Adversarial Fraud Generation for Improved Detection

Anubha Pandey
Mastercard
Gurgaon, India
Anubha.Pandey@mastercard.com

Alekhya Bhatraju
Mastercard
Gurgaon, India
Alekhya.Bhatraju@mastercard.com

Shiv Markam
Mastercard
Gurgaon, India
Shiv.Markam@mastercard.com

Deepak Bhatt
Mastercard
Gurgaon, India
Deepak.Bhatt@mastercard.com

ABSTRACT

Generative Adversarial Networks (GANs) are known for their ability to learn data distribution and hence exist as a suitable alternative to handle class imbalance through oversampling. However, it still fails to capture the diversity of the minority class owing to their limited representation, for example, frauds in our study. Particularly the fraudulent patterns closer to the class boundary get missed by the model. This paper proposes using GANs to simulate fraud transaction patterns conditioned on genuine transactions, thereby enabling the model to learn a translation function between both spaces. Further to synthesize fraudulent samples from the class boundary, we trained GANs using losses inspired by data poisoning attack literature and discussed their efficacy in improving fraud detection classifier performance. The efficacy of our proposed framework is demonstrated through experimental results on the publicly available European Credit-Card Dataset and CIS Fraud Dataset.

ACM Reference Format:
Anubha Pandey, Alekhya Bhatraju, Shiv Markam, and Deepak Bhatt. 2022. Adversarial Fraud Generation for Improved Detection. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3533271.3561723

1 INTRODUCTION

With ever-increasing innovations in the Fintech space, it has become easier for consumers to make transactions with a seamless experience. While this has added ease to consumer interactions with payment terminals or interfaces, it becomes equally essential to develop innovative solutions to capture evolving fraudulent patterns. The increased adoption of the cashless transaction has made the domain lucrative for fraudsters, and it raises millions of dollars for the global economy each year [1]. Credit card fraud impacts businesses with economic loss and affects the customers' experience and trust. It is the most pressing issue in most industries around the world, and research is conducted to mitigate the causalities created due to credit card fraud [21].

The growing popularity of cashless transactions has produced massive transaction data, enabling the adoption of machine learning algorithms for fraud detection [12, 27, 28]. Recently, deep-learning-based methods [16] present a promising solution to the problem of credit card fraud detection by enabling institutions to make optimal use of the historical transactional data in reducing overall risk. Although the volume of credit card transaction data is huge, the ratio of fraudulent transactions to legitimate transactions is meager. Most of the risk modeling problem deals with class imbalance issues, thereby limiting the ability of the trained classifier to capture rare fraud patterns.

There exist several techniques to handle class imbalance, the most preferred ones are oversampling to increase the proportion of minority samples using SMOTE and its variants [5, 7, 14, 15]. However, SMOTE generates samples using linear interpolation of the existing samples and is limited in its ability to capture the class distribution [2, 10]. Recent advancements explored various architectures of deep learning-based generative models to oversample the minority fraud class data leading to improved fraud detection rate [8, 19, 22, 26]. However, there are several challenges involved in these GAN based oversampling methods:

(1) They fail to capture the real distribution of the fraudulent class. Many generative models often generate fraud samples similar to commonly occurring fraud patterns.
(2) They are prone to detect false positives, i.e., misclassify legitimate transactions as fraud.

The paper describes a data augmentation pipeline to provide better representation for a classifier in the realm of credit fraud detection. Towards the challenges mentioned above, we use a generative adversarial network conditioned on legitimate transactions to generate fraud samples. Traditionally, GAN-based approaches learn to map the randomly generated vector to a vector sampled from fraudulent dataset distribution. This approach often generates minority class samples that are well represented in the true data distribution, limiting the generation of rare samples that fall near the majority class boundary. Thus drawing inspiration from the existing limitations, we propose to train GANs that learn to generate fraud using non-fraud samples. The trained generator samples data equally well from the boundary where non-fraud overlaps with fraud space. To further guide the training of GANs

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICAIF '22, November 2–4, 2022, New York, NY, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561723

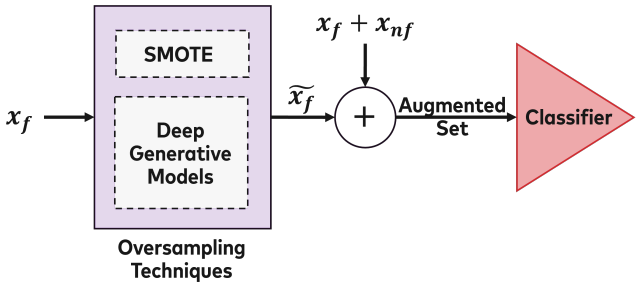


Figure 1: An overview of fraud detection pipeline using oversampling techniques. Existing oversampling based methods for fraud detection learn to generate fraud samples from the real fraud samples and train the classifier model on the augmented dataset with reduced class-imbalance.

with desired learning, we propose the use of specific losses inspired by feature collision and convex polytope attacks, which ensures the model learns to generate samples near the boundary with a non-fraud vector as input. The efficacy of our proposed framework is demonstrated through experimental results on the publicly available European Credit-Card Dataset and CIS Fraud Dataset. On the European Credit Card dataset, there is a 1.3% and 1% absolute lift in Recall and F1-score, respectively, as compared to the state-of-the-art model. And on the CIS Fraud dataset, there is a 0.5% and 0.6% absolute lift in Recall and F1-score, respectively, as compared to the state-of-the-art model.

2 RELATED WORKS

Fraud detection is a very challenging problem. Fraudulent transactions are rare and represent a tiny fraction of all the activities in an organization. Hence, making the fraud detection problem an imbalanced classification problem. Also, a fraud detection system should be efficient enough to minimize false negatives (misclassified frauds) and false positives (legitimate transactions misclassified as fraud). A high number of undetected fraud would cause a huge loss to both businesses and customers. On the other hand, increased cases of stating a legitimate one as fraud would result in lost trust of customers on the financial institutions. Hence, the problem becomes quite challenging.

Machine learning-based algorithms have proved to be promising in credit card fraud detection due to transactional information availability. [20] has done a comparative study of several supervised and unsupervised machine learning algorithms to handle the class imbalance in credit card fraud detection. In Supervised learning, fraud detection is a binary classification problem where the model is trained on historical data labeled as fraudulent or legitimate to identify the fraudulent transactions [1, 9]. Unsupervised ML-based algorithms, on the other hand, do not have access to the labeled dataset and consider the problem as an anomaly detection task assuming frauds to be an anomaly [6, 23]. Unsupervised problems are usually solved using Clustering and compression.

Supervised learning techniques to handle class imbalance that can be broadly classified into two categories: cost-based methods, undersampling and oversamplig techniques [6, 8, 9, 17, 23]. In undersampling, the majority class samples are reduced to balance the dataset. However, this is not a recommended approach as the model might lose some essential information. Oversampling techniques have proved to be effective in handling class imbalance. A commonly used fraud detection pipeline using oversampling techniques is shown in Figure 1. Existing pipeline augment the dataset with synthetic fraud samples to reduce the class imbalance and then train a fraud detection classifier to improve the performance. Earlier methods for oversampling would create multiple copies of existing fraud samples. This doesn't add any new information, thus limiting the generalizing ability of the classifier. Later, researchers introduced SMOTE(Synthetic Minority Oversampling Techniques), and different variants of SMOTE [5, 7, 14]. SMOTE synthesizes new samples of minority class (fraudulent transactions) from the interpolations of existing samples. This limits the ability to generate possibly unseen samples.

Recently, Deep generative networks have received a lot of attention from the research community of credit card fraud detection. Several works [4, 11, 22, 24, 29] have shown the efficacy of GAN for augmenting the dataset with synthetic minority (fraud) samples. However, mode collapse is a common phenomenon that occurs with GANs. Mode collapse happens when GAN generates limited samples and hence fails to capture the whole data distribution. To overcome the issue of mode collapse, researchers [4, 22, 24] have used different architectures of GAN like WGAN[3], Least Square GAN[18], Relaxed WGAN[13] to augment the dataset and have shown an improvement in the classifier's performance. [29] has trained a GAN-based architecture to generate complimentary samples of the legitimate transactions. They train the discriminator to differentiate between legitimate and complimentary samples. At test time, they use the discriminator to identify fraudulent transactions. [26] has used Variational Autoencoders for minority oversampling and have claimed to generate better synthetic samples compared to GANs. [22] has explored several auxiliary networks like Siamese and Classifier and discriminator rejection sampling on top of WGAN architecture for improved fraud generation. [30] uses the siamese neural network structure to solve the problem of sample imbalance in online transaction.

3 METHODOLOGY

3.1 Problem Statement

Consider a transaction dataset with N samples $\{\mathbf{x_i}\}_{i=1}^{N} \in \mathcal{X}$. In this paper, we consider a highly class imbalance setting for the dataset that consists of fraudulent (i.e., the minority class) and genuine (i.e., the majority class) transactions. We represent fraudulent samples as $\mathbf{x_f}$ and genuine or non-fraud samples as $\mathbf{x_{nf}}$. Our goal is to augment the dataset with synthetic samples from the minority class for improved fraud detection.

3.2 Generative Adversarial Network

Generative Adversarial Networks (GANs) have gained popularity for generating realistic-looking images compared to other generative models like variational auto-encoders or restricted Boltzmann machines. GANs are also capable of learning from smaller datasets via semi-supervised learning. Because of these achievements, they

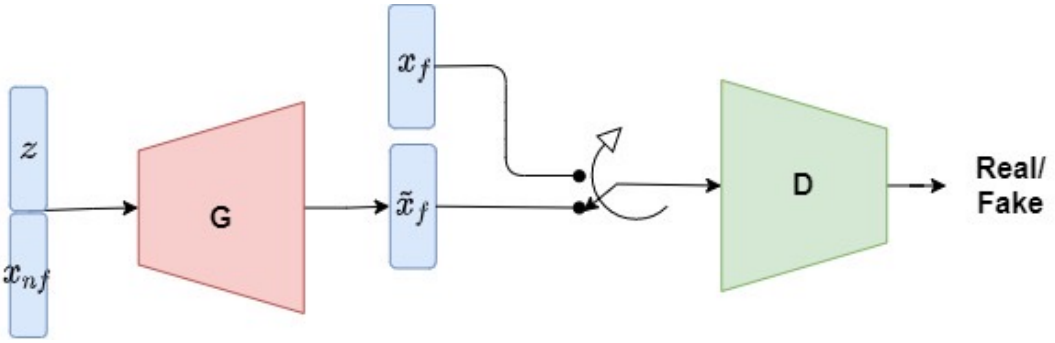


Figure 2: Diagram representing the Adversarial Fraud Generation framework. The generator module G generates synthetic fraud samples conditioned on the real non-fraud samples.

gain a lot of interest in the academic and commercial sectors. More recently, researchers have started using GANs to synthesize financial data. Generative Adversarial Networks consist of two modules, a Generator G and a Discriminator D. Both the modules are multilayer neural networks. The Generator's objective is to generate data from a prior p_z on noise variable z. The Generator learns to synthesize data similar to the original data, i.e., it learns a distribution p_G similar to the original data distribution p_{data}. On the other hand, Discriminator D is a critic that learns to distinguish fake data $G(z)$ from the real data x. Several versions of GANs were introduced over time to improve the training stability and overcome mode collapse. In this paper, we use the most widely used version of GAN called Wasserstein GAN (WGAN). WGAN stabilized the training of traditional GANs by adopting the Wasserstein distance for measuring the distance between two probability distributions. Below are the loss functions to train the Discriminator(D) and Generator(G) modules in WGAN with Gradient Penalty:

$$L_D = \frac{1}{n}\sum_{i=1}^{n}(D_{\theta_D}(\tilde{\mathbf{x}}_{\mathbf{f}_i}) - D_{\theta_D}(\mathbf{x}_{\mathbf{f}_i}) + \lambda(||\nabla_{\tilde{\mathbf{x}}} D_{\theta_D}(\tilde{\mathbf{x}}_{\mathbf{f}_i})||_2 - 1)^2 \quad (1)$$

where, $\tilde{\mathbf{x}}_{\mathbf{f}_i} = t\hat{\mathbf{x}}_{\mathbf{f}_i} + (1-t)\mathbf{x}_{\mathbf{f}_i}$ with $0 \le t \le 1$

.

$$L_G = \frac{1}{n}\sum_{i=1}^{n}(-D_{\theta_D}(G_{\theta_G}(\mathbf{z}_i))) \quad (2)$$

Where θ_G and θ_D represent parameters of the Generator and Discriminator network, respectively.

3.3 Adversarial Fraud Generation

Several works have evaluated the effectiveness of GAN architectures for minority oversampling in credit card fraud detection [4, 22, 24, 29]. However, these methods generate commonly occurring fraud patterns and fail to capture the entire fraud space, particularly from the boundary. We aim to synthesize minority (i.e., fraud) samples from the class boundary for effective data augmentation. To overcome this challenge, we regularize the data generation process with loss functions inspired by the data poisoning attacks literature like Feature Collision [25] and Convex Polytope [31].

Feature collision attack [25] add small adversarial perturbations to the base data such that their feature representation is extremely close to the target data. Regularizing the synthetic data generation using a loss function inspired by the Feature Collision attack will enable the model to generate samples closer to the class boundary. Below is the loss function inspired by the feature collision attack, where $\tilde{\mathbf{x}}_{\mathbf{f}_i}$, $\mathbf{x}_{\mathbf{f}_i}$ and $\mathbf{x}_{\mathbf{nf}_i}$ represents synthetic fraud, real fraud and real non-fraud samples respectively.

$$L_{fc} = \frac{1}{n}\sum_{i=1}^{n} ||\tilde{\mathbf{x}}_{\mathbf{f}_i} - \mathbf{x}_{\mathbf{nf}_i}||_2^2 + \beta(||\tilde{\mathbf{x}}_{\mathbf{f}_i} - \mathbf{x}_{\mathbf{f}_i}||_2^2) \quad (3)$$

Additionally, we regularize the data generation on another type of attack called Convex Polytope [31]. Convex Polytope crafts attacks such that the target data lie in the convex combination of the poison's feature representation. This attack will help generate boundary fraud samples surrounding the non-fraud space. The loss function inspired by convex polytope attack to regularize the data generation process is given below:

$$L_{cp} = \frac{1}{n}\sum_{i=1}^{n} \frac{||\mathbf{x}_{\mathbf{nf}_i} - \sum_{\mathbf{j}=1}^{\mathbf{m}} \mathbf{c_j}(\tilde{\mathbf{x}_{\mathbf{f}_i}}^{\mathbf{j}})||_2^2}{||\mathbf{x}_{\mathbf{f}_i}||_2^2} + \beta(||\tilde{\mathbf{x}}_{\mathbf{f}_i} - \mathbf{x}_{\mathbf{f}_i}||_2^2) \quad (4)$$

where m represents the number of synthetic fraud samples generated corresponding to a non-fraud sample passed as an input

Algorithm 1 Adversarial Fraud Generation Training

Require: η_1, η_2: step-size hyperparameters
Randomly Initialize: θ_D, θ_G
1: **while** True **do**
2: **for** k steps **do**
3: Sample a batch of fraud and non-fraud samples and pair them $(\mathbf{x_f}, \mathbf{x_{nf}})$ randomly
4: Compute L_D given in the Equation 5 and update discriminator's parameters $\theta_D \leftarrow \theta_D - \eta_1 \nabla_{\theta_D} L_D$
5: **end for**
6: Sample a batch of fraud and non-fraud samples and pair them $(\mathbf{x_f}, \mathbf{x_{nf}})$ randomly
7: Compute L_G given in the Equation 7 and update generator's parameters $\theta_G \leftarrow \theta_G - \eta_2 \nabla_{\theta_G} L_G$
8: **end while**

condition to WCGAN. And c_j represents convex combination coefficient. The hyperparameter $\beta(>0)$ trades off the balance between these two terms in the equations 3 and 4.

We leverage the conditional WGAN model (WCGAN) to generate more realistic samples. Although GANs can generate new random plausible examples for a given dataset, there is no way to control the types of data generated other than learning the complex relationship between the latent space input and the generated data. [4] has shown the fraud detection results on conditional GAN where they cluster the fraudulent samples and pass the cluster label as a condition to the GAN. This strategy is ineffective as the data generation depends on the clustering algorithm. In this paper, we provide non-fraud samples $\mathbf{x_{nf}}$ as a condition to the WGAN network and learn a transformation function from non-fraudulent space to fraudulent space. The loss functions to train the Discriminator(D) with parameters θ_D and Generator(G) with parameters θ_G in a mini-batch setting are described below:

$$L_D = \frac{1}{n}\sum_{i=1}^{n}(D_{\theta_\mathrm{D}}(\tilde{\mathbf{x}}_{\mathbf{f_i}}) - D_{\theta_\mathrm{D}}(\mathbf{x}_{\mathbf{f_i}}) + \lambda(||\nabla_{\tilde{\mathbf{x}}} D_{\theta_\mathrm{D}}(\tilde{\mathbf{x}_{\mathbf{f_i}}})||_2 - 1)^2 \tag{5}$$

$$L_G = \frac{1}{n}\sum_{i=1}^{n}(-D_{\theta_\mathrm{D}}(G_{\theta_\mathrm{G}}(\mathbf{z_i}, \mathbf{x_{nf_i}}))) \tag{6}$$

We regularize the Generator module with the adversarial losses mentioned above. The updated loss function for the Generator is defined below:

$$L_G = \frac{1}{n}\sum_{i=1}^{n} -D_{\theta_\mathrm{D}}(G_{\theta_\mathrm{G}}(\mathbf{z_i}, \mathbf{x_{nf_i}})) + \alpha_1 L_{fc} + \alpha_2 L_{cp} \tag{7}$$

where the hyper-parameters $\alpha_1(>0)$ and $\alpha_2(>0)$ are the regularization coefficients.

3.4 Fraud Detection Pipeline

Figure 3 shows an illustration of the fraud detection pipeline from our proposed framework. Below are the steps followed to detect fraudulent transactions at test time:

(1) Generate synthetic fraud using trained Generator conditioned on real non-fraud samples.
(2) Augment the training set with the projected synthetic fraud samples to increase the event rate i.e. the ratio of no. of fraud samples and the total samples.
(3) Train a tree-based classifier on the augmented training set for fraud detection.

4 EXPERIMENTS

4.1 Dataset Description

We have evaluated our proposed method on two publicly available datasets: European Credit-Card dataset [1] and CIS Fraud Detection dataset [2]. The details are provided below:

European Credit-Card Dataset: The dataset consists of two days of transactions by European cardholders from September 2013. The fraudulent transactions are 0.172% of the total 284807 transactions. The dataset consists of transactional features like Amount, Time, Class, and 28 other numerical features obtained from PCA. The class label 1 indicates fraudulent transactions and 0 for legitimate transactions. The dataset has no missing values. We did a log transformation of Amount features to get a normal distribution with normalized values between 0 and 1.

CIS Fraud Dataset: The dataset consists of real-world e-commerce transactions provided by Vesta [3]. Fraudulent transactions are 3.50% of the total 590540 transactions. There are a total of 434 transactional features comprising both categorical and numerical features. We target encoded all the categorical features for pre-processing, applied PCA on Vesta engineered features (present with column names V1-V339), and reduced their size to 20. Then we performed the Standard Scaling on all features. We imputed the missing feature values with the most frequently occurring values. A total of 105 features were used for the modeling.

4.2 Implementation Details

We have implemented the proposed framework in Pytorch. The generator module is a series of 4 fully connected (FC) layers. The hidden layers have output dimensions 128, 256, and 512, respectively, followed by ReLU activation. The output layer dimension for European Credit-Card Dataset is 30 and 104 for CIS Fraud Dataset. We feed a Normal Gaussian noise of dimension 30 to the generator module. The discriminator module takes synthetic fraud samples obtained from the generator and real fraud samples and tries to distinguish between them. The discriminator is a series of 4 FC layers with output dimensions 512, 256, 128 and 1, respectively. The ReLU activation function follows all the linear layers except the last linear layer.

We train the entire framework for 200 epochs on a batch size of 64 with the Adam optimizer and a learning rate of 2e-4. We train the discriminator first for five iterations in each epoch and then train the generator module. We select the best performing model on a validation set across ten independent runs and take the mean to report the Precision, Recall, and F1 score. We set the value of m in equation 4 as 10 and β as 0.3. The hyper-parameters in equation 7 α_1 and α_2 are set as 0.3 for CIS Fraud Dataset and 0.9 and 0.5 respectively for European Credit-Card Dataset. For experiments, we split both the datasets into 70% train and 10% validation and 20% test maintaining the event rate (the ratio of no. of fraud samples and the total samples) in each set. Splitting is done based on timestamps, with train always before validation and validation before test in terms of timestamps .

4.3 Results

We propose adversarial fraud generation as part of the supervised learning framework for oversampling the minority class. To evaluate the proposed framework, we compare the classifier's performance when trained on the augmented set from the proposed framework and other existing oversampling methods. In particular, we compare our method with (1) without Data Augmentation (1)SMOTE [11] (2)WGAN [4] (3)Conditional WGAN (WCGAN) [4] (4) WGAN-GP+ Auxiliary Classifier + DRS [22].

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud
[2] https://www.kaggle.com/c/ieee-fraud-detection
[3] https://www.vesta.io/

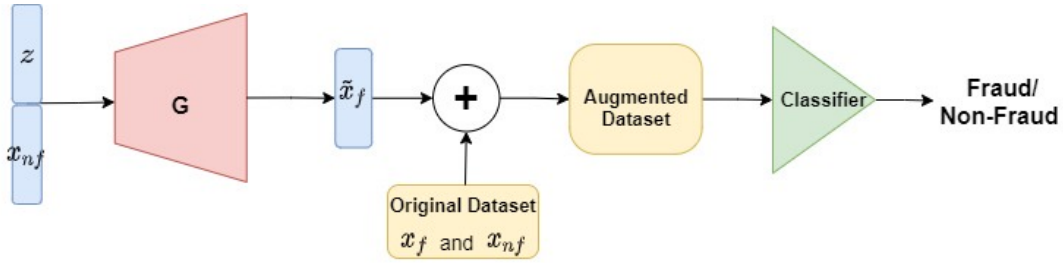


Figure 3: An illustration of the fraud detection pipeline using the proposed Adversarial Fraud Generation framework. We generate the synthetic fraud samples through the trained Generator module G. Next we augment the training set with the synthetic fraud samples and train a classifier for fraud detection task. At the test time, we pass the samples through the trained classifier to detect fraud.

	European Credit Card Dataset			CIS Fraud Dataset		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Logistic Regression	0.87	0.74	0.80	0.56	0.15	0.23
Decision Tree	0.87	0.73	0.79	0.27	**0.33**	0.30
Random Forest	0.86	0.76	0.81	**0.79**	0.29	**0.42**
XGBoost	**0.90**	**0.76**	**0.82**	0.68	0.28	0.40

Table 1: Model Benchmarks on European Credit Card and CIS Fraud Dataset. The performances are reported at the default threshold of 0.5.

Augmentation Method	Precision	Recall	F1-score
None	0.900	0.762	0.824
SMOTE [11]	0.981	0.703	0.819
WGAN [4]	0.880	0.788	0.831
WCGAN [4]	0.880	0.789	0.832
WGAN-GP+ Auxiliary Classifier + DRS [22]	0.929	0.790	0.854
WCGAN-GP + Adversarial loss (Our)	**0.934**	**0.803**	**0.864**

Table 2: European Credit Card Dataset: Performance of XGBoost classifier trained on the dataset augmented by synthetic fraud samples from different augmentation techniques using deep generative models.

Augmentation Method	Precision	Recall	F1-score
None	0.795	0.289	0.424
SMOTE [11]	0.778	0.289	0.422
WGAN [4]	0.802	0.295	0.432
WCGAN [4]	0.80	0.296	0.432
WGAN-GP+ Auxiliary Classifier + DRS [22]	**0.805**	0.297	0.434
WCGAN-GP + Adversarial loss (Our)	0.802	**0.302**	**0.440**

Table 3: CIS Fraud: Performance of Random Forest classifier trained on the dataset augmented by synthetic fraud samples from different augmentation techniques using deep generative models.

For both the datasets, European credit card fraud and CIS fraud, we compare the performance of several ML classifiers like Logistic Regression, Decision Tree, Random Forest, and XGBoost and finally pick the best performing model for further evaluation. Table 1 shows the best performing model for European Credit Card and CIS fraud Datasets are XGBoost and Random Forest, respectively. We compare the above-mentioned oversampling baselines with the best-performing classifier for the respective dataset. The performance of each classification method is measured in terms of Recall, Precision, and F1 score. F1-score is the harmonic mean of recall and precision. We include the F1 score into our metrics to make a conclusive decision when there is a trade-off between precision and

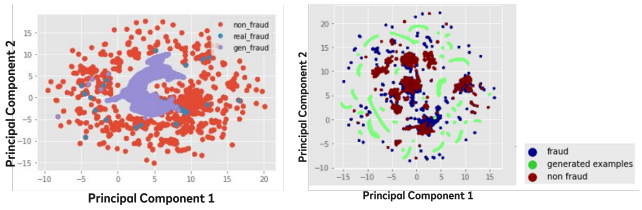


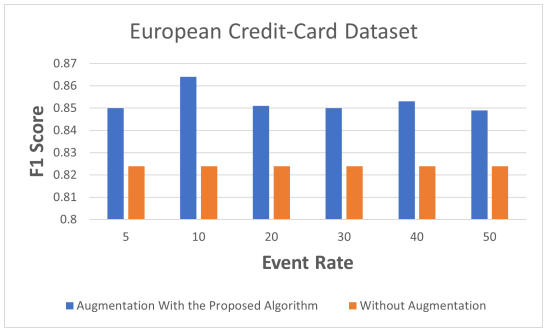
Figure 4: Visualization of synthetic fraud samples generated from (a) WGAN (b) Proposed Adversarial Fraud Generation Model.

recall. We present the classification results observed after ten runs for each oversampling technique. We add synthetic fraud samples equivalent to the number of real fraud samples in the dataset for data augmentation. Further increasing the synthetic fraud samples in the augmented set led to no significant improvement in the performance. All the performances were reported at the default threshold of 0.5.

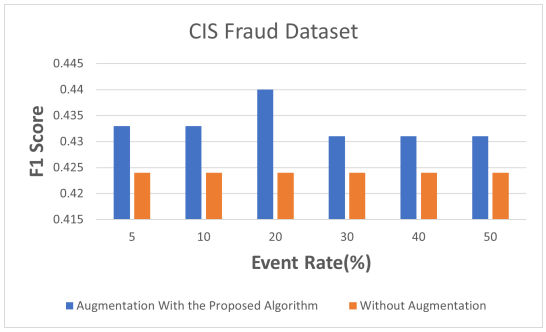
Table 2 and 3 shows the fraud detection results on the two imbalanced public fraud detection datasets. From both the tables, we can observe a reduction in the F1 score with SMOTE. SMOTE is a commonly used machine learning technique for minority oversampling. It does a linear interpolation of the existing samples to generate new samples. However, they fail to learn the fraudulent class distribution and can't generate new fraud samples lying on complex manifolds. Among the existing generative models for oversampling, WGAN-GP+Auxialiary Classifier + DRS [22] has proved to generate better quality samples. We can see from both the tables that augmenting the dataset with samples generated from the model has improved the F1 score. However, our method has outperformed all the baselines in terms of Recall and F1 score while maintaining Precision. On the European Credit Card dataset, there is a 1.3% and 1% absolute lift in Recall and F1-score, respectively, as compared to the best performing model. And on the CIS Fraud dataset, there is a 0.5% and 0.6% absolute lift in Recall and F1-score, respectively, as compared to the best performing model.

We further analyze the quality of synthetic data generated by our proposed algorithm and compare it with the synthetic samples generated by a basic WGAN model. Figure 4 shows the visualization of samples produced by both the algorithms. We can observe that the WGAN model fails to capture the diversity of the fraud samples, and all the samples are generated from the region where fraud density is high, i.e., it generates fraud samples similar to commonly occurring fraud patterns. However, our model is able to generate samples closer to the scattered fraud samples that got missed by the WGAN architecture. Hence augmenting the training set with these synthetic fraud samples helps in improving the Recall, as shown in the Tables 2 and 3.

We additionally compare the performance of our framework with another method called one-class adversarial nets (OCAN) [29] on fraud detection. OCAN has evaluated the performance on the European Credit Card dataset with 1000 non-fraud samples and 100 fraud samples in the test set and reported the area under the ROC curve to be 0.96 and 0.975 with raw features and transaction representation obtained from an Autoencoder, respectively. We have



(a)



(b)

Figure 5: Effect of increasing the synthetic fraud samples in the augmented set on F1 score of fraud detection. It can be observed that adding the augmented data result in improved performance (blue bars) across different event rates and datasets as compared to the model trained without augmentation (orange bars).

used the same test set and the performance metric for evaluation to have a fair comparison. The AUC for the ROC curve using our framework is 0.976.

4.4 Impact of amount of oversampling on Fraud Detection

In this section, we study the impact of increasing the number of synthetic samples in the augmented set on the classifier's performance. We increase the event rate (the ratio of no. of fraud samples and the total samples) by increasing the number of synthetic fraud samples in the augmented set. And plot the classifier's performance on fraud detection by varying event rate from 5% to 50% as shown the Figure 5. The best F1 score is obtained at 10% event rate for European Credit-Card Dataset and at 20% event rate for CIS Fraud Dataset. With addition of the synthetic samples performance is bound to improve. However, the event rate at which the maximum lift is obtained varies with the dataset. This could be a potential research direction to identify how many samples should be added to obtain the best performance.

5 CONCLUSION

The proposed method overcomes major drawbacks in current deep generative model-based data augmentation techniques for fraud detection. Existing methods are often limited in their capability to generate a wide variety of minority class samples, especially those that overlap with the majority class. To this end, our solution generates more realistic fraud samples for enhancing the representation of minority class resulting in improved classifier performance with low false positives. The use of non-fraud samples to generate fraudulent samples paired with chosen adversarial loss selected for our purpose improves the GAN training, resulting in an enhanced representation of frauds in the generated samples. The proposed method has shown to improve the fraud classifier performance when trained using a dataset with augmented generated frauds.

REFERENCES

[1] S Akila and U Srinivasulu Reddy. 2018. Cost-sensitive Risk Induced Bayesian Inference Bagging (RIBIB) for credit card fraud detection. *Journal of computational science* 27 (2018), 247–254.

[2] John Alipio, Patricia Castro, Hong Kaing, Noreen Shahid, Omar Sherzai, George L Donohue, and Karl Grundmann. 2003. Dynamic airspace super sectors (DASS) as high-density highways in the sky for a new US air traffic management system. In *IEEE Systems and Information Engineering Design Symposium, 2003.* IEEE, 57–66.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[4] Hung Ba. 2019. Improving Detection of Credit Card Fraudulent Transactions using Generative Adversarial Networks. *arXiv preprint arXiv:1907.03355* (2019).

[5] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. 2012. DBSMOTE: density-based synthetic minority over-sampling technique. *Applied Intelligence* 36, 3 (2012), 664–684.

[6] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Yacine Kessaci, Frédéric Oblé, and Gianluca Bontempi. 2019. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences* (2019).

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[8] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. 2017. Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems* 29, 8 (2017), 3784–3797.

[9] Alex GC de Sá, Adriano CM Pereira, and Gisele L Pappa. 2018. A customized classification algorithm for credit card fraud detection. *Engineering Applications of Artificial Intelligence* 72 (2018), 21–29.

[10] Vaishnavi Nath Dornadula and S Geetha. 2019. Credit card fraud detection using machine learning algorithms. *Procedia Computer Science* 165 (2019), 631–641.

[11] Ugo Fiore, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. 2019. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences* 479 (2019), 448–455.

[12] Jiaxin Gao, Zirui Zhou, Jiangshan Ai, Bingxin Xia, and Stephen Coggeshall. 2019. Predicting credit card transaction fraud using machine learning algorithms. *Journal of Intelligent Learning Systems and Applications* 11, 3 (2019), 33–63.

[13] Xin Guo, Johnny Hong, Tianyi Lin, and Nan Yang. 2017. Relaxed wasserstein with applications to gans. *arXiv preprint arXiv:1705.07164* (2017).

[14] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*. Springer, 878–887.

[15] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 1322–1328.

[16] Youngjoon Ki and Ji Won Yoon. 2018. PD-FDS: purchase density based online credit card fraud detection system. In *KDD 2017 Workshop on Anomaly Detection in Finance*. PMLR, 76–84.

[17] Eunji Kim, Jehyuk Lee, Hunsik Shin, Hoseong Yang, Sungzoon Cho, Seung-kwan Nam, Youngmi Song, Jeong-a Yoon, and Jong-il Kim. 2019. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. *Expert Systems with Applications* 128 (2019), 214–224.

[18] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2794–2802.

[19] Sumit Misra, Soumyadeep Thakur, Manosij Ghosh, and Sanjoy Kumar Saha. 2020. An autoencoder based model for detecting fraudulent credit card transaction. *Procedia Computer Science* 167 (2020), 254–262.

[20] Xuetong Niu, Li Wang, and Xulei Yang. 2019. A comparison study of credit card fraud detection: Supervised versus unsupervised. *arXiv preprint arXiv:1904.10604* (2019).

[21] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. 2020. Deep learning for financial applications: A survey. *Applied Soft Computing* (2020), 106384.

[22] Anubha Pandey, Deepak Bhatt, and Tanmoy Bhowmik. 2020. Limitations and Applicability of GANs in Banking Domain.. In *ADGN@ ECAI*.

[23] Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo. 2020. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* 133 (2020), 113303.

[24] Akhil Sethia, Raj Patel, and Purva Raut. 2018. Data Augmentation using Generative models for Credit Card Fraud Detection. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 1–6.

[25] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792* (2018).

[26] Huang Tingfei, Cheng Guangquan, and Huang Kuihua. 2020. Using Variational Auto Encoding in Credit Card Fraud Detection. *IEEE Access* 8 (2020), 149841–149853.

[27] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. 2018. Random forest for credit card fraud detection. In *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 1–6.

[28] Yusuf Yazici. 2020. Approaches to Fraud Detection on Credit Card Transactions Using Artificial Intelligence Methods. *arXiv preprint arXiv:2007.14622* (2020).

[29] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. 2019. One-class adversarial nets for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1286–1293.

[30] Xinxin Zhou, Zhaohui Zhang, Lizhi Wang, and Pengwei Wang. 2019. A model based on siamese neural network for Online transaction fraud detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.

[31] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*. PMLR, 7614–7623.