GANs for Fraud Detection

Advanced Topics on Machine Learning

Cátia Teixeira, *Student*
MSc Data Science and Engineering
Faculty of Engineering,UP
up200808037@fe.up.pt

Henrique Ribeiro, *Student*
MSc Data Science and Engineering
Faculty of Engineering,UP
up202204383@fe.up.pt

Ian Karkles, *Student*
MSc Data Science and Engineering
Faculty of Engineering,UP
up202200596@fe.up.pt

Vitor Pereira, *Student*
MSc Data Science and Engineering
Faculty of Engineering,UP
up202210497@fe.up.pt

***Abstract*—This work explores the topic of anomaly detection for credit card fraud. The article provides the background and characterization of the problem, an overview of related work and techniques for this task as well as a description of the methodology and implementation details of a total of fourteen models. The aim is to explore the task of anomaly detection using both Data Augmentation and no Data Augmentation, as well as different Generative Adversarial Networks (GANs) architectures and different Classification approaches. Three different GANs were used for Data Augmentation: Vanilla GAN (VGAN), Conditional Tabular GAN (CTGAN) and Wasserstein GAN (WGAN). Three different classification approaches were applied for each Data Augmentation approach: Classic Machine Learning, Deep Neural Network (DNN) and Long-Short Term Memory (LSTM). A WGAN was also used for Anomaly Detection. Additionally, an innovative Multi-Critic Classifier approach was implemented based on the probability aggregation of the previously obtained models (Baseline-DNN, VGAN-DNN, CTGAN-DNN and WGAN-DNN). The best results were obtained by the Multi-Critic Classifier with the better compromise of higher rates of True Positives (TP) and True Negatives (TN) and lower percentages of False Positives (FP) and False Negatives (FN): 0.91, 0.95, 0.05 and 0.09 respectively.**

***Index Terms*—Generative Adversarial Networks (GAN), Conditional Tabular Generative Networks (CTGAN), Wasserstein Generative Adversarial Networks (WGAN), Credit Fraud Detection, Deep Neural Networks (DNN), Long-Short Term Memory (LSTM)**

I. INTRODUCTION

GENERATIVE ADVERSARIAL NETWORKS (GANs) are a technique that has been emerging for both supervised and unsupervised learning [1]. First proposed in 2014 [2], they can be described as training a pair of networks in competition with each other. They have gained attention due to its simplicity and effectiveness. In fact, in a short time span, considerable progress was made to the initial application of GANs [3]. Generative models, such as GANs, are able to generate new data instances. Given an observable variable X and a target variable Y, these models provide a statistical model of the joint probability distribution on $P(X|Y)$. GANs can be used in many fields such as data augmentation, computer vision and images, tabluar data, anomaly detection, among others.

Credit card fraud is becoming a serious and growing problem as a result of the emergence of innovative technologies and communication methods, such as contactless payment. Thus, the evolution and expansion of credit card use has led to the emergence of multiple forms of fraud [4], [5].

A. Background

Generative adversarial networks (GANs) are a developed techinique to learn both in semi-supervised and unsupervised data [6]. Hence, this type of generative models are the most promising architectures in what concerns unsupervised learning [6] [7]. These GANs have a variety of advantages over the existing models like Boltzman machine [8] or Autoencoders [9]. Since [8] and [9] techniques are based on Markov chains, GANs were designed to avoid this property due to their associated computational constraint. This was a serious issue solved by the newest techniques. Nevertheless, there was also a lack of generalisation in these Markov chain-based techniques. To overcome this, in 2014, Goodfellow et al, proposed a new architecture named GANs. Based on the game theory, there are two different networks. One works as the generator, and the other one is the discriminator [10]. GANs have been sucessfully implemented for solving several tasks across multiple domains and areas of expertise, such as image generation through descriptions [11] or from low-resolution images getting new ones with high-resolution [12], with regard to data augmentation. As far as predictions are concerned, successful implementations have also been made, such as predicting which drug can treat a potential disease [13] or for object detection [14].

B. Problem Characterization

Given the progress of the digital world, credit card has been a much more efficient and convinent payment mode [15]. However, in parallel with this growth, new credit card fraud techniques have emerged, making the task of detecting these techniques much more difficult, thus leading to huge monetary for individuals or banks [16].

In recent years, machine learning techniques, particularly deep learning, have played a key role in detecting credit card fraud [17]. Although it has become an interesting area for both companies and academia, there are still some challenges to overcome. Imbalanced datasets are the major problem when training a fraud detection model [18]. Since fraudulent transactions rarely occur, when compared with legal transactions, these datasets display the highest degree of class imbalance problem and it becomes difficult for the classification algorithms to identify the fraudulent transactions[19], [20].

Thus, our approach aims to propose a possible solution to deal with this problem by augment the minority class. This is possible by reproducing new instances similar to the original data. So, augmentation helps to solve the underrepresentation of the minority, while at the same time, helps to avoid overfitting [21].

C. Related work

Several advancements have been made in Anomaly Detection in the past years. Concepts worth noting:

1) Vanilla Generative Adversarial Network (VGAN): The Vanilla GAN is a type of neural network used for generative modelling consisting of two components: the generator and the discriminator. Whilst the generator generates new data instances resembling real data distribution, the discriminator is a binary classifier that tries to distinguish between the real data and the data generated by the discriminator. D and G play the following two-player minimax game with value function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \tag{1}$$

The key in the VGAN lies in the loss functions. The loss function for the Discriminator revolves around the discrepancy between predictions for real and generated data, while for the Generator, it focuses on the negative mean of the Discriminator's evaluation of the generated data - this, and the adversarial setup, encourages the Generator to produce increasingly realistic data that the Discriminator will misclassify as real [2].

2) Condicional Tabular Generative Adversarial Network (CTGAN): Shortly after the publication of the original GAN, several variations followed. One particular useful variant is the Conditional GAN [22]. GANS are extended to a conditional model if both the generator and discriminator are conditioned on some extra information y. y could be any kind of auxiliary information, such as class labels or data from other modalities. Although the first application of this concept was applied to Computer Vision, soon enough it was adapted for the generation of tabular data. To the date, existing statistical and deep neural network models failed to properly model this type of data and this development proved to outperform previous methods for the task [23].

3) Wasserstein Generative Adversarial Network (WGAN): WGAN appears as an alternative to traditional GAN training. The model shows that it can improve the stability of learning such as getting rid of problems like mode collapse [24].

The innovation stems from employing the Wasserstein distance loss, which quantifies the dissimilarity between real and generated data distributions by focusing on the transportation cost of moving one distribution to another, ensuring smoother and more stable gradient updates during GAN training [25].

Whereas traditional Vanilla GANs normally measure how distinguishable the two distributions are, the Wasserstein distance measures the "cost" of transforming one distribution into the other [26].

The Wasserstein distance, also known as the Earth-Mover (EM) distance is presented in the following Eq.2.

$$W(P_r, P_g) = \inf_{\gamma\in\Pi(P_r,P_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{2}$$

where $\Pi(P_r, P_g)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively P_r and P_g. $\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions P_r into the distribution P_g . The Wasserstein-1 distance then is the "cost" of the optimal transport plan [24].

4) WGAN for Anomaly Detection: GANs have been investigated to perform unsupervised anomaly detection where the goal is to identify "deviant" samples from the original dataset. The procedure is to train the GAN on normal data and define a score function able to separate normal test data from abnormal data [27].

5) Aggregated Probability Forecasting Transformation: The standard practice when combining probability estimates from many experts to form a single overall estimate is to take a (perhaps weighted) average of the individual estimates.

Several investigators ([28], [29], [30], [31]) have found that averaged estimates are typically conservative and can be improved by transforming the average so that probabilities become more extreme, closer to 0 or 1:

$$t(p_i) = \frac{p_i^a}{p_i^a + (1-p_i)^a} \tag{3}$$

This approach is normally applied to forecasting but given it is a techinique that aggregates probabilities from different sources, it can be interesting to explore to other purposes.

II. METHODOLOGY

This project focuses in anomaly detection in the domain of financial fraud. Due to class imbalance in problems of this area, we propose the implementation of Machine Learning and Deep Learning techniques, such as data augmentation, to mitigate this issue.

A. Formulation of the problem

Whereas there are several examples over the literature that have been implemented GANs for fraud detection [32] [33], our approach was designed to perform a different task using the same architectures (GANs). The applied methodology was mainly focused on performing data augmentation of the minority class using GANs, thus solving the problem of class imbalance commonly present in this domain, followed by the application of predictive algorithms for classification of fraud using a fusion of the generated data and the real data. A novel approach applied was to combine the different classifiers into a Multi-Critic Classifier. A GAN was also used as a classification model for experimental purposes.

B. Implementation details

The dataset for this study was obtained through Kaggle[1]. It contains transactions made by credit cards in September 2013

[1]https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

by European cardholders. The dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

The data contains only numerical input variables which are the result of a PCA transformation and no further details are provided due to confidentiality. Features V1 to V28 are the principal components obtained with PCA. 'Time' and 'Amount' were not transformed. 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.

To implement the procedures, Python programming language was used in Visual Studio Code integrated development environment as well as Google Colaboratory.

The code is divided in the following modules:

- **project_part_A.ipynb**: VGAN training and data augmentation as well as its classification using CatBoost, DNN and LSTM models.
- **project_part_B.ipynb**: Baseline classification (without data augmentation) using CatBoost, DNN and LSTM models. WGAN training and data augmentation and its classification using CatBoost, DNN and LSTM models. Anomaly WGAN implementation. Multi-Critic approach using Baseline-DNN, VGAN-DNN, CTGAN-DNN and WGAN-DNN trained models for Classification.

Several Python libraries are used such as *pycaret*, *ctgan*, *sklearn*, *numpy*, *pytorch* and *pandas*. Experiments were performed using cuda and cuDNN libraries on a NVIDIA GeForce RTX 3080 Ti and Google Colab T4 GPU.

C. Plan

Several approaches were taken for solving the task of this work. A summary of the methodology applied can be found in Fig. 1. The original dataset was handled using both data augmentation and no data augmentation. For data augmentation three different GAN architectures were used: Vanilla GAN (VGAN), Conditional Tabular GAN (CTGAN) and Wasserstein GAN (WGAN). The dataset without data augmentation was also considered for the sake of comparison of results. Afterwards, theree different classification approaches were taken: classical machine learning (ML) models, Deep Neural Network (DNN) model and Long Short Term Memory (LSTM) network model. In parallel a WGAN for anomaly detection was also applied for fraud detection in the original non augmented dataset. Additionally, an aggregated classifer was also used using several previously trained models into a Multi-Critic Classifier.

D. Implemented algorithms

The first step of pre-processing was the dataset split into train and test sets. The train test consists in the first 40 hours of data and the test corresponds to the final 8 hours. The test is hold-out and only used for the testing of each classication

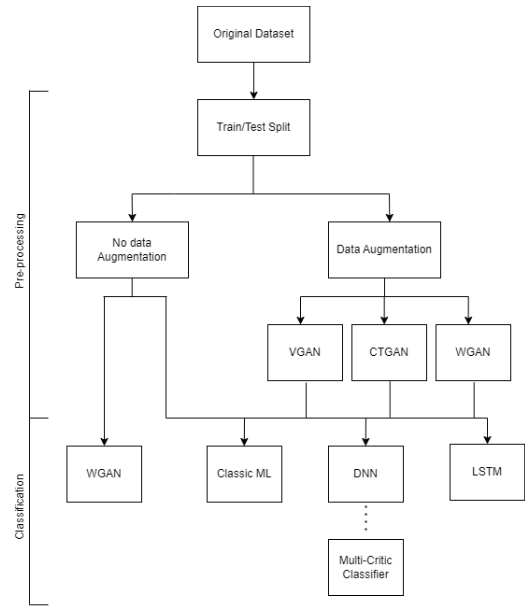

Fig. 1: Implemented work plan

approach. A baseline is considered consisting of the training set without data augmentation that is fed into the classification models for the sake of comparison.

1) Data Augmentation: The minority class is augmented to roughly the size of the majority class using the different approaches described below.

a) VGAN: The Generator begins with an input layer that receives a latent vector, followed by a series of five fully connected layers with dimensions progressively changing (128 $\rightarrow$ 1024 $\rightarrow$ 512 $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ 30) Each of these layers, except the final one, is followed by a ReLU activation function and a Drop Out of 0.2. The GAN wass trained for 1000 epochs.

The Discriminator takes input data, progresses through a series of dense layers with ReLU activations to eventually yield a single value (30 $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ 1). During training, both the Generator and Discriminator undergo optimization through Adam optimizers, utilizing a relatively low learning rate of 0.00005.

For this GAN, the loss function is calculated based out of the Eq.1.

b) CTGAN: For the implementation of this technique, *ctgan* library was used with 300 epochs of training.

c) WGAN: The Generator begins with an input layer that receives a latent vector, followed by a series of five fully connected layers with dimensions progressively changing (128 $\rightarrow$ 512 $\rightarrow$ 512 $\rightarrow$ 1024 $\rightarrow$ 1024 $\rightarrow$ 30). Each of these layers, except the final one, is followed by a ReLU activation function. The Discriminator takes input data, progresses through a series of dense layers with LeakyReLU activations, employing batch normalization to eventually yield a single value (30 $\rightarrow$ 1024 $\rightarrow$ 1024 $\rightarrow$ 512 $\rightarrow$ 256 $\rightarrow$ 1). During training, both the Generator and Discriminator undergo optimization through RMSprop optimizers, utilizing a relatively low learning rate of 0.00005. The loss is calculated according to Eq. 2. The batch size for processing data during each iteration is set at 64. The training process involves an alternating update pattern, where the Discriminator is updated more frequently (5 times) than the Generator for each iteration. Additionally, the Discriminator's weights are clipped within a range (-0.01 to 0.01) to satisfy the Lipschitz continuity condition, a crucial element in WGANs.

2) Classification: The approaches are described below.

a) Classic ML models: Within the Classic Classification Models in the realm of ML, the approach utilized was to compare multiple machine learning models in its vanilla parameters and compare it's performance. For this purpose *pycaret* library was ued and CatBoost model was considered since it yielded better results in the majority of the approaches with the exception o VGAN where RF was the best model.

b) DNN: The implemented DNN is a Feedforward Neural Network consisting of an input layer, three hidden layers ($1024 \rightarrow 512 \rightarrow 256$ neurons) and an output layer (Sigmoid activation). Batch Normalization and ReLU activations are applied after each hidden layer. The training is performed using Adam optimizer, Binary Cross-Entropy loss for 100 epochs with batch size of 64. The learning rate is 0.0001.

c) LSTM: A bidirectional LSTM is applied with 5 layers and 256 hidden units, capable of processing time-series data. The output is passed through a linear layer with a ReLU activation, followed by a final linear layer with a sigmoid activation, reducing it to a single value representing the probability of fraud. The model is trained using a batch size of 128 and a learning rate of 0.0001, Binary Cross-Entropy loss function and Adam optimizer.

d) WGAN: The same architecture and parameters are used as in described in *1.c)* of this chapter. An additional second Discriminator is used, with the same configuration except for a sigmoid output used for fraud classification. The WGAN (including the additional Discriminator) is trained for 100 epochs with the majority class of the training set.

e) Multi-Critic Classifier: The DNN models that result from the training with the different Data Augmentation techniques (Baseline, VGAN, CTGAN and WGAN) are combined to form a "Multi-Critic" Classifier. The aggregated probabilities are calculated according to Eq.3. For Baseline $a = 0.40$ is applied, and for the remaining classifiers $a = 3.50$ is considered since the aim is to maximize True Positives and True Negatives.

III. Results and Discussion

Balanced Accuracy metric is used in both binary and multi-class classification. Its the arithmetic mean of sensitivity and specificity, its use case is when dealing with imbalanced data, i.e. when one of the target classes appears a lot more than the other. Sensitivity refers to $TP/(TP + FN)$ and Specificity refers to $TN/(TN + FP)$. This metric was chosen since our main goal was to maximize TP and TN and minimize FP and FN. The summary results of the experiments can be found in Table I.

Data Augmentation approach was taken with different GANs in order to improve TP in comparison with baseline models that show a poor performance for anomaly detection. In both VGAN, CTGAN and WGAN data augmented models TP value increased but also the unwanted effect of higher FP occurs. This is a commonly known drawback in data augmentation. The average BA scores across classifier techniques for each GAN show the highest average is achieved with WGAN (0.71) followed by CTGAN (0.66) and VGAN (0.57).

TABLE I: Summary results of applied approaches

	TP	TN	FP	FN	BA
Baseline-CatBoost	0.71	1.00	0.00	0.29	0.69
Baseline-DNN	0.88	0.97	0.03	0.12	0.85
Baseline-LSTM	0.66	1.00	0.00	0.34	0.64
VGAN-RF	0.79	1.0	0.0	0.21	0.77
VGAN-DNN	0.99	0.16	0.64	0.01	0.17
VGAN-LSTM	0.74	1.00	0.00	0.26	0.76
CTGAN-CatBoost	0.77	1.00	0.00	0.23	0.75
CTGAN-DNN	0.94	0.49	0.51	0.06	0.48
CTGAN-LSTM	0.78	1.00	0.00	0.22	0.76
WGAN-CatBoost	0.68	1.00	0.00	0.32	0.65
WGAN-DNN	0.86	0.82	0.18	0.14	0.71
WGAN-LSTM	0.68	1.00	0.00	0.32	0.65
Anomaly WGAN	0.84	0.99	0.01	0.16	0.83
Multi-Critic Classifier	0.91	0.95	0.05	0.09	0.86

Both DNN and LSTM show an improvement when compared to with CatBoost (gradient boosting on decision trees).

In order to access if the 'time' variable could be relevant for the detection of frauds, an LSTM model was applied to all approaches. From the Table above we can infer the use of a LSTM network does not show a significant improvement in relation to their respective DNN models. This could be due to the fact that 'time' variable is a relative measure of time and does not represent absolute time. Additionally, the time range of the data is very short and probably not long enough for the network to detect temporal patterns.

In order to decrease the unwanted high FP values discussed previously, an Anomaly WGAN (without data augmentation) was implemented. The results show an improvement in relation to the other models. However its TP value is not very suited for an anomaly detection model.

The final approach taken was to combine all DNN models that result from training with augmented data from the different GANs against the Baseline-DNN. The results show the Multi-Critic Classifier achieves the highest BA value of all experiments achieving the better combination of high values for TP and TN and low values of FP and FN.

IV. Conclusions

In this work several approaches and implementations using GANs were taken to perform anomaly detection of credit card frauds.

A total of 14 models were applied and the best performing model was found to be the Multi-Critic Classifier model which aggregates 4 different models, with a BA score of 0.86.

Hence, we can conclude GANs are a powerful and useful methodology for the mentioned task.

V. Acknowledgements

The authors would like to thank the Machine Learning Group of Université Libre de Bruxelles (ULB)[2] for making the dataset available publicly, and also for its collection and anonymization preprocessing in collaboration with Worline[3]company.

An additional appreciation to all the referrenced authors.

References

[1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[3] F. Eckerli and J. Osterrieder, "Generative adversarial networks in finance: an overview," *arXiv preprint arXiv:2106.06364*, 2021.

[4] K. Chaudhary, J. Yadav, and B. Mallick, "A review of fraud detection techniques: Credit card," *International Journal of Computer Applications*, vol. 45, no. 1, pp. 39–44, 2012.

[5] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 1, pp. 145–174, 2023.

[6] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.

[7] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of Generative Adversarial Networks (GANs): An Updated Review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 525–552, Mar. 2021.

[8] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, pp. 147–169, Jan. 1985.

[9] . Ekici, *Introduction to Deep Learning - Sandro Skansi*.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," June 2014. arXiv:1406.2661 [cs, stat].

[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," Dec. 2017. arXiv:1701.07875 [cs, stat].

[12] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a GAN to learn how to do image degradation first," July 2018. arXiv:1807.11458 [cs].

[13] "GANs for medical image analysis - ScienceDirect."

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[15] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, pp. 937–953, Nov. 2017.

[16] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, pp. 37–48, Jan. 2008. Conference Name: IEEE Transactions on Dependable and Secure Computing.

[17] E. Strelcenia and S. Prakoonwit, "A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection," *Machine Learning and Knowledge Extraction*, vol. 5, pp. 304–329, Mar. 2023. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[18] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "GMM-based Undersampling and Its Application for Credit Card Fraud Detection," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2019. ISSN: 2161-4407.

[19] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[20] Z. Cai, X. Wang, M. Zhou, J. Xu, and L. Jing, "Supervised class distribution learning for gans-based imbalanced classification," *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 41–50, 2019.

[21] A. Kuppa, L. Aouad, and N.-A. Le-Khac, "Towards Improving Privacy of Synthetic DataSet," June 2021.

[22] S. O. Mehdi Mirza, "Conditional generative adversarial nets," *arXiv*, 2014.

[23] A. C.-I. K. V. Lei Xu, Maria Skoularidou, "Modeling tabular data using conditional gans," *arXiv*, p. 3, 2019.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 214–223, PMLR, 06–11 Aug 2017.

[25] M. Pfenninger, D. N. Bigler, S. Rikli, and J. Osterrieder, "Wasserstein gan: Deep generation applied on financial time series," *International Journal of Computer Vision*, July 2021.

[26] Q. Wang, X. Zhou, C. Wang, Z. Liu, J. Huang, Y. Zhou, C. Li, H. Zhuang, and J.-Z. Cheng, "Wgan-based synthetic minority oversampling technique: Improving semantic fine-grained classification for lung nodules in ct images," *IEEE Access*, vol. 7, pp. 18450–18463, 2019.

[27] V. F. Ilyass Haloui, Jayant Sen Gupta, "Anomaly detection with wasserstein gan," *arXiv*, 2018.

[28] J. Baron, B. A. Mellers, P. E. Tetlock, E. Stone, and L. H. Ungar, "Two reasons to make aggregated probability forecasts more extreme," *Decision Analysis*, vol. 11, no. 2, pp. 133–145, 2014.

[29] D. Ariely, W. Tung Au, R. H. Bender, D. V. Budescu, C. B. Dietz, H. Gu, T. S. Wallsten, and G. Zauberman, "The effects of averaging subjective probability estimates between and within judges.," *Journal of Experimental Psychology: Applied*, vol. 6, no. 2, p. 130, 2000.

[30] I. Erev, T. S. Wallsten, and D. V. Budescu, "Simultaneous over-and underconfidence: The role of error in judgment processes.," *Psychological review*, vol. 101, no. 3, p. 519, 1994.

[31] U. S. Karmarkar, "Subjectively weighted utility: A descriptive extension of the expected utility model," *Organizational behavior and human performance*, vol. 21, no. 1, pp. 61–72, 1978.

[32] C. Charitou, A. d. Garcez, and S. Dragicevic, "Semi-supervised GANs for Fraud Detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2020. ISSN: 2161-4407.

[33] B. Alshawi, "Utilizing GANs for Credit Card Fraud Detection: A Comparison of Supervised Learning Algorithms," *Engineering, Technology & Applied Science Research*, vol. 13, pp. 12264–12270, Dec. 2023. Number: 6.

[2]https://mlg.ulb.ac.be

[3]https://worldline.com/