Comparative Analysis of Machine Learning Algorithms using GANs through Credit Card Fraud Detection

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— **In more recent years, credit card fraudulent transactions became a major problem. These fraudulent transactions not only incur huge monetary losses to commercial banks and financial institutions, but also stress and trouble to the lives of customers. Furthermore, with the passage of time this issue is increasing and the monetary loss is expected to increase significantly. However, efficient fraud detecting and prevention measures can trim down the monetary loss due to financial fraud activities. Credit card fraud detection has gained much interest from academia. Generative Adversarial Networks (GANs) are an effective class of generative approaches that has been able to generate synthetic data to assist with the classification of credit card fraudulent activities. In this research study we're going to compare architectures of various GAN models which demonstrate the evolution of these models. It was observed that GANs have received much attention from researchers and also attained promising results in the field of credit card fraud detection.**

Keywords—component, GANs, hyperparameter setting, Imbalanced data, fraud detection

I. Introduction

Credit card fraud is defined as use of someone else's credit card to steal money or property. Credit card fraud is linked with identity theft and this fraud occur when someone use forged card to buy goods and services. The most occurring credit card frauds are application fraud, shoplifting/ stolen fraud, account takeover fraud, and card not present fraud.

Several deep learning models were introduced by researchers to deal with credit card fraud. However, it is imperative to mention that there still may have deficiencies in these proposed models, specifically supervised algorithms as they need balanced datasets of both legal and illegal credit card transactions. In credit card fraudulent instances, the number of legitimate instances is much higher than the illegal transactions. This difference between the ratio of legitimate and fraudulent transactions create the issue of imbalance classification, in which one class is very smaller than the other class. The distribution of fraudulent and non-fraudulent cases is highly skewed. Since, the distribution ratio of different classes in the dataset play a vital role in model precision and accuracy, pre-processing of the data is important. Machine learning algorithms, such as GANs generate synthetic credit card datasets in order to improve the statistical dispersion between both the fraudulent and non-fraudulent transactions. As a result, financial institutions can easily understand the state of distribution of the data. It is noteworthy to mention that these models do not work efficiently to detect fraudulent transactions as the difference between the ratios of legal and illegal transactions is very high. However, many researchers have introduced Generative Adversarial Network (GAN) based frameworks for detecting fraud to address the above issues.

The Generative Adversarial Networks (GANs) generate synthetic data to support the classification of credit card illegitimate instances. This neural based network is based on the concept of game theory. The two players are a Discriminative D model and a Generative G model. Generally, both the G and the D are Multi-layer Neural Networks (MNNs), where the role of G is to learn the distribution of instances. On the other hand, the role of the D model is the estimation of the probability that an instance occurs from the original generated data [1].

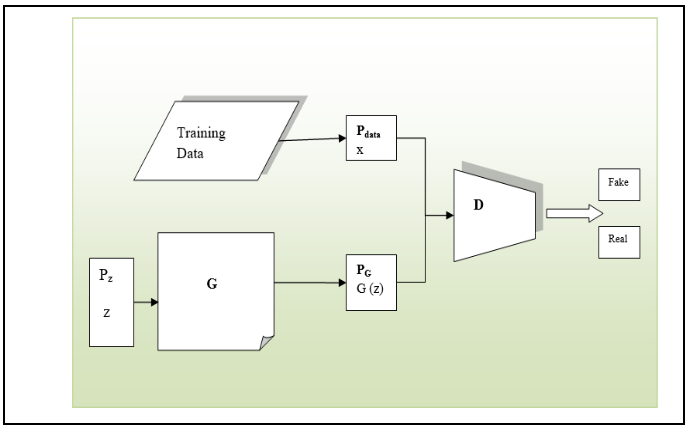

Fig. 1. GAN framework

The figure 1 shows the basic framework of conventional GAN approach. In figure 1, the noise z is randomly generated, while G(z) indicates how G attempts to learn a distribution PG from the distribution of noise P_z and makes P_G closer to the distribution of real-world data, which is denoted by P_{data}. On one hand, the Discriminator attempts to spot whether the sample is fake or real. On the other hand, input is needed to adjust both discriminator and generator till the stage where the discriminator fails to discriminate between the real-world data and the generated data while training. As a consequence, we can attain the optimal point where P_{data} is equal to P_G.

979-8-3503-9801-4/22/$31.00 ©2022 IEEE

Researchers argue that GANs are a more fitting and effective technique for handling the imbalanced class issue compared to other machine learning approaches. In addition, it is highly robust towards overlapping and overfitting due to its ability to understand hidden structures of data and its flexibility.

Fraud detection mechanism is crucial in various important and sensitive fields, such as financial institutions. Hence, fraud detection is a topic that receives much attention from researchers and policymakers. Several GAN based variants have been introduced in recent years to address this issue. Some of the most prominent methods are SDG GANs, LS GANs, NS GANs, and WS GANs, which are briefly explained in this paper.

The financial sector is among the leading segment influenced by the recent advancements in Artificial Intelligence. More accurately, machine learning algorithms have been introduced to detect monetary frauds in financial institutions. Machine learning algorithms can be categorized into classification and resampling algorithms. The widely employed classification machine learning techniques in the credit card fraud detection domain are Naïve Bayes, Random forest, K-nearest neighbour, Support Vector Machine (SVM), decision tree and Artificial Neural Networks. These methods can be employed as computational effective tools for detecting credit card based fraud. On the other hand, the most widely used non-GAN based resampling techniques are SMOTE, ADASYN, Borderline-SMOTE, under-sampling, over-sampling and Borderline SMOTE SVM.

II. Synthetic Data Generation GAN

The discriminator and generator of the SDG-GAN both are convolutional networks with an MLP architecture in the SDG-GAN framework. A typical GAN generator seeks to produce false data that closely resembles the genuine distribution. A normal GAN's discriminator determines if a generator's input is genuine or not. It is essential to train the GAN to assess the scattering of data before creating new instances of the minority class. After the training phase has been finished, the generator's capabilities can be used to produce fresh training sets.

In addition, SDG GAN framework is based on conditional GAN, where the G is a feed-forward Neural Network that attempts to learn the actual data distribution. This novel technique employed a feature matching technique to train the G. The feature matching technique transforms the cost function for the G to minimize the statistical dissimilarities between the features of the synthetic and real data. This step alters the scope of the network from tricking the opposite to matching features in the actual data.

In the SDG-GAN structure, the discriminator and generator are both MLP-configured convolutional channels. A typical GAN generator seeks to generate fictitious data that closely resembles the true distribution. The discriminator in a standard GAN decides whether or not the input from a generator is genuine. The loss function of SDG GAN may be expressed as

$$\min_G \max_D \left|\left| \underbrace{E_{x\sim pdata} f\left({}^{x}/_{y}\right) - E_{z\sim pz({}^{z}/_{y})} f(G(z)) \left|\right|}_{\text{FM Loss}} \right._2^2 + E_{x\sim pdata}\left[\log\left(D\left({}^{x}/_{y}\right)\right)\right] \quad (1)$$

The binary cross entropy between the true class label, y (0, 1), and the predicted class probability make up the remaining portion of the objective function, where FM is the feature matching loss. The conditional distribution is estimated using the cGAN architecture, where px|y is modified to produce samples from the minority class. The SDG-GAN adapts feature matching loss rather than ordinary loss. In order to reduce the statistical disparities between the characteristics of the actual data and the produced data, feature matching adjusts the objective functions for the generator [2]. As a result, the generative network's focus shifts from deceiving the adversary to matching features in the actual data.

A. SDG GAN Hyper Paramter Settings

With size dimensions set to 50, the noise parameter dispersion was configured as a Gaussian distribution. Both hidden units of the discriminator, as well as the generator, have a dropout ratio of 0.2. The batch size was adjusted to 64, and the epoch number to 100. In respect of the input signal, rectified linear units were employed for the hidden nodes, with stochastic for the discriminator's output nodes and tanh for the generator's output nodes. It is argued that for the training datasets, the Adam optimizer was employed [3].

TABLE I. Hyper Parameter Setting for SDG GANs

Hyper Parameter Setting	
Parameter	***Value***
Learning Rate	0.0001
Output Optimizer	Adam
Epochs	100
Batch Size	64
Generator Layers	(Noise, 128), (128, 64), (64, *datasize*)
Discriminator Layers	(*datasize*, 128), (128, 64), (64, 32), (32,1)
Activation function	ReLU
Noise Distribution	N (0,1)
Noise	50

III. Least Square GAN

Conventional GANs uses sigmoid cross-entropy loss function for the D, but the adoption of this loss function may face the issue of vanishing gradient while updating the G. Conversely, the LS-GAN uses the least square loss function for the D. This loss function has capability to move the fake instances to the decision boundary. On the basis of this LS-GAN trait, this novel framework has ability to generate samples closer to real data. Furthermore, the learning process of LS-GAN is more stable than conventional GANs.

Furthermore, the issue of the vanishing gradient problem in traditional GANs will arise for data that are on the proper side of the decision function but are nonetheless removed from the actual data due to this loss function. Least-square GANs are used to solve this issue. Assume that the discriminator used by the Least - square GAN is coded using the a-b coding system, in which a and b represent the identifiers for false and actual data, accordingly [4]. The loss function for the "D" is different in the least squares, GAN, a variation of the standard approach. By employing the least square error as the loss, this approach demonstrates that the model trains more steadily and is better equipped to tackle the gradient vanishing problem than the vanilla technique.

$$\min_D V_{LSGAN}(D)= \frac{1}{2}E_{x\sim pdata(x)}[\left(D(x)\text{-}b\right)^2]+ \frac{1}{2}E_{z\sim pz(Z)}[(D\left(G(z)\right)\text{-}a)^2] \tag{2}$$

$$\min_G V_{LSGAN}(G)=\frac{1}{2}E_{z\sim pz(z)}[(D\left(G(z)\right)\text{-}c)^2] \tag{3}$$

where b is the label for real data, a for fake data, and c is used as a label for testing the discriminator with generator samples. i.e. b = 1, a = 0 and c = 1

$$\min_D V_{LSGAN}(D)= \frac{1}{2}E_{x\sim pdata(x)}[\left(D(x)\text{-}1\right)^2]+ \frac{1}{2}E_{z\sim pz(Z)}[(D\left(G(z)\right))^2] \tag{4}$$

$$\min_G V_{LSGAN}(G)=\frac{1}{2}E_{z\sim pz(Z)}[(D\left(G(z)\right)\text{-}1)^2] \tag{5}$$

The discriminator aims to reduce the overall shaped change between predicted and anticipated values to discern between legitimate transactions and malicious ones. Nevertheless, the generator makes an effort to minimize the total square discrepancy between predicted and actual numbers in order to make the generated credit card seem as realistic as feasible.

A. Selection of Parameters

In order to diminish the Pearson χ^2 divergence among pd+pg and 2pg, the following loss functions can be minimized by setting b-c=1 and b-a=2, respectively. Thus when Equation 2 is minimized, the Pearson 2 divergence among pd + pg and 2pg is minimized [4]. Setting a = 1, b = 1, and c = 0, for instance, results in the following objective functions

$$\min_D V_{LSGAN}(D)= \frac{1}{2}E_{x\sim pdata(x)}[\left(D(x)\text{-}1\right)^2]+ \frac{1}{2}E_{z\sim pz(Z)}[(D\left(G(z)\right)+1)^2] \tag{6}$$

$$\min_G V_{LSGAN}(G)=\frac{1}{2}E_{z\sim pz(z)}[(D\left(G(z)\right))^2] \tag{7}$$

Setting c = b is another way to help G create samples that are as accurate as feasible. For instance, it may get the following goal functions by employing the 0-1 binary coding scheme:

$$\min_D V_{LSGAN}(D)= \frac{1}{2}E_{x\sim pdata(x)}[\left(D(x)\text{-}\ 1\right)^2]+ \frac{1}{2}E_{z\sim pz(Z)}[(D\left(G(z)\right))^2] \tag{8}$$

$$\min_G V_{LSGAN}(G)=\frac{1}{2}E_{z\sim pz(z)}[(D\left(G(z)\right)-1)^2] \tag{9}$$

In practice, both models are employed. However, the equation is more common in the real world.

IV. WS GAN

This design modifies the default application's loss function and uses a weight clip to promote successful training. They suggest utilizing the earth mover distance to calculate the loss function rather than the Jensen Shanon divergence. This reserve measure remains constant and visible throughout, measuring how closely the data distributions from the exercise dataset and the produced dataset resemble each other. Because it just presents the distribution of the data that the generator (G) gathers and does not evaluate whether the data is correct or not, the "D" is also known as an opponent network [5]. To confirm that the loads in the generator adhere to Lipschitz restrictions, weight clipping is also employed.

$$L=\max_G E_{Ex\sim pz} D(z)) \tag{10}$$

$$L=\max_D E_{Ex\sim pr} D(x)\text{-}E_{Ex\sim p(z)} D(G)(z))) \tag{11}$$

Where, z stands for the input noise variable, D for the discriminator, G for the generator, and Pz for the Gaussian noise distribution. The real model distribution is represented by the sign pr.

V. Non-saturating GAN

Even though the aforesaid loss function exhibits amazing theoretical findings, it performs badly in real-world applications. The GAN struggles with convergence, maintaining stability throughout training, and providing a variety of samples. Instead of training the aforementioned loss function for G, it is preferable to use better gradients from earlier training. Non-saturating GAN is the kind of GAN that is most frequently utilized as a standard in academic studies and real-world applications (NS-GAN).

$$J^{(G)}(G)= -\ E_{z\sim pz}\ \log D(G(z)) \tag{12}$$

G is a model of a probability distribution with the letter p. (x). A sample of this distribution is obtained by the generator network using a noise vector z sampled from Pz, which equals x = G. (z). Although any distribution with sufficient variability is workable, z typically originates from a uniform or Gaussian distribution. The discriminator D(x) attempts to identify if input value x is genuine or fraudulent by comparing it to training data[6].

A. *NS general Adversarial network system loss*

In order to solve the saturation issue, the generator loss was modified to become the non-saturating GAN Loss. Instead of decreasing the log of the reversed discriminator probability for created credit cards, the generator now maximizes the log of the discriminator probability for created credit cards. Since the Jensen-Shannon divergence-based generator gradient works poorly in practice, a non-saturating generator gradient is often employed in its place

[1].

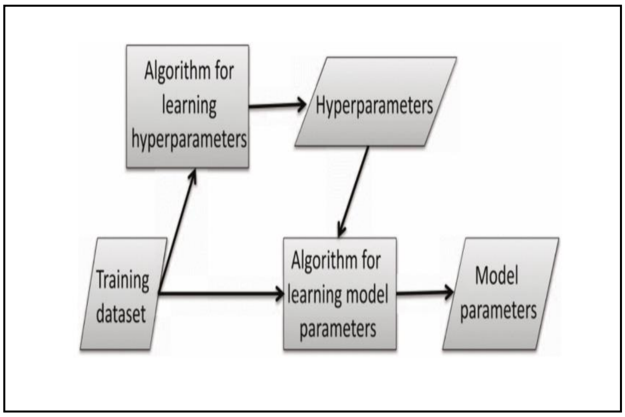
Algorithm for learning hyperparameters
Hyperparameters
Training dataset
Algorithm for learning model parameters
Model parameters

Fig. 2. Process of tuning hyperparameters

$$\mathrm{Ef}(p,q\lambda,d)=-\mathrm{Ez}\left[\mathrm{af}\left(d\left(g\lambda(z)\right)\right)\right]+c \qquad (13)$$

The generator and data dispersion are frequently not well suited initially in training, having data from p being extremely improbable beneath q and vice versa. In areas where d has a big value, the probability mass of p and q is thereby concentrated. Jensen-Shannon effectively absorbs at its highest value in this regime, hence it is not unexpected that this can cause problems with efficiency. Additional f-divergences such as KL and reverse KL do not have the same issues, although a different "non-saturating" generator gradient has nonetheless been proposed for f-GANs [7]. The particular adjustment is to swap out bf for af when computing the generator gradient in both f-GANs and GANs.

VI. GAN Training and Hyperparameter Tuning

A GAN comprises 2 feed-forward neural networks that compete with one another: a Generator G and a Discriminator D, with the former developing new candidates and the latter assessing their merit. Typically, the two networks' deep neural network [8] have many layers linked so that the inputs for units within the layer over each layer are the output of the units within the layer. Layers can optimally be related to degrees of conceptualization or compositional abilities by seeing what is learned at each level as a depiction of the original input. Different levels of abstraction can be achieved by altering the number and size of the layers [9]. The basic purpose of GANs is to improve generative models by forcing them to compete against discriminative models that aim to distinguish between produced instances and actual examples [10]. The discriminator learns to identify instances created by the generator or not by taking input of random noise z, transforming it through a function, and producing examples.

The majority of hyperparameters are what are known as tuning parameters though their values need to be properly tuned because the best values depend on the dataset at hand [11]. Further, they argued that over-fitting is a key idea in parameter tuning; it occurs when parameter values for complicated rules seem to generalize the training data, producing forecasting rules that are overly specific to the training data. These rules perform extremely well on the training data but are likely to perform worse on independent data. By employing a test dataset or cross-validation techniques for tuning, it is possible to partially prevent the selection of such inappropriate parameter values.

A hyperparameter governs the process of learning, and as a result, the values of these parameters have a direct impact on some other training sets such as weights and biases, which in turn affects how well the model works [12]. By tweaking these hyperparameters, any machine learning model's accuracy is frequently increased. Therefore, a deep learning researcher must have a solid understanding of these hyperparameters. Further, the authors argued that hyperparameter affects training data performance in the following ways.

When compared to stochastic gradient descent with the same learning rate, adaptive learning rate optimizers such as Adam, AdaMax, and RMSprop were found to be more successful at learning and achieving greater accuracy more quickly (0.001).

VI.Results and Discussion

In this digital era, it is essential for financial institutions to detect credit card frauds. On the other hand, researchers argue that machine learning techniques can be used to resolve this problem. However, many machine learning algorithms have multiple weaknesses and are unable to detect credit card based fraud effectively. For instance, many supervised techniques are unable to detect fraud patterns during credit card patterns as they need balanced datasets of both legal and illegal transactions. On the other hand, the difference between the ratios of legal and illegal transactions is very high.

Generative Adversarial Networks have achieved significant progress in credit card fraud detection. GANs are capable to address the class imbalance issue as they approximate the distribution of real data and generate synthetic data for the minority class (fraudulent transactions). The findings of this literature review suggest that the GAN framework is more flexible than other generative models, as GANs do not need too many statistical assumptions and other inferences for capturing distribution of data.

Moreover, the findings of this comparative study summarize that the GAN variants discussed in this paper are highly effective to detect credit card fraud instances as weaknesses persisted in the traditional GAN method were addressed in these novel methods, such as the vanishing gradient problem. Furthermore, the Non Saturating GAN was introduced to in order to solve the saturation issue. For the said purpose, the generator loss was modified to become the non-saturating GAN Loss. Thus, novel GAN methods are more effective than other machine learning methods as they are flexible yet effective to detect anomalies in credit card fraud domain.

VII. Conclusion

This paper aims to present an impression of Generative Adversarial Networks in financial institutions. For that said purpose, this work was directed at describing various GANs architectures, recent developments, and their applicability in finance.

To conclude, credit card fraud detection approaches utilize the notion of classification and they require balanced training data streams which should have negative as well as positive transactions. On the contrary, credit card datasets usually have highly skewed datasets with very few fraud cases, making it difficult for fraud detection approaches to training datasets.

However, machine learning algorithms have proved to be capable frameworks for the prediction and prevention of fraud in credit card domain. In addition, an innovative technique, GAN, is capable to tackle the imbalanced class problem.

It is imperative to mention that GANs are in the transactional phase but have made significant progress in the financial markets. Some of the features of GANs, such as generating synthetic data, are gaining solid footholds.

References

[1] "Techopedia.com," 5 September 2018. [Online]. Available: https://www.techopedia.com/definition/33264/hidden-layer-neural-networks. [Accessed 10 December 2022].

[2] C. Charitou, S. Dragicevic and A. Garcez, "Synthetic Data Generation for Fraud Detection using GANs," arXiv, pp. 1-8, 2021.

[3] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv, pp. 1-15, 2014.

[4] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang and S. Paul Smolley, "Least squares generative adversarial networks," IEEE, pp. 2794-2802, 2017.

[5] J. Brownlee, "A gentle introduction to generative adversarial network loss functions," MLM, 2019.

[6] X. Xie and et al., "Generative adversarial network-based credit card fraud detection," in International Conference in Communications, Signal Processing, and Systems, Singapore, 2018.

[7] A. Sethia, R. Patel and P. Raut, "Data augmentation using generative models for credit card fraud detection," IEEE, vol. 4, pp. 1-6, 2018.

[8] I. Goodfellow and et al., "Generative adversarial networks," Communications of the ACM, vol. 63(11), pp. 139-144, 2020.

[9] S. Nowozin, B. Cseke and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," Advances in neural information processing systems, no. 29, pp. 1-13, 2016.

[10] B. Wang and N. Gong, "Stealing hyperparameters in machine learning," IEEE symposium on security and privacy (SP), pp. 36-52, 2018.

[11] C. Hsu, C. Chang and C. Lin, "A practical guide to support vector classification," Data Science, pp. 1-16, 2003.

[12] S. Mittal and S. Tyagi, "Performance evaluation of machine learning algorithms for credit card fraud detection," IEEE, no. 9, pp. 320-324, 2019.

[13] J. Brownlee, "Train-test split for evaluating machine learning algorithms," Machine Learning Mastery, no. 23(7), 2020.

[14] W. Yang, Y. Zhang, K. Ye, L. Li and C. Xu, "Ffd: A federated learning based method for credit card fraud detection," Springer, pp. 18-32, 2019.

[15] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," nature, no. 521(7553), pp. 436-44, 2015.

[16] Y. Bengio, A. Courville and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, no. 35(8), pp. 1798-1828, 2013.

[17] 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, no. 479, pp. 448-455, 2019.

[18] P. Probst, M. Wright and A. Boulesteix, "Hyperparameters and tuning strategies for random forest," Wiley Interdisciplinary Reviews: data mining and knowledge discovery, no. 9(3), pp. 1-19, 2019.

[19] A. Dalli, "Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector," Mathematical Problems in Engineering, vol. 2022, pp. 1-11, 2022.

[20] B. Delyon, Stochastic approximation with decreasing gain: Convergence and asymptotic theory, Université de Rennes, 2000.

[21] P. Murphy Kevin , Machine learning: a probabilistic perspective, Cambridge: The MIT Press, 2012.

[22] V. Lendave, 1 December 2021. [Online]. Available: https://analyticsindiamag.com/what-is-activity-regularization-in-neural-networks/. [Accessed 2 September 2022].

[23] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv, no. 1609.04747, pp. 1-14, 2016.

[24] M. Zeiler, "Adadelta: an adaptive learning rate method," arXiv, vol. 1, no. 1212.5701, pp. 1-6, 2012.

[25] H. McMahan and et al., "Ad click prediction: a view from the trenches," ACM, pp. 1222-1230, 2013.

[26] T. Ma, "Artificial Intelligence technology in a portal frame structure measuring," Research Space, Auckland, 2022.

[27] M. Uzair and N. Jamil, "Effects of hidden layers on the efficiency of neural networks," IEEE, no. 23, pp. 1-6, 2020.

[28] P. Baldi and P. Sadowski, "Understanding dropout," NEURIPS, pp. 2814-2822, 2013.

[29] Google, "The Generator," 18 July 2022. [Online]. Available: https://developers.google.com/machine-learning/gan/generator. [Accessed 6 November 2022].

[30] S. Jang and Y. Son, "Empirical evaluation of activation functions and kernel initializers on deep reinforcement learning," IEEE, pp. 1140-1142, 2019.

[31] S. Kumar , "On weight initialization in deep neural networks," arXiv, vol. 1704.08863v2, pp. 1-9, 2017.

[32] J. Brownlee, "Weight initialization for deep learning neural networks," Machine Learning Mastery, p. 1, 2021.

[33] Musstafa, "medium.com," MLearning.ai, 27 May 2021. [Online]. Available: https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0. [Accessed 6 November 2022].

[34] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," arXiv, vol. 1811.11264v1, pp. 1-12, 2018.