numerical correlation matrix.

These metrics show that the new data we simulated have the same characteristics as the original real data. It is important to note that neither of them has a high correlation between any two variables.

Then, we explored the correlations among multiple features to work on a more mathematical analysis. We performed regression and classification on each feature for feature “price” with logistic regression on that with the other 6 variables. Here are the coefficient significance levels.

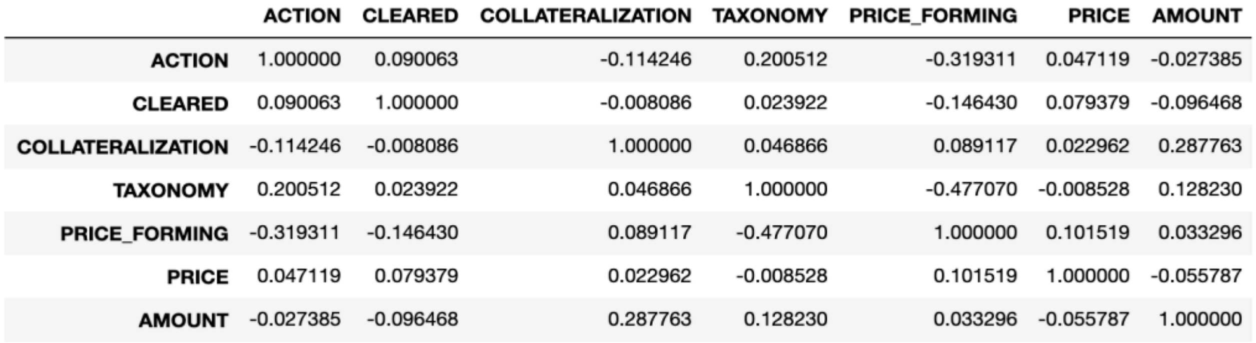
In the figure above, we can observe that some stars are marked on the right hand of the feature. The number of stars indicates the level of significance. A three-star label indicates the most significant dependency, and a nonzero label means nonrelevance. Here, the feature “price” shows a high dependency with feature collateralization due to three-star labeling. We concluded that price is highly related to collateralization, taxonomy, and the order type (“price forming continuation data”).

With the same logic, we performed the regression and classification on our simulated fake data, and we obtained the following results.

Then, we performed classification on the other categorical data for a more general idea. With the outputs, we determine that there exists high dependency among the variables. When classifying the variables ‘price forming continuation data’, ‘taxonomy’, and ‘cleared’, we see very high accuracies of 0.8572, 0.9317, and 0.8496, respectively. In general, a higher accuracy value indicates that this feature is more dependent on other features. Here, the ‘taxonomy’ feature is most dependent on other features. For features ‘price forming continuation data’ and ‘cleared’, we also noticed high dependency on other features.

5.2. Numerical proportion

By comparing the proportions of original CDS transactions and synthetic transactions, we can see that numerical proportions in the synthetic data vary from the original. However, the diversity is not random. In each feature, the proportion rank of synthetic transactions still follows the same rank as the original CDS transaction. In Table 2, taking the action of the original data as an instance, category NEW takes the largest proportion and cancel takes the least. So does the synthetic data.



	ACTION	CLEARED	COLLATERALIZATION	TAXONOMY	PRICE_FORMING	PRICE	AMOUNT
ACTION	1.000000	0.090063	-0.114246	0.200512	-0.319311	0.047119	-0.027385
CLEARED	0.090063	1.000000	-0.008086	0.023922	-0.146430	0.079379	-0.096468
COLLATERALIZATION	-0.114246	-0.008086	1.000000	0.046866	0.089117	0.022962	0.287763
TAXONOMY	0.200512	0.023922	0.046866	1.000000	-0.477070	-0.008528	0.128230
PRICE_FORMING	-0.319311	-0.146430	0.089117	-0.477070	1.000000	0.101519	0.033296
PRICE	0.047119	0.079379	0.022962	-0.008528	0.101519	1.000000	-0.055787
AMOUNT	-0.027385	-0.096468	0.287763	0.128230	0.033296	-0.055787	1.000000

Fig. 10. Original data numerical correlation matrix.

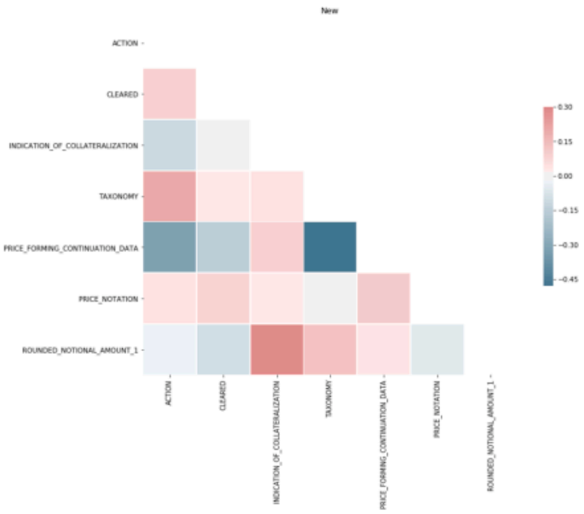
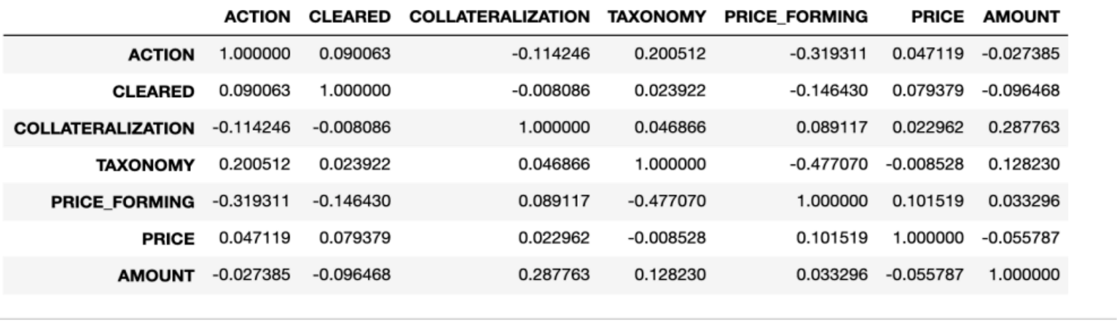
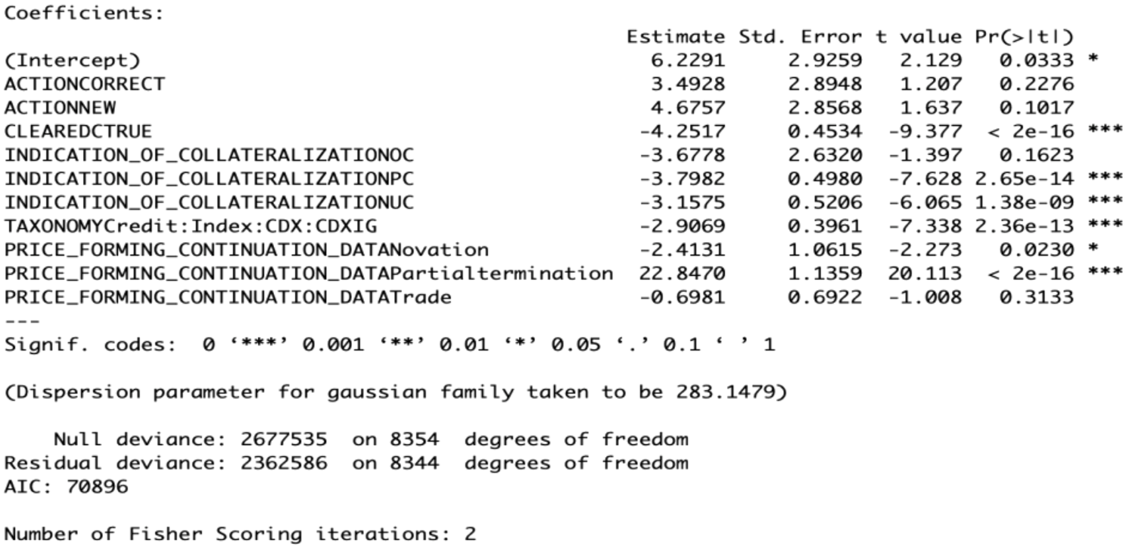


Fig. 11. New data correlation matrix.



	ACTION	CLEARED	COLLATERALIZATION	TAXONOMY	PRICE_FORMING	PRICE	AMOUNT
ACTION	1.000000	0.090063	-0.114246	0.200512	-0.319311	0.047119	-0.027385
CLEARED	0.090063	1.000000	-0.008086	0.023922	-0.146430	0.079379	-0.096468
COLLATERALIZATION	-0.114246	-0.008086	1.000000	0.046866	0.089117	0.022962	0.287763
TAXONOMY	0.200512	0.023922	0.046866	1.000000	-0.477070	-0.008528	0.128230
PRICE_FORMING	-0.319311	-0.146430	0.089117	-0.477070	1.000000	0.101519	0.033296
PRICE	0.047119	0.079379	0.022962	-0.008528	0.101519	1.000000	-0.055787
AMOUNT	-0.027385	-0.096468	0.287763	0.128230	0.033296	-0.055787	1.000000

Fig. 12. New data numerical correlation matrix.



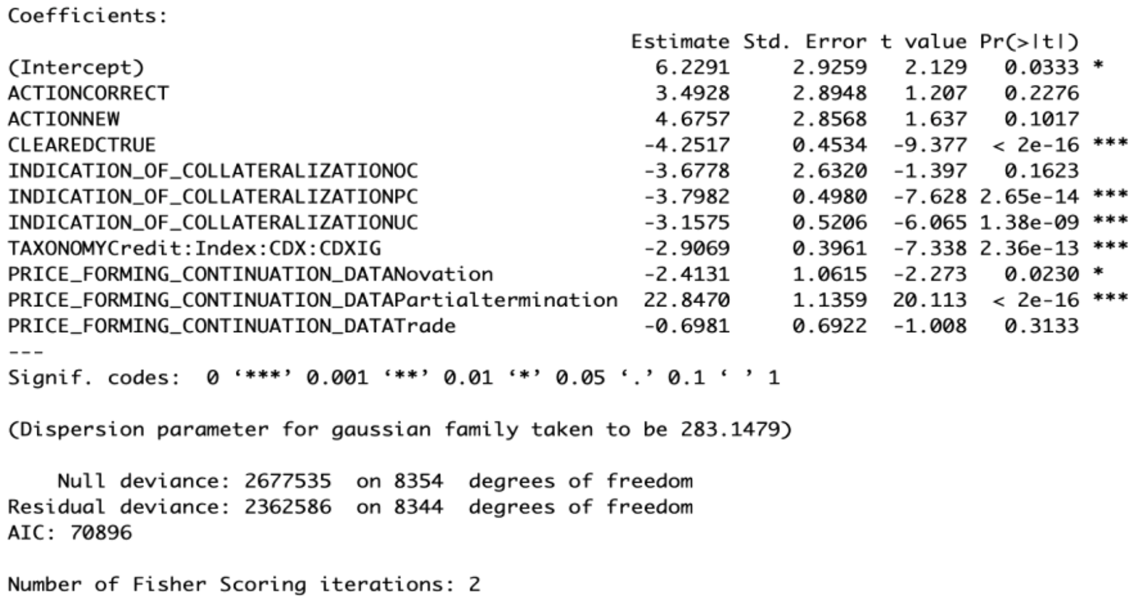
```
Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)                                         6.2291     2.9259   2.129   0.0333 *  
ACTIONCORRECT                                       3.4928     2.8948   1.207   0.2276    
ACTIONNEW                                           4.6757     2.8568   1.637   0.1017    
CLEAREDCTRUE                                       -4.2517     0.4534  -9.377  < 2e-16 ***
INDICATION_OF_COLLATERALIZATIONOC                  -3.6778     2.6320  -1.397   0.1623    
INDICATION_OF_COLLATERALIZATIONPC                  -3.7982     0.4980  -7.628 2.65e-14 ***
INDICATION_OF_COLLATERALIZATIONUC                  -3.1575     0.5206  -6.065 1.38e-09 ***
TAXONOMYCredit:Index:CDX:CDXIG                     -2.9069     0.3961  -7.338 2.36e-13 ***
PRICE_FORMING_CONTINUATION_DATANovation            -2.4131     1.0615  -2.273   0.0230 *  
PRICE_FORMING_CONTINUATION_DATAPartialtermination  22.8470     1.1359  20.113  < 2e-16 ***
PRICE_FORMING_CONTINUATION_DATATrade               -0.6981     0.6922  -1.008   0.3133    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 283.1479)

    Null deviance: 2677535  on 8354  degrees of freedom
Residual deviance: 2362586  on 8344  degrees of freedom
AIC: 70896

Number of Fisher Scoring iterations: 2
```

Fig. 13. Original logistic regression data result.



```
Coefficients:
                                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                                    6.2291     2.9259   2.129   0.0333 *  
ACTIONCORRECT                                  3.4928     2.8948   1.207   0.2276    
ACTIONNEW                                      4.6757     2.8568   1.637   0.1017    
CLEAREDCTRUE                                  -4.2517     0.4534  -9.377  < 2e-16 ***
INDICATION_OF_COLLATERALIZATIONOC             -3.6778     2.6320  -1.397   0.1623    
INDICATION_OF_COLLATERALIZATIONPC             -3.7982     0.4980  -7.628 2.65e-14 ***
INDICATION_OF_COLLATERALIZATIONUC             -3.1575     0.5206  -6.065 1.38e-09 ***
TAXONOMYCredit:Index:CDX:CDXIG                -2.9069     0.3961  -7.338 2.36e-13 ***
PRICE_FORMING_CONTINUATION_DATANovation       -2.4131     1.0615  -2.273   0.0230 *  
PRICE_FORMING_CONTINUATION_DATAPartialtermination 22.8470     1.1359  20.113  < 2e-16 ***
PRICE_FORMING_CONTINUATION_DATATrade          -0.6981     0.6922  -1.008   0.3133    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 283.1479)

    Null deviance: 2677535  on 8354  degrees of freedom
Residual deviance: 2362586  on 8344  degrees of freedom
AIC: 70896

Number of Fisher Scoring iterations: 2
```

Fig. 14. New data result of logistic regression.

Table 2
Category proportion of transaction data.

Feature	Value	Original data proportion	Synthetic data proportion
Action	New	74.6%	79.3%
	Correct	25.1%	17.7%
	Cancel	0.3%	3%
Collateralization	UC	46.6%	32.7%
	PC	27.3%	38.7%
	OC	0.5%	6%
	FC	25.6%	22.6%
Price Forming	Trade	85.3%	42.7%
	Partial termination	3.7%	6.7%
	Novation	3.8%	34.2%
	Amendment	7.2%	16.4%
Cleared	U	46.8%	78.7%
	C	53.2%	21.3%
Taxonomy	IG	57.6%	73.5%
	HY	42.4%	26.5%

5.3. Synthetic ranking agreement

With respect to the further application of our synthetic data, we train machine learning models with synthetic data and improve accuracy if the performance rank of the machine learning algorithms from the synthetic dataset remains similar to the original. In other words, when classifying one selected feature with the remaining variables in several machine learning algorithms, we keep the rank of these algorithms' performances on the generated dataset the same as on the original dataset and improve each algorithm performance accordingly with the generated dataset. In this case, we use synthetic ranking agreement (SRA) specifically to evaluate the performance of the generated dataset from given algorithms (Jordon et al., 2018). SAR can be interpreted as the empirical probability of a comparison on the synthetic dataset matching the original data. This paper examines whether the relative performance of the different algorithms on the synthetic dataset (trained and tested) is similar to the applicability of the model on the original dataset (trained and tested) by calculating the generation of the ranking agreement (SRA) scores. Five different machine learning models are used to predict the classified data (Prabhat & Vishwakarma, 2020; Yu, Chang, & Guo, 2020). To test whether synthetic data can achieve a similar contribution to the risk regulation model to real data, we focus on the feature "cleared" describing the risk of financial products. The feature "cleared" is classified using random forest, logistic regression, linear support vector regression, polynomial naive Bayes and Gaussian naive Bayes to determine whether the transaction was cleared. A classifier is a commonly used data model to judge financial risks. We compare the accuracy trained from the raw dataset and the composite dataset to calculate the SRA score and compare the probability of being "correct" for the synthetic transaction data. If the result is close to 1, the quality of the synthetic data is higher. This also means that the more similar the generated data and real data are in the algorithm model.

Table 3
Raw data and generated data test results.

Classifier	Accuracy	
	Original R	Synthetic S
Random Forrest	87%	98%
Logistic Regression	74%	79%
Linear SVM	69%	78%
Poly- Bayes	62%	64%
Gaussian Naive Bayes	57%	32%

$$SRA = \frac{1}{k(k-1)} \sum_{i=1}^{k} \sum_{j \neq 1} \left(\left(R_i - R_j \right) \times \left(S_i - S_j \right) > 0 \right) \tag{10}$$

K is the number of test model algorithms, R is the accuracy of the original data in the model, and S is the accuracy of the generated data in the model. The input analysis results from synthetic transactions are shown in Table 3.

The other six features were used to predict the feature "cleared", and the SRA index of five classifiers was 0.95, indicating that the quality of the data generated by the sequence generation versus network was helpful to select the appropriate algorithm model for the real data.

6. Conclusion

To solve the problems of slow promotion, poor liquidity and difficult pricing of credit default swaps, this paper proposed an auxiliary risk assessment model based on data augmentation.

(1) Discrete financial data such as credit default swap transaction data can be supplemented by a neural network.
(2) The SeqGAN architecture has good training stability and data applicability. To test the model effect of the generated data, the SRA index is proposed to quantitatively evaluate the quality of the generated data. The experimental results show that the credit default swap transaction data generated by the model are similar to those of the real credit default swap repository model.

This paper focuses on how to generate financial data and how to substitute the generated data for real data to provide more possibilities for the credit default swap transaction analysis model. In the next step, the author will seek opportunities to cooperate with financial institutions, combining account information and the risk assessment model with market certification to verify the supplementary effect of generated data on real data, and further look for a credit default swap repository, to improve the generalization ability of the model to meet the requirements of various engineering scenarios.

CRediT authorship contribution statement

Xi Fan: Methodology, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Xin Guo:** Conceptualization, Resources. **Qi Chen:** Data curation. **Yishuang Chen:** Visualization. **Tongyao Wang:** Investigation. **Yuxin Zhang:** Resources.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2022.102889.

References

Aldasoro, I., & Ehlers, T. (2018). The credit default swap market: What a difference a decade makes. *BIS Quarterly Review*. June.
Augustin, P., Subrahmanyam, M., Tang, D., & Wang, S. (2014). Credit default swaps: A survey. *Foundations and Trends in Finance, 9*(1-2), 1–196.
Abad, J., Aldasoro, I., Aymanns, C., D'Errico, M., Fache-Rousová, L., Hoffmann, P., Langfield, S., Neychev, M., & Roukny, T. (2016). Shedding light on dark markets: First insights from the new EU-wide OTC derivatives dataset. *ESRB Occasional Papers*, (11). September.
Guo, Y., Chen, Qi, Chen, J., et al. (2019). Auto-Embedding generative adversarial networks for high resolution image synthesis. *IEEE Transactions on Multimedia*. Retrieved from: https://ieeexplore.ieee.org/document/8676365.
Guo, Q., Li, Y., Song, Y., et al. (2019). Intelligent fault diagnosis method based on full 1-D convolutional generative adversarial network. *IEEE Transactions on Industrial Informatics*. Retrieved from: https://ieeexplore.ieee.org/document/8794731.
Gong, M., Niu, X., Zhang, P., et al. (2017). Generative adversarial networks for change detection in multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*. Retrieved from: https://ieeexplore.ieee.org/document/8094357.
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems, 3*, 2672–2680.
Han, X., Lu, J., Zhao, C., et al. (2018). Semisupervised and weakly supervised road detection based on generative adversarial networks. *IEEE Signal Processing Letters*. Retrieved from: https://ieeexplore.ieee.org/document/8302923.
H. Xinjiang, C. Wenting. The pricing of credit default swaps under a generalized mixed fractional Brownian motion. 2014, 404:26-33.
Jordon, et al. "Measuring the quality of synthetic data for use in competitions." ArXiv.org, 29 June 2018, arxiv.org/abs/1806.11345.

Kasem, H. M., Hung, K. W., & Jiang, J. (2019). Spatial transformer generative adversarial network for robust image super-resolution. *IEEE Access*. Retrieved from: https://ieeexplore.ieee.org/document/8933156.

Liao, X., Si, J., Shi, J., et al. (2020). Generative adversarial network assisted power allocation for cooperative cognitive covert communication system. *IEEE Communications Letters*. Retrieved from: https://ieeexplore.ieee.org/document/9069247.

Liang, J., & Tang, W. (2019). Sequence generative adversarial networks for wind power scenario generation. *IEEE Journal on Selected Areas in Communications*. Retrieved from: https://ieeexplore.ieee.org/document/8895757.

Prabhat, N., & Vishwakarma, D. K. (2020). Comparative analysis of deep convolutional generative adversarial network and conditional generative adversarial network using hand written digits. In *Proceedings of the international conference on intelligent computing and control systems*. Retrieved from: https://ieeexplore.ieee.org/document/9121178.

R. De, A. Chakraborty, R. Sarkar. 2020. Document image binarization using dual discriminator generative adversarial networks. document image binarization using dual discriminator generative adversarial networks. Retrieved from: https://ieeexplore.ieee.org/document/9122442.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and Xi Chen. “Improved techniques for training GANs.” (2016). Retrieved from:https://arxiv.org/pdf/1606.03498.pdf.

"SDR services - real-time dissemination dashboard". 2019. Rtdata. Dtcc. Com. Retrieved from: https://rtdata.dtcc.com/gtr/dashboard.do.

Shao, G., Huang, M., Gao, F., et al. (2020). DuCaGAN: Unified dual capsule generative adversarial network for unsupervised image-to-image translation. *IEEE Access*. Retrieved from: https://ieeexplore.ieee.org/document/9178425.

Tang, D. Y., & Yan, H. (2016). Understanding transactions prices in the credit default swaps market. *Journal of Financial Markets*.

Yang, W., Hui, C., Chen, Z., et al. (2019). FV-GAN: Finger vein representation using generative adversarial networks. *IEEE Transactions on Information Forensics and Security*. Retrieved from: https://ieeexplore.ieee.org/document/8658115.

Yin, R. (2019). Multi-resolution generative adversarial networks for tiny-scale pedestrian detection. In *Proceedings of the IEEE international conference on image processing (ICIP)*. Retrieved from https://ieeexplore.ieee.org/document/8803030.

Yu, W., Chang, T., Guo, X., et al. (2020). UGAN: Unified generative adversarial networks for multidirectional text style transfer. *IEEE Access*. Retrieved from: https://ieeexplore.ieee.org/document/9036960.

Y.U. Lantao, Z. Weinan, W. Jun, SeqGAN:Sequence generative adversarial nets with policy gradient.EB/OL. (2017-08-25) 2019-12-23. https://arxiv.org/abs/1609.05473.

Zhong, Z., Li, J., Clausi, D. A., et al. (2019). Generative adversarial networks and conditional random fields for hyperspectral image classification. *IEEE Transactions on Cybernetics*. Retrieved from: https://ieeexplore.ieee.org/document/8726302.