Generating Syntetic Data for Credit Card Fraud Detection Using GANs

Emilija Strelcenia
Department of Creative Technology
Bournemouth University
Bournemouth, United Kingdom
strelceniae@bournemouth.ac.uk

Simant Prakoonwit
Department of Creative Technology
Bournemouth University
Bournemouth, United Kingdom
sprakoonwit@bournemouth.ac.uk

***Abstract*—Deep learning-based classifiers for object classification and recognition have been utilized in various sectors. However according to research papers deep neural networks achieve better performance using balanced datasets than imbalanced ones. It's been observed that datasets are often imbalanced due to less fraud cases in production environments. Deep generative approaches, such as GANs have been applied as an efficient method to augment high-dimensional data.**

In this research study, the classifiers based on a Random Forest, Nearest Neighbor, Logistic Regression, MLP, Adaboost were trained utilizing our novel K-CGAN approach and compared using other oversampling approaches achieving higher F1 score performance metrics.

Experiments demonstrate that the classifiers trained on the augmented set achieved far better performance than the same classifiers trained on the original data producing an effective fraud detection mechanism. Furthermore, this research demonstrates the problem with data imbalance and introduces a novel model that's able to generate high quality synthetic data.

Keywords—fraud, GANs, synthetic data, class imbalance

I. Introduction

Imbalanced class is one of the most difficult tasks while detecting the credit card fraud. In order to address this problem, [1] introduced two different frameworks, i.e. data-oriented and algorithmic approach. First of all, for algorithmic framework, these scholars used the RF approach, KNN, ID3, and Naïve Bayes for multiple samples, picked from 3 datasets. Moreover, they have selected the most effective classifiers based on misclassification cost with the help of the probability threshold. Apart from the algorithm framework, they have also proposed a data-oriented framework, which deals with the resampling procedures such as SMOTE, under-sampling, and over-sampling. According to the authors of this study, the data-centric method with the help of the over-sampling technique attains the desired results. On the other hand, the under-sampling method achieves inferior results. Furthermore, they have also implemented an algorithmic-based framework employing the F1 score as an evaluation metric and recognized Random Forest as an excellent classifier.

In another study, [2] introduced a framework for handling the imbalanced class issue, with the help of the data mining method. Their approach creates a novel model whenever new data in the system arrives. In addition, the authors have conducted a comparative study on the approaches such as MNET, SVM, RF and sampling techniques such as under-sampling, SMOTE. In addition, they also described the significance of revising a model in non-stationary circumstances to attain desired results. In this study, [2] acknowledged the RF method is the most effective approach when compared with other models.

An empirical study conducted by [3] aimed to address the class imbalance issue. They considered various approaches to address this issue in the credit card based fraud domain. They discussed over-sampling, under-sampling, SMOTE, and cost-sensitive learning threshold techniques. Furthermore, they carried out a comparative study and also identified the impact of the degree of skewed distribution on given classifiers. In addition, they have also conducted their study on the Naïve Bayes approach with multiple degrees of skewed distribution and assessed their findings.

Similarly, [4] introduced an innovative method by using weighted extreme learning machines to address skewed data problems. This approach comprises an improved neural network with a single hidden layer neural network. They assigned several weights to all samples. The findings suggest that their approach is different when dealing with the data imbalance issue. Also, the results confirm that there is a performance improvement when compared with other approaches. Furthermore [5] in their study emphasize that imbalance of class is a regular challenge when dealing the classification task via machine learning algorithms. They argue that this problem is not associated with the detection of fraud in the credit card domain only. Their study was focused on the imbalanced class problem linked with the bankruptcy prediction task. The authors of this study introduced two models to handle the imbalanced class issue. In addition, they developed a hybrid framework of sensitive learning and over-sampling methods. At the beginning of the study, they applied the over-sampling technique over the validation set with the help of an optimal balancing ratio to get optimal output. Additionally, for bankruptcy prediction, they used a cost-sensitive learning model, C-Boost. The data they employed was highly imbalanced with a ratio of 0.0026. Furthermore, the authors of this study stressed the likelihood of model over-fitting by using over-sampling methods as it creates copies of minority classes to balance the data.

II. Related Work

[6] have explored multiple aspects related to GAN. They argued that GANs are a more appropriate and effective framework for handling imbalanced class problems than other sampling models. The authors believe that GAN is highly robust towards overlapping and over-fitting, as GAN understands the hidden patterns of data by utilizing deep networks. Furthermore, they have also emphasized the effectiveness of GANs employing several facets like architectural design, difficulties associated with GAN, multiple variants to address specific traits, application areas and so on. In addition, they also pointed out the empirical study conducted for evaluating GAN with the help of metrics. Moreover, they also performed a comparative study on the performance of GAN with resampling methods such as SMOTE. Their study reveals that GAN is more effective than other resampling methods. The finding of this study reveals that GAN variants such as WGAN and WGAN GP are most suitable to mitigate the above issues.

The study conducted by [7] aimed to review several aspects of GANs. The authors considered GAN variants such as CGAN, and fully connected GAN and explored the pros and cons connected with these GAN variants.

The study by [8] is unique from the above studies as their study examined GAN in theoretical and mathematical approaches. This study provides a deep insight into the training complications linked with GAN variants. In addition, this study has presented 3 different points of view to tackle the problems while training GAN. These points are skills, GAN structure and the objective of the framework. The authors of this study assert that inception score, multi-scale structural similarity, model score and freshet inception distance are the most effective metrics to evaluate the capability of GAN.

[9] focused on the limitation and suitability of GAN while dealing with banking challenges. In their study, they made use of the WGAN GP variant to augment data. The findings of their study noticed a major increase of 5 percent in the recall value of the XG Boost classifier after training on augmented data compared with real-world data training. It is also noteworthy to mention that they detected a decrease in F1 score and precision values.

To sum up the above discussion, the most common challenge while dealing with fraud in the credit card domain is the class imbalance problem. In more recent years, scholars have presented various machine learning techniques to deal with this problem. One of the most popular and effective technique to handle imbalanced class are GANs. Furthermore, many scholars have also proposed various GAN variants to deal with this issue. These developments in GAN are making it the most effective method. However, more research work is needed in future to improve the predictability, efficacy, accuracy and applicability of GAN variants.

III. GANs

GANs, a series of machine learning approaches for generation, was proposed by [10]. These machine learning algorithms obtained much attention due to their efficiency and simplicity. In a brief period, researchers introduced novel variants of the conventional GAN approach. Furthermore, regarding the applicability of GANs, considerable developments were made around various areas such as image creation, computer vision, social media fraud, gambling fraud, credit card-based fraud detection, and others.

The traditional GAN approach estimates models via an adversarial mechanism, in which two neural networks are trained. GANs utilize two neural networks: the Generator G and the Discriminator D networks. The function of G is to input a random noise vector to synthetic data that nearly reflects the actual data. On the other hand, the use of D is to take actual samples and to perform as a teacher that can evaluate the performance of output and check if data is fake or real. G and D are trained in such a way that, through a min-max game, the losses of G get minimized, and the losses of D get maximized [7].

The Discriminator is a classifier that gets real and artificial data from the Generator, and the D tries to discriminate the data. Firstly, the D classifies the real/ artificial data, and secondly, the D penalizes for misclassification.

While the Generator uses the input from the D to learn to generate artificial data that must have the same traits as the original data.

A standard GAN is made of a generator neural network G and a discriminator neural network D. They are trained in competition with each other known as a two-player min-max game. The discriminator network D rebalances it's weights in order to determine real data samples $x \sim pd(x)$ from fake data samples $G(z)$ produced by adding randomly sampled from some distribution via the generator network. Following by the balancing it's weights to trick.

Then the discriminator allocates probability for the case where is a "real" training data sample while the probability of for the case where is a "fake" sample produced by the generator. These two networks are going through the iterative training utilizing the loss function provided by:

$$L_{GAN}(G,D)= E_{x\sim p_d(x)}[\log D(x)]+ E_{x\sim z(x)}[\log(1-D(G(z))] \quad (1)$$

Where G tries to minimize $L_{GAN}(G, D)$ while D tries to maximize it. In practice, the assumptions are replaced by empirical mean values over a mini-lot of samples, while the loss function is further minimized and maximized from the first mini-lot to the next, as in the gradient descent of the mini-lot.

Figure 1 shows the process of preprocessing data and creating balanced data sets. The proposed solution comprises of two neural network classifiers, which are defined as discriminator (D) and generator (G). [10] introduced this type of architecture that was inspired by game theory. Using these neural networks, the GAN generates new data samples that are similar to training data based on the probability distribution model. However by its nature of being a very adaptable and general algorithm, meticulous fine tuning of GANs proved to resolve its drawbacks, which in the end may produce the optimized architecture design that can be applied for various ML purposes.

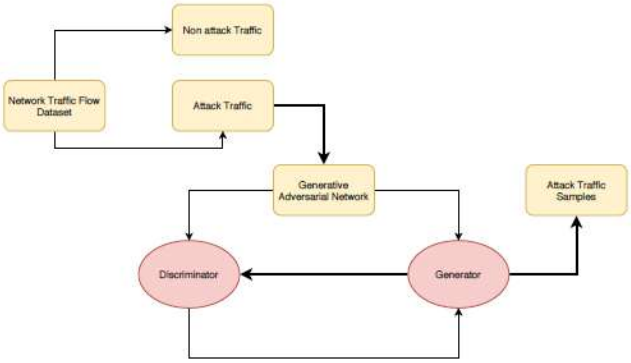

Fig. 1. Process of data preprocessing and balanced dataset generation (Goodfellow, in [10]).

A. Experimental Design

There are a few different loss functions that can be used in GANs, and the choice of which one to use depends on the type of data being generated. For example, if the data is images, then the loss function might be based on the mean squared error between the generated image and the real image. Other types of data might use other loss functions. The most important thing to remember about GANs is that the loss function is used to train the generator, not the discriminator. The reason for this is that the generator is trying to generate data that is realistic enough to fool the discriminator, while the discriminator is trying to learn to distinguish between real and fake data. This means that the generator is trying to minimize the loss function, while the discriminator is trying to maximize it. One common way to think about this is that the generator is trying to find a "sweet spot" in the loss function landscape, where the fake data is realistic enough to fool the discriminator but not so realistic that it is indistinguishable from the real data. The other important thing to remember about GANs is that they are inherently unstable. This is because the generator and discriminator are both trying to learn at the same time, and they are both trying to learn from the same data. This can cause them to "fight" with each other, and can lead to training instabilities. There are a few ways to deal with this, such as using different types of GANs (see below), or using different loss functions. There are various types of GANs, and each one has its own advantages and disadvantages. The most common type of GAN is the vanilla GAN, which is the simplest type of GAN. Vanilla GANs are good for generating simple data, such as images of handwritten digits. They are also relatively easy to train, and don't require a lot of computational power. However, they are not very good at generating complex data, such as natural images. Another type of GAN is the conditional GAN, which is similar to a vanilla GAN but with one additional condition. The condition can be anything, but it is usually something that can help the generator generate more realistic data. For example, if the data is images of faces, then the condition might be the age of the person in the image. This would help the generator generate more realistic images of people of different ages. Conditional GANs are more difficult to train than vanilla GANs, but they can generate more realistic data. In our multiple experiments we've been utilizing CGAN architecture with fine-tuned hyperparameters with the novel loss function.

B. Discriminator Loss

The objective of discriminator network is to maximize likelihood of sample x if belongs to real data and minimize likelihood of sample x if belongs to fake data. The equation below shows the Discriminator loss:

$$\text{Loss} = -\frac{1}{\text{Output size}} \sum_{i=1}^{\text{output size}} y_i . \log \hat{y}_i + (1-y_i) . \log (1-\hat{y}_i) \qquad (2)$$

C. Generator Loss

The objective of generator network is to fool the discriminator by generating fake samples which look like real samples. In our proposed K-CGAN model we've added a new loss term, KL Divergence, to our equation. The difference between two distributions is calculated using the KL divergence. As a result, our Generator loss has two objectives:

- Make the Discriminator fool. We use binary cross entropy for this loss
- Make sure synthetic data distribution is the same as original data distribution. We used KL Divergence for this loss.

The equations below show binary cross entropy and KL divergence losses:

$$-\frac{1}{\text{Output size}} \sum_{i=1}^{\text{output size}} y_i . \log \hat{y}_i + (1-y_i) . \log (1-\hat{y}_i) + \sum p_i(x) \log\left(\frac{p_i(x)}{q_i(x)}\right) \qquad (3)$$

Kilberg divergence is a measures of how close distributions are. Many hyperparameters had to be adjusted to achieve the best performance possible with our proposed method. The hyperparameters below have been identified as the best option after extensive experimenting. The settings we used are shown in Table I and Table II. Learning rate was set to .001, hidden layer optimizer Relu, random noise vector 100. The dropout ratio was set to .1 for both the discriminator and generator hidden layers. Bath size 64 and number of epochs is 100. Relu activation function for the generator and for LeakyRelu for the discriminator. Adam optimizer was defined. We have discovered that by utilizing the Weight Initialization (glorot_uniform) and Weight Regularizer (L2 method) methods, we were able to reduce the size of our neural network during training.

We have also tried to use different Dropout values and found that a value of .1 worked best for our case. Kernel regularizer L2 method worked best for this dataset. This is probably due to the fact that there are many features in this dataset, and some of them are likely to be highly correlated. L2 regularization helps to prevent overfitting by penalizing high weights, and thus encourages the model to find a simpler solution. Figure 2 and 3 demonstrate the architecture of K-CGAN Discriminator and Generator neural networks.

PCA is a good method for dimensionality reduction, but it can sometimes introduce information loss. In this case, we are not too worried about information loss because we are only interested in the class prediction (fraud or not fraud), and not in the details of the individual features.

TABLE I. GENERATOR NEURAL NETWORK HYPERPARAMETER SETTINGS

Parameter	Value
Learning Rate	.0001
Hidden Layer Optimizer	Relu
Output Optimizer	Adam
Loss Function	Trained Discriminator Loss **+ KL Divergence**
Hidden Layers	2 - 128 ,64
Dropout	.1
Random Noise Vector	100
Kernel Initializer	glorot_uniform
Kernel Regularizer	L2 method
Total Learning Parameters	36,837

TABLE II. DISCRIMINATOR NEURAL NETWORK HYPERPARAMETER SETTINGS

Parameter	Value
Learning Rate	.0001
Hidden Layer Optimizer	LeakyRelu
Output Optimizer	Adam
Loss Function	Binary Cross Entropy
Hidden Layers	2 -20,10
Dropout	.1
Kernel Regularizer	L2 method
Total Learning Parameters	1,519

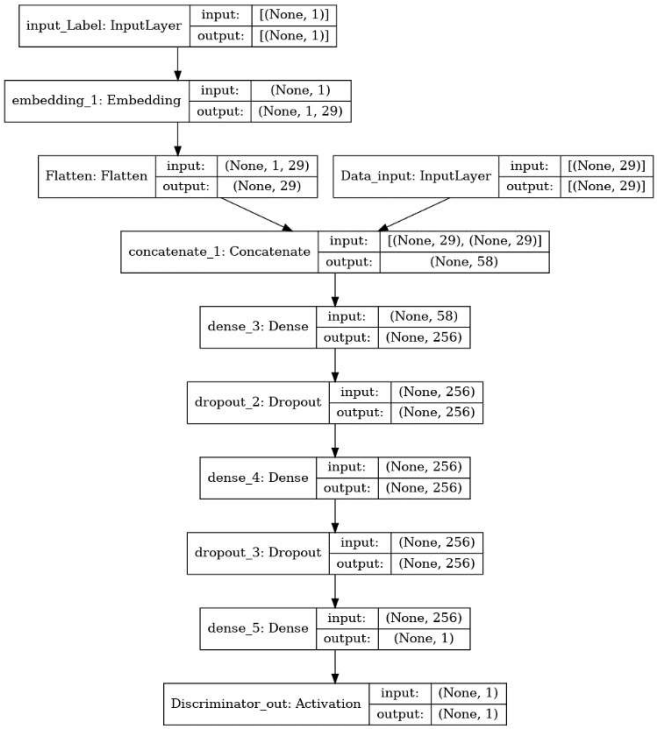

Fig. 2. K-CGAN discriminator architecture.

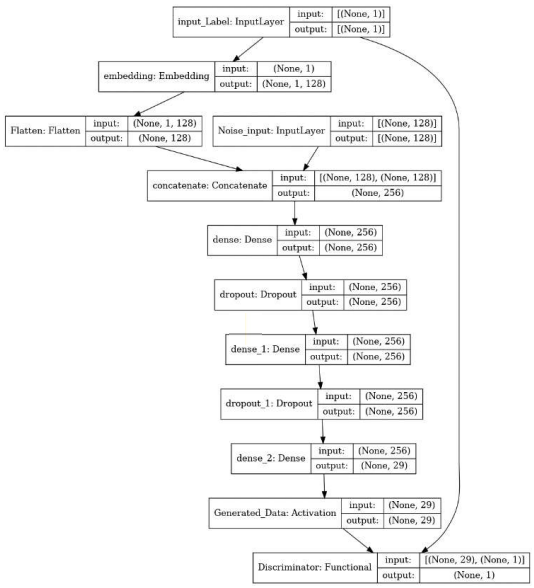

Fig. 3. K-CGAN generator architecture with novelty loss.

We have used publicly available imbalanced Credit Card Fraud dataset from Kaggle.

TABLE III. REAL-WORLD CREDIT CARD DATASET

ID	Data Set	#Features	#Instances	IR
1	Credit Card Fraud	30	2,492	1:4.07

This is a public dataset that can be accessed and downloaded from Kaggle. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,315 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

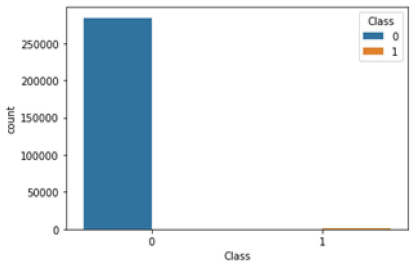

Fig. 4. Original imbalanced dataset (Kaggle).

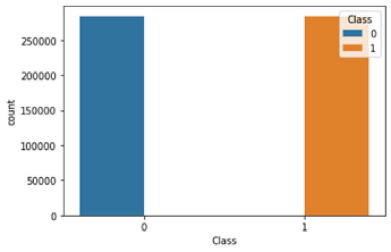

Fig. 5. Balanced dataset showing equal number of minority and majority class samples.

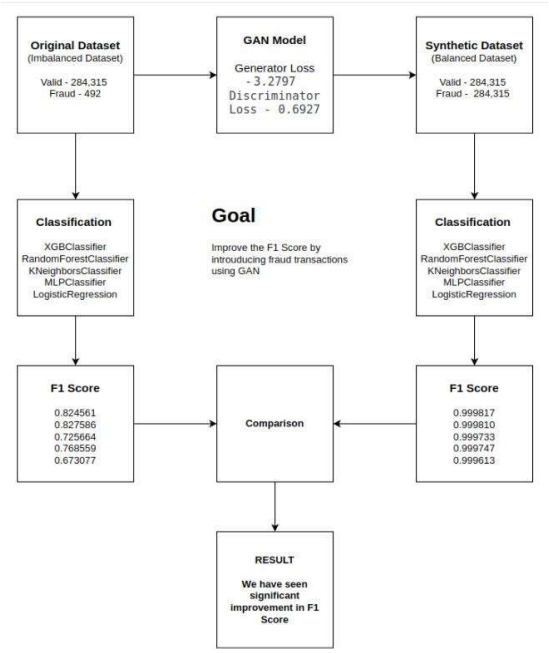

Fig. 6. Flowchart of our experimental process.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Figure 4 demonstrate state of the original imbalanced dataset and figure 5 show state of the dataset upon introducing equal number of samples from minority class distribution.

TABLE IV. CREDIT CARD DETECTION RESULTS: F1 SCORE MEASURE

Algorithm	Original	*Smote*	*Adasyn*	*B-Smote*	*cGAN*	*K-CGAN*
XGBoost	0.824561	0.999613	0.999726	0.999760	0.882100	0.999817
Random Forest	0.824561	0.999745	0.999673	0.999431	0.888700	0.999810
Nearest neighbor	0.725664	0.990713	0.987067	0.996691	0.853900	0.999733
MLP	0.768559	0.998146	0.998131	0.998634	0.880700	0.999747
Logistic regression	0.673077	0.945500	0.884381	0.985338	0.750000	0.999613

To characterize our approach as successful, the following criteria must be satisfied:

H1. Utilizing K-CGAN to improve imbalanced datasets will result in better performance of algorithms on those datasets.

H2. These were evaluated by combining the original and artificial sets with the four classification algorithms, including Xgboost, LR, RF, XGBoost, and MLP.

With the original dataset, we trained our K-CGAN model to produce a synthetic dataset. We then tested it with various classification algorithms and saw an improvement in the f1 score when introduced fraud transactions through the K-CGAN. The experiment process we followed is detailed in the flowchart below (Figure 6).

IV. RESULTS

For credit card fraud, we show the classification results obtained after 100 epochs for each oversampling technique and classification algorithm.

We divided the data into testing and training sets. The training set included 80% of each class's samples, while the testing set contained the remaining 20%.

We report the F1-score. The best results for each metric are in bold As can be seen from the table IV, our method improves the performance of all the classification algorithms. This demonstrates the effectiveness of our method in terms of imbalanced data classification

V. CONCLUSION

We created a new method, K-CGAN, for generating synthetic data with CGANs that uses KL divergence in the Generator loss function. We compared our approach against well-known oversampling techniques (SMOTE, B-SMOTE and ADASYN) as well as other adversarial network architectures used to generate new data (cGANs).

We conducted a study to assess how well K-CGAN can generate high-quality synthetic data. We compared the performance of five machine learning classification algorithms that were combined with our method, using a publicly available credit card fraud dataset. The results in Table IV

show that K-CGAN outperformed all other oversampling methods, achieving the highest overall rank. In addition to SMOTE, ADASYN, B-SMOTE and cGAN, our method had the best performance. In future we are planning to use K-CGAN for detecting other types of anomaly not just in credit card dataset but also in time series and computer network traffic dataset.

REFERENCES

[1] Brennan, P., 2012. A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection. Institute of technology Blanchardstown Dublin, Ireland.

[2] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S. and Bontempi, G., 2014. Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications, 41(10), pp.4915-4928.

[3] Thabtah, F., Hammoud, S., Kamalov, F. and Gonsalves, A., 2020. Data imbalance in classification: Experimental evaluation. Information Sciences, 513, pp.429-441.

[4] Zhu, H., Liu, G., Zhou, M., Xie, Y., Abusorrah, A. and Kang, Q., 2020. Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection. Neurocomputing, 407, pp.50-62.

[5] Le, T., Vo, M.T., Vo, B., Lee, M.Y. and Baik, S.W., 2019. A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction. Complexity, 2019.

[6] Ngwenduna, K.S. and Mbuvha, R., 2021. Alleviating class imbalance in actuarial applications using generative adversarial networks. Risks, 9(3), p.49.

[7] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. and Bharath, A.A., 2018. Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), pp.53-65

[8] Gui, J., Sun, Z., Wen, Y., Tao, D. and Ye, J., 2021. A review on generative adversarial networks: Algorithms, theory, and applications. IEEE Transactions on Knowledge and Data Engineering.

[9] Pandey, A., Bhatt, D. and Bhowmik, T., 2020. Limitations and Applicability of GANs in Banking Domain. In ADGN@ ECAI.

[10] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

[11] Assefa, S.A., Dervovic, D., Mahfouz, M., Tillman, R.E., Reddy, P. and Veloso, M., 2020, October. Generating synthetic data in finance: opportunities, challenges and pitfalls. In Proceedings of the First ACM International Conference on AI in Finance (pp. 1-8).

[12] Charitou, C., Dragicevic, S. and Garcez, A.D.A., 2021. Synthetic Data Generation for Fraud Detection using GANs. arXiv preprint arXiv:2109.12546.

[13] Eckerli, F. and Osterrieder, J., 2021. Generative Adversarial Networks in finance: an overview. arXiv preprint arXiv:2106.06364.

[14] Ferreira, F., Lourenço, N., Cabral, B. and Fernandes, J.P., 2021. When Two are Better Than One: Synthesizing Heavily Unbalanced Data. IEEE Access, 9, pp.150459-150469.

[15] Koshiyama, A., Firoozye, N. and Treleaven, P., 2019. Generative adversarial networks for financial trading strategies fine-tuning and combination. arXiv preprint arXiv:1901.01751.

[16] Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[17] Rundo, F., Trenta, F., di Stallo, A.L. and Battiato, S., 2019. Machine learning for quantitative finance applications: A survey. Applied Sciences, 9(24), p.5574.

[18] Takahashi, S., Chen, Y. and Tanaka-Ishii, K., 2019. Modeling financial time-series with generative adversarial networks. Physica A: Statistical Mechanics and its Applications, 527, p.121261.

[19] Wiese, M., Knobloch, R., Korn, R. and Kretschmer, P., 2020. Quant GANs: deep generation of financial time series. Quantitative Finance, 20(9), pp.1419-1440.

[20] Zhang, Z., Yang, L., Chen, L., Liu, Q., Meng, Y., Wang, P. and Li, M., 2020. A generative adversarial network–based method for generating negative financial samples. International Journal of Distributed Sensor Networks, 16(2), p.1550147720907053.

[21] Ibtissam Benchaji, Samira Douzi, and Bouabid El Ouahidi, "Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks," Journal of Advances in Information Technology, Vol. 12, No. 2, pp. 113-118, May 2021. doi: 10.12720/jait.12.2.113-118.

[22] Maria R. Lepoivre, Chloé O. Avanzini, Guillaume Bignon, Loïc Legendre, and Aristide K. Piwele, "Credit Card Fraud Detection with Unsupervised Algorithms," Vol. 7, No. 1, pp. 34-38, February, 2016. doi: 10.12720/jait.7.1.34-38.