Journal of the Operational Research Society

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjor20

Generative adversarial networks for data augmentation and transfer in credit card fraud detection

Alex Langevin, Tyler Cody, Stephen Adams & Peter Beling

To cite this article: Alex Langevin, Tyler Cody, Stephen Adams & Peter Beling (2021): Generative adversarial networks for data augmentation and transfer in credit card fraud detection, Journal of the Operational Research Society, DOI: 10.1080/01605682.2021.1880296

To link to this article: https://doi.org/10.1080/01605682.2021.1880296

Published online: 01 Mar 2021.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2021.1880296

ORIGINAL ARTICLE

Generative adversarial networks for data augmentation and transfer in credit card fraud detection

Alex Langevin, Tyler Cody, Stephen Adams and Peter Beling

Department of Engineering Systems & Environment, University of Virginia, Charlottesville, Virginia, USA

ABSTRACT

Augmenting a dataset with synthetic samples is a common processing step in machine learning with imbalanced classes to improve model performance. Another potential benefit of synthetic data is the ability to share information between cooperating parties while maintaining customer privacy. Often overlooked, however, is how the distribution of the data affects the potential gains from synthetic data augmentation. We present a case study in credit card fraud detection using Generative Adversarial Networks to generate synthetic samples, with explicit consideration given to customer distributions. We investigate two different cooperating party scenarios yielding four distinct customer distributions by credit quality. Our findings indicate that institutions skewed towards higher credit quality customers are more likely to benefit from augmentation with GANs. Relative gains from synthetic data transfer, in the absence of feature set heterogeneity, also appear to asymmetrically favour banks operating on the lower end of the credit spectrum, which we hypothesise is due to differences in spending behaviours.

ARTICLE HISTORY
Received 10 February 2020
Accepted 14 January 2021

KEYWORDS
Synthetic data; fraud detection; generative adversarial networks; machine learning; transfer learning; distributional analysis

1. Introduction

Payment card fraud represents a large and growing problem faced by consumers and financial institutions around the globe. On a worldwide basis, losses from card fraud were estimated to be $27.85 billion in 2018, an increase of 16.2% from 2017 (HSN Consultants Inc, 2019). Much of this growth appears to be linked to the increase in e-commerce payments and the amount of personal information being stored – and leaked – online. In the UK for example, of the £671.4 million in payment card fraud losses in 2018 (up 19% from 2017), e-commerce related card fraud made up over half the total at £393.4 million, an annual increase of 33%, and card identity theft made up £47.3 million, an increase of 59% over 2017 (UK Finance, 2019). Meanwhile in the United States, the Federal Trade Commission catalogued 163,257 reports of credit card fraud in 2018, of which 130,298 were tied to new account openings, an increase of 24% over the previous year (Federal Trade Commission, 2019). To help combat this growing problem, financial institutions are increasingly turning to machine learning as an automated, real-time solution.

One of the main challenges in fraud detection is the rarity of occurrence. In the UK, there were 2.6 million fraudulent transactions in 2018 out of 20.4 billion payments, or 0.013% of payment volumes (UK Finance, 2019). By transaction value, fraud losses equated to 0.084% of payments in the UK in 2018 (UK Finance, 2019), the global figure is 0.069% (HSN Consultants Inc, 2019). Compounding this, fraudulent behaviour is constantly changing (Cody et al., 2018; Mead et al., 2018; Zeager et al., 2017), making older datasets and the fraudulent instances in those datasets obsolete when training new machine learning models to detect fraud. Many of the recent advances in machine learning, and in particular deep learning, rely on access to large quantities of data that are representative of the population being modelled. Given the rarity of positive cases of fraud relative to non-fraudulent payments, the limited lifespan of training data, and the expense of creating and labelling large datasets, fraud datasets often do not contain enough samples to train effective machine learning models that can generalise to the population in question.

The scarce number of fraud cases relative to non-fraud instances is one manifestation of the general problem of class imbalance whereby observations of one or more minority classes of interest are dominated by the frequency of occurrence of some other majority class(es). Class imbalance occurs in a number of fields from fraud detection to cybersecurity to medical diagnosis and bioinformatics, and often results in poor classification performance of machine learning models for the minority classes,

CONTACT Alex Langevin arl4zk@virginia.edu Department of Engineering Systems & Environment, University of Virginia, Charlottesville, VA, USA

which are often the classes of greatest interest (Ali et al., 2015; Leevy et al., 2018).

A common solution to help alleviate problems associated with class imbalance is to augment the under-represented class with synthetic samples. Synthetic data generation methods are used not only for class re-balancing but also have other applications as well, e.g. computer vision where it is referred to as data augmentation and used to reduce model overfit (Shorten & Khoshgoftaar, 2019).

One class of generative models that have seen tremendous growth in popularity and domains of application, due in part to their ability to model and sample from complex, high-dimensional, often unspecifiable distributions in "Big Data" settings, is Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). In our study, we choose GANs as a flexible and scalable approach to generating synthetic samples for credit card fraud detection, building on the work of Fiore et al. (2019), and adding to the literature on data augmentation for class imbalanced learning in several key regards. Through a partner financial institution, we have been given access to a dataset of nearly 80 million credit card transactions, with a diverse feature set containing multiple data types, allowing for an evaluation of data augmentation in a Big Data setting. Several of the data types require special consideration in modelling the distribution of the data, and we propose and evaluate a simple and straightforward mechanism for modelling boundary-constrained features with GANs. In addition, given the size of the dataset, we control the customer distribution according to customer credit quality, partitioning the data into four distinct customer distributions with varying proportions of higher and lower credit quality customers and examining the impact of data augmentation subject to distributional considerations. To the best of our knowledge, such a distributional analysis has not been conducted before, perhaps due to dataset size constraints.

An additional benefit to using GANs in generating synthetic samples is that they implicitly capture the underlying distribution of the data being modelled. We leverage this feature to solve a potential circular problem in imbalanced data scenarios, namely that there may be too few minority class samples from which to create synthetic samples of sufficient quality for learning. Our proposed majority-minority GAN transfer framework first models the conditional distribution of the majority class, then utilises some portion of the learned majority GAN architecture to train a separate GAN on the minority class, with the main hypothesis being that the majority class can be modelled with greater accuracy, including regions of the distribution that overlap with the minority class – information that can then be used to better model the minority class.

It is also the case in class imbalanced learning that data augmentation generally is performed on the minority class only, often in combination with some form of under-sampling of the majority class. In our general statistical framework for data augmentation, detailed in Section 3, we note an argument can also be made for augmenting the majority class with synthetic data if it can provide a learning algorithm with access to previously unobserved samples, and so in our data augmentation experiments we also investigate potential benefits to adding synthetic majority samples to our training sets.

Since in practice GANs will tend to model a distribution only approximately, can generate synthetic samples in arbitrary quantities, and are not created directly from real customer transactions as in some other generation techniques (Chawla et al., 2002), there is a potential secondary use of these samples as a form of private information transfer between cooperating parties.

Both the GAN transfer framework and sharing of synthetic samples are a form of transfer learning (Pan & Yang, 2010), which can be loosely described as taking knowledge gained from one domain or distribution (the source) and applying it to another domain or distribution (the target). The GAN transfer framework is referred to as parameter transfer, while the synthetic data sharing would be considered sample-based transfer. Transfer learning is useful for learning in settings with limited data and for avoiding retraining models from scratch when underlying distributions change, or in this setting where limited data from one class may prevent the effective modelling of the data distribution for synthetic sampling and/or classifier model training. In both transfer learning cases there lies an opportunity to leverage outside sources of information for improved model performance. In the latter case, however, there are several reasons why cooperating institutions would not share real customer data, including legal and ethical considerations, as well as competitive concerns.

Synthetic data transfer between financial institutions represents one mechanism for private knowledge sharing, the other main tools being federated learning (McMahan et al., 2017), and privacy-preservation mechanisms such as differential privacy (Dwork, 2011). Differential privacy (DP) is a framework in which each time a private database is queried, whether for machine learning or other purposes, a carefully selected amount of random noise is added to the response, such that a mathematical bound can be placed on the potential amount of privacy loss to individuals in that database, one of the main reasons for its popularity. Federated learning is a form of multiparty machine learning in which a common model is trained between

cooperating institutions or individuals while each party maintains exclusive access to and control over its own private database. These methods need not be mutually exclusive – previous work has combined differential privacy with both synthetic data generation (Beaulieu-Jones et al., 2019; Jordon et al., 2019; Li et al., 2014) as well as multiparty deep learning (Wang et al., 2018; Zhang et al., 2019).

We see synthetic data sharing as a promising avenue for private multiparty learning, since it requires neither a common model nor common feature set, allowing each party to select their own (possibly proprietary) model, while potentially gaining access to both previously unobserved samples and features. Admittedly synthetic data generation on its own does not provide provable privacy guarantees such as with DP, but as mentioned can be combined with DP or other privacy mechanisms such as postprocessing to remove synthetic records too closely resembling real client data.

In anticipation of future research into synthetic data and privacy, our study includes a secondary analysis that examines the potential benefits of synthetic sample transfer between financial institutions. We present two transfer scenarios – one in which the cooperating parties have similar customer distributions, and another where the two hypothetical financial institutions have distributions skewed towards either end of the credit spectrum. We intentionally do not consider heterogeneity in feature sets between cooperating parties, but rather focus on how distributional differences between parties impact the gains from transfer, or potentially losses – a phenomenon known as negative transfer.

Our contributions can be grouped and summarised as follows:

1. Synthetic Data Generation on Imbalanced Data
   a. Outlining a general framework for generating synthetic data samples for one or more imbalanced classes using a novel GAN transfer procedure
   b. An investigation of the use of synthetic data from both majority and minority classes, rather than just the minority class
   c. A simple mechanism for generating bounded features with GANs
2. Private Transfer Learning with Synthetic Data
   a. An evaluation of the benefits of synthetic sample transfer learning between parties with non-identical customer distributions

- Improved Realism of Data Augmentation Evaluation
  a. Testing Data Augmentation on a "Big Data" scale
  b. Incorporating a diverse feature set into the modelling process
  c. Evaluating the benefits of data augmentation under several customer distribution scenarios

The remainder of the article is structured as follows: Section 2 provides a background and technical primer of the core concepts that form the backbone of the study. Section 3 presents a general statistical framework for understanding class imbalance and data augmentation, and outlines our proposed majority-minority GAN transfer methodology. Section 4 lays out the study design and experiments, with results presented in Section 5. We conclude the article with a more in-depth discussion the results and their implications in Section 6 along with study limitations and avenues for future research.

2. Background

2.1. Class imbalance & data augmentation

Kaur et al. (2019) classify solutions to the class imbalance problem into three broad categories: preprocessing approaches, algorithmic approaches, and hybrid approaches. Algorithmic approaches tend to focus on the design of the learning model or loss function, while preprocessing approaches involve some form of data manipulation, with the hybrid approach being a combination of the two. Of the various methods surveyed, Kaur et al. (2019) note that sampling methods are some of the most popular mechanisms for dealing with class imbalance, in particular the SMOTE algorithm (Chawla et al., 2002). Sampling methods typically involve some sort data re-sampling in order to reduce imbalance between the classes. SMOTE does so by generating synthetic data to "over-sample" the minority class, which in the original framework was achieved by taking linear combinations of similar minority samples, as determined by the K-Nearest Neighbours algorithm. This can be done in combination with random under-sampling or "down-sampling" of the majority class. There are multiple proposed methods for the creation of synthetic samples, see e.g. Goodfellow et al. (2014); Kingma and Welling (2014); Sun et al. (2019) – Ali et al. (2015); Kaur et al. (2019) and Hittmeir et al. (2019) provide literature reviews or conduct comparative studies of additional methods.

The process of generating synthetic samples with which to augment a training dataset is not particular to class-imbalanced learning, and is often employed in image-based or computer vision tasks as a way to limit model overfit, for example to prevent a model from simply "memorising" a dataset during the

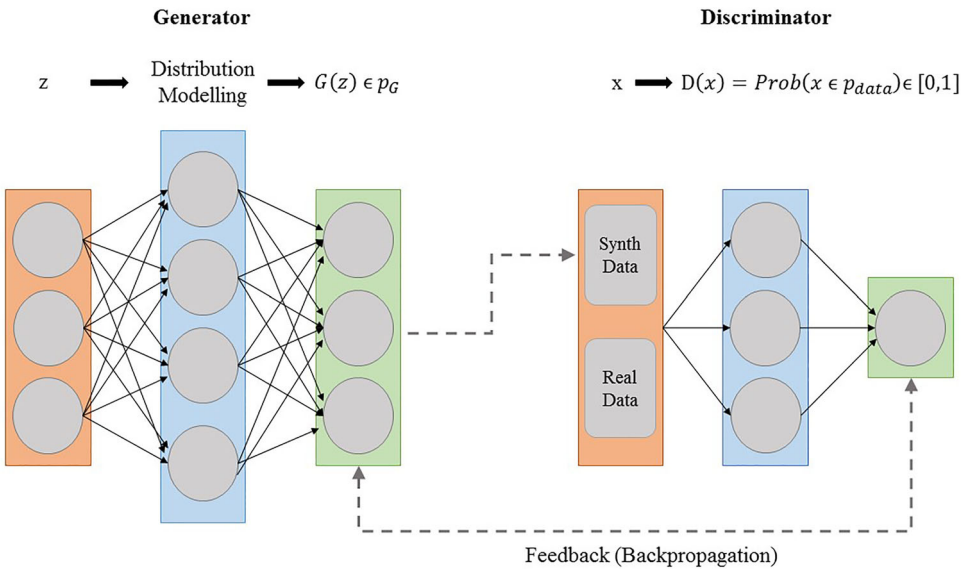


Figure 1. GAN training framework.

training. Shorten and Khoshgoftaar (2019) provide a comprehensive review of general data augmentation techniques for image-based tasks, including GAN-based methods.

GANs have enjoyed great success in image generation, and have begun to see applications to other domains (Beaulieu-Jones et al., 2019; Choi et al., 2017; Fiore et al., 2019), which has coincided with GAN extensions to generate non-continuous data such as discrete and categorical variables (Choi et al., 2017; Hjelm et al., 2018; Jang et al., 2017). In addition to image classification (e.g. Liu et al., 2018) and other computer vision applications, GANs and similar adversarial frameworks have been used for data augmentation in fields including biomedical informatics (Beaulieu-Jones et al., 2019; Choi et al., 2017; Lan et al., 2020), and machine fault detection (Shao et al., 2019). Fiore et al. (2019) have also recently applied GANs to generate synthetic minority samples for credit card fraud detection, while Douzas and Bacao (2018) have compared GANs for imbalanced data scenarios against several state-of-the-art methods.

2.2. Generative adversarial networks

GANs (Goodfellow et al., 2014) are a neural network-based training framework that uses a generator network to produce synthetic data and a discriminator network to distinguish between real and synthetic data. Through an iterative training process, the generator becomes progressively better at generating realistic looking synthetic samples, while the discriminator improves at discerning real data from fake. In theory, a GAN with sufficient model capacity will reach an equilibrium whereby the generator learns to sample directly from the data distribution, p_{data}. The generator does so by first drawing a noisy sample z from some prior distribution p_z, and applying a map $G : z \mapsto x$, where x ideally mimics a draw from p_{data}. The discriminator D then receives a sample x from either the generator or real dataset and returns a probability of x coming from p_{data}, as opposed to the generator distribution p_G. This results in the following minimax game,

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

Given enough capacity in both G and D, if D is trained to optimality after each update of G, then, the GAN will reach an equilibrium such that the Jensen–Shannon divergence between p_{data} and p_G is minimised, i.e. $p_G \sim p_{\text{data}}$. A visual representation of the GAN setup and training process appears in Figure 1.

The deep architecture of GANs naturally allow them to scale to high-dimensional scenarios. The widespread adoption of GANs by researchers and academics has been bolstered by advancements in network architectures and optimisation methods, such as conditional GANs (cGANs) (Mirza & Osindero, 2014) and Wasserstein GANs (WGANs) (Arjovsky et al., 2017; Gulrajani et al., 2017). cGANs are a variant on the original GAN framework whereby both generator and discriminator have additional input nodes to represent the class of the sample being generated or tested, and so the generator network instead learns to produce samples from B p_{data} conditioned on the class label C, i.e. $p_{\text{data}|C}$. WGANs are an alteration to the GAN training framework where instead of attempting to minimise the Jensen–Shannon divergence between p_G and p_{data}, the discriminator (now referred to as a critic) outputs a real number, with the loss function approximating the Wasserstein distance between p_G

and p_{data} up to a multiplicative constant. Using Wasserstein distance has some more desirable properties such as being continuous everywhere and differentiable almost everywhere, and can result in more stable training of GANs (Arjovsky et al., 2017). The WGAN proposed by Arjovsky et al. (2017) requires the imposition of a rather strict Lipschitz condition on the critic by directly clipping the weight parameters, which is later relaxed in the work of Gulrajani et al. (2017) by instead adding a gradient penalty to the loss function (WGAN-GP), which steers the critic towards learning a function with the desired Lipschitz condition. The WGAN architectures are tested alongside the original GAN framework in our experiments.

3. Methodology

3.1. A statistical framework for data augmentation

The problem of lack of positive samples in card fraud detection can be viewed as a more general problem of class imbalance with potentially multiple under-represented (minority) classes. Let X represent a random variable with parameters θ, i.e. $X \sim \theta$, where θ are generally unknown. Suppose that we have a random sample of n observations generated by X, denoted $\mathbf{x} = (x_1, ..., x_i, ..., x_n)^T$. Typically, $\mathbf{x}$ will contain only a subset of the possible observations generated by X, and follow a sample distribution $p(\mathbf{x}|\theta)$, denoted p_{data}, which may or may not match the population distribution of X, denoted p_X.

Let C_i represent the class of a given observation x_i. In the binary case, we have $C_i \in \{0, 1\}\ \forall i$ where 1 is the positive, or minority, class. In a supervised learning setting, we are attempting to train a model to learn the function $f : x_i \mapsto p_{C|X}(C_i = 1|x_i)$, where $p_{C|X}$ is the conditional distribution of the class labels, conditioned on the population random variable X. In reality we are training a model with $\mathbf{x}$ rather than X, and so we are learning a function $g : x_i \mapsto p_{C|\text{data}}(C_i = 1|x_i)$ (with a slight abuse of notation) which ideally closely approximates f.

Note that there are several sources of error that can lead to a poor approximation of $p_{C|X}$ for the minority class. One possibility is that the learned g is simply a poor approximator, whether by choice of model or loss function. In the event that the minority class suffers from severe imbalance, through the model training process we may obtain a local minimum for the loss function whereby $g(x_i) < k$ for most or all cases where $C_i = 1$, where $k \in [0, 1]$ is the classification threshold. Another possibility is that the model fits p_{data} very well, but p_{data} does not resemble p_X, either for the minority class(es) or more generally.

One potential solution to these problems with approximating f in the face of severe class imbalance is through data augmentation, or synthetic data generation. It is often the case that the minority classes are the focus of data augmentation with imbalanced datasets (Kaur et al., 2019; Ramyachitra & Manikandan, 2014), however, augmentation need not be restricted to minority classes. If we denote our synthetic sample of m observations as $\tilde{\mathbf{x}} = (\tilde{x}_1, ..., \tilde{x}_j, ..., \tilde{x}_m)^T$, we then have an augmented training set,

$$\mathbf{x}' = (\mathbf{x}, \tilde{\mathbf{x}})^T = (x_1, ..., x_n, \tilde{x}_1, ..., \tilde{x}_m)^T$$

which follows the distribution,

$$p_{\text{aug}} = \lambda p_{\text{synth}} + (1-\lambda) p_{\text{data}}$$

where $\lambda \in [0, 1]$ represents the proportion of synthetic data in the new augmented training set. If we were to augment all classes in equal proportion with synthetic data, we are assuming that for some value of λ, p_{aug} offers a better approximation to p_X than does p_{data}, either by providing previously unobserved samples to our dataset, or by altering the relative weights of observed samples in our training data. This in turn can lead to a better approximation of f, either through the selected model training on a better approximation of p_X, or by leading to the selection of a better model in the validation process. This formulation holds for data augmentation in general, and need not be specific to imbalanced data scenarios.

If we were to augment only the minority classes with synthetic data, the resulting p_{aug} has a slightly different formulation. We first note that by the law of total probability we can write p_{data} as,

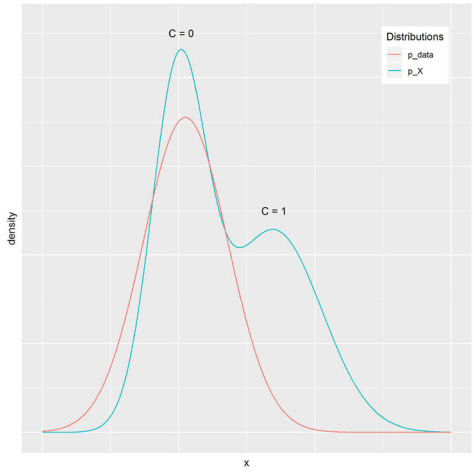
$$\begin{aligned} p_{\text{data}} = {} & p_{\text{data}|C=1} \cdot \text{Prob}_{\text{data}}(C = 1) \\ & + p_{\text{data}|C=0} \cdot \text{Prob}_{\text{data}}(C = 0) \end{aligned}$$

where $\text{Prob}_{\text{data}}$ is used to denote the fact that the probability of a certain class is dependent on the sample. By adding synthetic data to the minority class only p_{aug} now becomes,

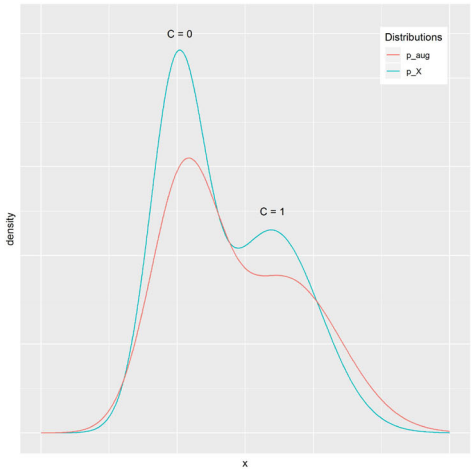
$$\begin{aligned} p_{\text{aug}} = {} & p_{\text{data}|C=0} \cdot \text{Prob}_{\text{aug}}(C = 0) \\ & + \left[\lambda p_{\text{synth}|C=1} + (1-\lambda) p_{\text{data}|C=1}\right] \\ & \cdot \text{Prob}_{\text{aug}}(C = 1) \end{aligned}$$

where $\text{Prob}_{\text{aug}}(C = 1) \neq \text{Prob}_{\text{data}}(C = 1)$ and $\text{Prob}_{\text{aug}}(C = 0) \neq \text{Prob}_{\text{data}}(C = 0)$. By augmenting the minority class only, we are altering our sample distribution in two ways. One is by directly changing the conditional distribution of the minority class $p_{\text{data}|C=1}$, and the other by indirectly shifting the sample class proportions through the addition of new minority samples.

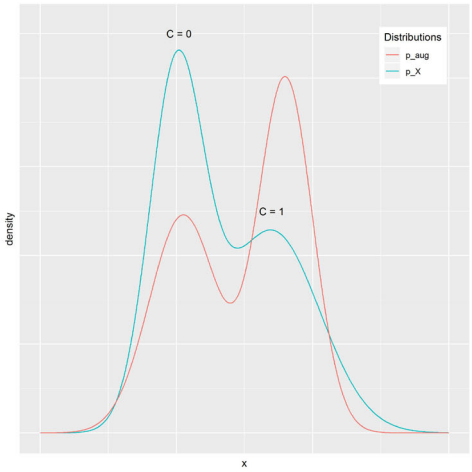
There are two possible interpretations of the choice to augment the minority classes only. One is



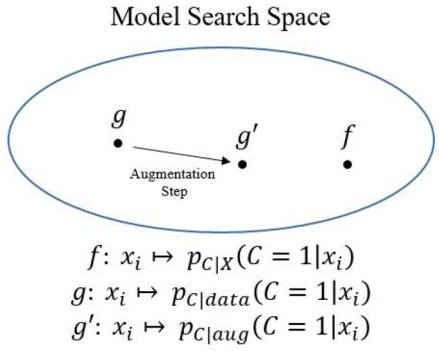
(a) Initial Sample Generated by p_X



(b) Augmented Sample - Interpretation 1



(c) Augmented Sample - Interpretation 2



(d) Desired Impact of Model Training with Augmented Data

Figure 2. A visual representation of the effects of data augmentation.

that $p_{\text{data}|C=1}$ provides a poor approximation to $p_{X|C=1}$ and also does not accurately reflect the true class proportions, which we attempt to correct through selective augmentation. Another possible interpretation is that by changing the relative weights of the classes, we can obtain a better approximation of f for the minority classes by altering the training behaviour of our model, regardless of whether p_{aug} offers a better approximation to p_X. This concept is depicted in Figure 2.

It is not immediately clear whether one augmentation procedure is preferable to the other, and so in our experiments we examine both. There is also the possibility that our choice of synthetic data generation method may not be able to create all possible observations that can be generated by X, in particular $\forall x \in X$ such that $p_{\text{data}}(x) \approx 0$. In the context of GANs, there is also a phenomenon known as mode collapse, where a generator learns to sample from a portion of the full distribution of p_{data}. In

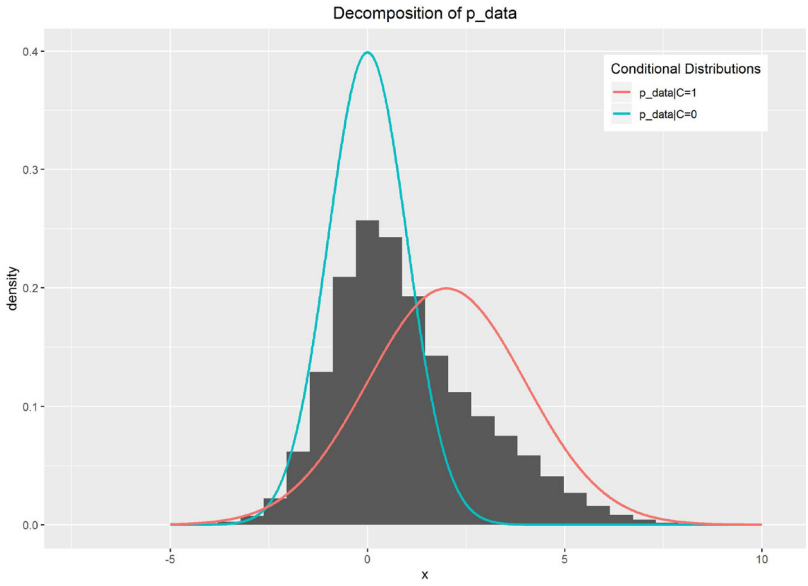


Figure 3. Visual representation of GAN transfer.

the case of a multimodal distribution, the learned p_{synth} may be centred around one of the many modes of p_{data}, and so in effect for a given x we may have $p_{\text{synth}}(x) \approx 0$ while $p_{\text{data}}(x) > 0$. This is where we see one potential benefit of sample transfer from a cooperating party, by providing additional samples that can be generated by X, but cannot be replicated locally. Since these transferred samples are generated by a different random variable Y which likely does not follow the same distribution as X, however, there will be a trade-off between the provision of additional samples that offer a better approximation to p_X, and skewing the distribution towards p_Y rather than p_X.

Although in theory a GAN generator G can model p_{data} exactly in the limit, in practice G will tend to offer only a noisy approximation. In a sense, we can then view data augmentation using G trained to model p_{data} as a special case of transfer learning, subject to the same trade-offs as when samples are transferred from a different institution. The main differences will be:

1. G is trained to approximate p_{data} and so the trade-off is likely to be less pronounced than with samples transferred from a different source
2. Transferred samples offer a second potential benefit in that the cooperating entity may collect additional features which are otherwise unobserved in p_{data}. The transferred samples then offer an additional source of information which will not be provided by samples generated from G

3.2. Majority-minority GAN transfer

When training a GAN to generate samples from minority classes, we run into a potential issue in that we wish to generate synthetic minority samples to augment our training set, but may have too few samples to effectively model this class, a "catch-22" of sorts. Our hypothesis is that by first training G to model the conditional distribution of the majority class $p_{\text{data}|C=0}$, we can learn some high level distributional characteristics that are common to $p_{\text{data}|C=0}$ and $p_{\text{data}|C=1}$, but can be more effectively modelled on the majority class given the relative abundance of samples from which to learn. We then first train G to approximate $p_{\text{data}|C=0}$, and in doing so capture any additional information that can be learned about $p_{\text{data}|C=1}$ from our sample, before transferring G to the minority class to continue training, to learn the low level distributional features specific to the minority class.

A visual intuition for our rationale is presented in Figure 3 – note the distribution of p_{data} is represented by the histogram. By leveraging the abundance of majority class examples, the intuition is that a generator can be trained to more accurately model $p_{\text{data}|C=0}$, including the overlapping regions of $p_{\text{data}|C=0}$ and $p_{\text{data}|C=1}$. By training first on $p_{\text{data}|C=0}$ then transferring to the minority class, the generator will capture more information about $p_{\text{data}|C=1}$ than would be possible by training only on the minority class. This assumes that $p_{\text{data}|C=0}$ and $p_{\text{data}|C=1}$ overlap for some region(s) of the distribution, which we see as reasonable, otherwise the classes would be easily separable making the classification task straightforward, and limiting the need for synthetic data to begin with.

This leads to our proposed framework for generating synthetic samples from imbalanced datasets, which can be applied to datasets with one or more minority classes, and well as include a transfer component. We denote a party's own data or models by a superscript L (local component), and transferred elements by a superscript T (transfer component).

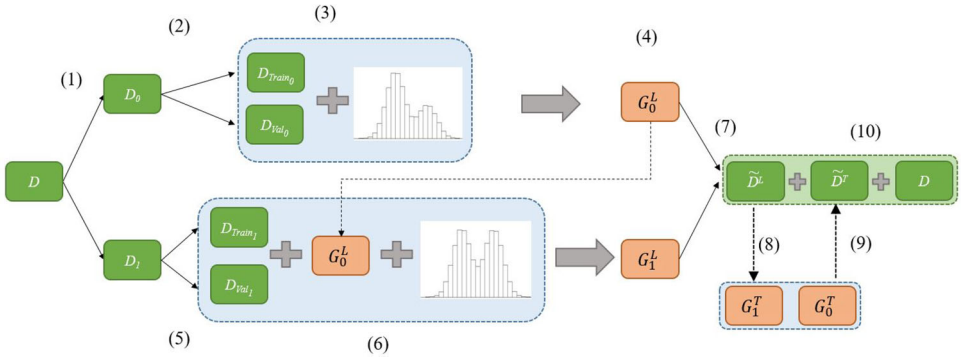


Figure 4. Majority-minority GAN transfer framework.

We also use the same notation as above, where an index value of $C = 0$ refers to the majority class, and $C = 1$ the minority class. We present our framework for the binary case:

1. For a given dataset D, partition D into its constituent classes D_0 and D_1
2. Further partition D_0 into training and validation sets, D_{Train_0} and D_{Val_0}
3. Train a generator G_0^L on D_{Train_0} using D_{Val_0} to select optimal hyperparameters
4. Save G_0^L, and initialise G_1^L using architecture and weight parameters of G_0^L.
5. Partition D_1 into D_{Train_1} and D_{Val_1}
6. Train G_1^L on D_{Train_1} using D_{Val_1} to select optimal hyperparameters
7. Generate synthetic data for desired class(es), $\tilde{D}_0^L$ and/or $\tilde{D}_1^L$
8. (optional) Transfer $\tilde{D}_0^L$ and/or $\tilde{D}_1^L$ to cooperating parties
9. (optional) Obtain transferred synthetic samples $\tilde{D}^T$
10. Augment D with synthetic data $D_{\text{aug}} = \{D, \tilde{D}^L, \tilde{D}^T\}$. If (9) ignored then $\tilde{D}^T = \emptyset$
11. Utilise D_{aug} as desired

Figure 4 gives a visual diagram of the majority-minority GAN transfer process. Note that in step (4) of the framework, when transferring a generator from the majority to minority class(es), we are seeking to transfer additional information on the distribution, which will be contained in the architecture and model weights. We do not see a justification for transferring model hyperparameters such as learning rates and batch size – these can be tuned for example by random search when training the minority class generator(s). There are also several possible ways in which to transfer the architecture and weight parameters of the majority class generator, which we test in our experiments.

4. Experiment details

This section outlines the dataset as well as the experimental setup used to test the hypothesis and framework described in Section 3 – additional details can be found in Appendix A.

4.1. Dataset preprocessing

The dataset for this case study was provided by a partner financial institution and is composed of 77.6 million credit card transactions from the first eight months of 2013. The original dataset has 69 discrete and approximately continuous (e.g. dollar-valued) variables, with a fraud rate of 0.14%. After deriving several additional features with transformations similar to those applied by Wang et al. (2018), and conducting feature selection to reduce the dimensionality of the data, we were left with 50 features, 20 numeric, and 30 categorical, for the GAN augmentation experiments.

Several of the selected features were either lower-bounded (e.g. card limits) or had both upper and lower bounds (e.g. proportion of transactions by day of week). In our proposed mechanism for handling these features, lower-bounded features were log-transformed, and features with both upper and lower bounds were scaled to [0,1] and logit transformed, with small amounts of noise added to boundary values to avoid "Not a Number" issues with the transformations.

4.2. Customer distribution scenarios

In order to facilitate the fraud detection experiments and introduce heterogeneity to the distributions, the processed and transformed data were partitioned into two datasets representing two hypothetical financial institutions, Bank A and Bank B. The full dataset (less 50,000 samples used for feature selection) was first ordered by account credit limit. Data were initially sampled from transactions falling below the median credit limit at a fixed sampling ratio and assigned to Bank A. Data for Bank A would then be over/under sampled above the median credit limit so as to equalise the expected number of fraudulent transactions for each bank.

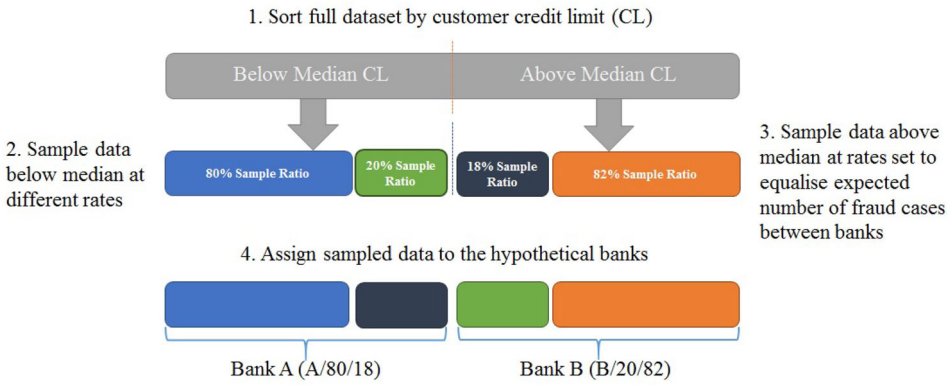


Figure 5. Dataset partitioning procedure - skewed distribution scenario.

The remaining transactions after sampling were assigned to Bank B.

Two alternate customer distribution scenarios were examined, resulting in four datasets in all. In the first scenario, Bank A's data were randomly sampled as 55% of all below median credit limit transactions, and 45% of above median transactions (referred to as A/55/45). Bank B's data were the mirror image of Bank A, composed of 45% of below median transactions, and 55% of all above median transactions (B/45/55). This first partition represents the case where the hypothetical banks have similar customer distributions as measured by credit quality, with Bank A having relatively lower credit quality, or "subprime" customers, assuming credit limit is a valid proxy for credit quality. In the second scenario, we sample Bank A/80/18 and Bank B/20/82, simulating two banks that have customer profiles skewed toward either end of the credit spectrum. Figure 5 illustrates the sampling procedure.

Once the data were divided, 5% of each dataset – circa 2 million transactions including approximately 2500 fraudulent samples – was set aside for out-of-sample testing in the fraud model experiments.

4.3. Distribution modelling

To allow for the generation of arbitrary quantities of fraud and non-fraud transactions, and to examine the benefits of the proposed GAN transfer step, the conditional distributions of each class of transaction were modelled separately. Two methods of training fraud data were examined in the model validation process:

1. Majority-minority GAN transfer as detailed in Section 3
2. Independently learn a GAN model on the fraud data

The best architectures using each method were taken to the fraud detection modelling step for further validation. cGANs were another possible method to model the sample distribution while being able to control the class balance of the generator, although it was hypothesised in the experimental design phase that without some mechanism to control the sampling frequency of each class in the GAN training process, in a Big Data setting the cGAN generator would effectively learn the non-fraud component of the distribution, and so was not considered in the experiments.

For each of the four datasets, 1.5 million non-fraud transactions were used for GAN architecture selection and hyperparameter tuning by random search, using the PyTorch deep learning library for model training. Categorical features were modelled using the Gumbel – Softmax trick (Jang et al., 2017), while several bounded feature approaches were tested in addition to the proposed transformations. A summary of the key hyperparameters of the selected models can be found in Appendix B.

GAN models were validated using three criteria. The first was an approximation to the Wasserstein distance, or Earth Mover's Distance (EMD) between the generator's distribution and the sample distribution, using the Gower distance metric as ground truth and computed using the Python Optimal Transport library (Flamary & Courty, 2017).

The other two validation criteria used were standard machine learning classifiers – logistic regression and random forests – trained to distinguish between the synthetic and real data. At each validation step, a new batch of synthetic data would be generated and the classifiers trained on this new synthetic batch along with the real validation samples. A second batch of synthetic data would then be generated, and the accuracy of the validation classifiers on these synthetic samples would be taken as the validation score. A score of 0.5 signifies that the classifiers achieve coin toss accuracy in determining whether the synthetic data is indeed fake. GANs that showed strong and consistent performance across all three metrics were selected for both the fraud GAN training step, and further validation in the data augmentation experiments.

As mentioned previously, for fraud GAN training we attempted to transfer the best performing non-fraud architectures for further training on fraud data, as well as attempted to construct new architectures by random search of the hyperparameters. The following transfer methods were attempted when re-training non-fraud GANs on fraud data:

1. Re-train the selected non-fraud model directly on fraud data
2. Re-initialise the final 1–2 layers of the critic and/or generator before re-training on the fraud data
3. Re-initialise all weights and re-train on fraud data using the selected non-fraud architectures only

The fraud GANs were trained on circa 42,500 fraud transactions, with 5000 samples held aside for model validation. As with the non-fraud GANs, the candidate models which consistently performed best across all three validation criteria were selected for further validation with the data augmentation experiments.

4.4. Fraud detection experiments

4.4.1. Validation of benchmarks and GAN architecture selection

Once candidate GAN models for each bank were selected, the datasets were combined and re-partitioned into training and validation sets for fraud detection model training. The re-use of data across two sets of models does not present an issue, since the data are being used for two different purposes – distribution modelling in one instance, and fraud detection in the other. After partitioning, each of the four banks across the two scenarios had a training set, two validation sets, and a test set. While the training and validation sets were formed from data re-purposed from the distribution modelling step, the test sets were not used at any point apart from obtaining out-of-sample performance estimates for the fraud detection models.

In the first round of validation, model selection was performed using only real data in the training set. In early experiments the LightGBM (LGBM) (Ke et al., 2017) implementation of gradient boosted trees produced the most competitive results, and so all further experiments used LGBM models. In imbalanced data settings, Area Under the Curve (AUC) can give a misleading picture of model performance (Davis & Goadrich, 2006), and so we opted for F1 score for fraud classifier validation and performance measurement.

F1 score can be interpreted as the harmonic mean of recall and precision – the ability of a model to correctly identify positive cases, and the accuracy of the predicted positive cases. By selecting a model based on F1 score, we are simultaneously seeking high precision and recall as well as a balance between the two. This is particularly appropriate in a financial crime detection setting, where false positives can make up a substantial portion of overall fraud costs – one analysis estimates that direct and indirect false positive costs make up close to half of losses from payment card fraud (Crossfield & Griffin, 2017).

For the real data only model, the non-fraud transactions were first randomly down-sampled at different rates, a commonly employed data sampling technique in class imbalanced learning (Leevy et al., 2018). The training set was down-sampled to achieve fraud rates of 1–15%, and for each fraud rate, 250 models were tested by random search of the hyperparameters. The best performing model was selected by F1 score on the first validation set, which then served as the benchmark for the data augmentation experiments.

A similar experiment was conducted to select the best performing synthetic data generators from the candidate GANs. For each candidate generator, datasets of equal size to their respective real datasets were generated at fraud rates of 1–15%, and 250 LGBM models were tested on each synthetic dataset with hyperparameters tuned by random search. The generators that produced the best performing LGBM models were selected for the data augmentation experiments.

Once the optimal fraud rates, real data model benchmarks, and synthetic data generators had been selected, we undertook data augmentation experiments to examine how the use of synthetic data in classifier training affects model performance under each of the four distribution scenarios.

4.4.2. Data augmentation experiments

In the first data augmentation experiment, real non-fraud transactions were down-sampled to the best validation fraud rate, and the training set was subsequently augmented with both synthetic fraud and non-fraud data, with the total proportion of synthetic data varying from 1% to 50%. This was done using either local augmentation or transfer augmentation. Model performance after local GAN augmentation was benchmarked against the performance of the real data only model, as well as model performance after partner GAN augmentation. Once the datasets were augmented, the same candidate models obtained by random search on the real datasets (1000 models across the four datasets) were re-

trained on the augmented datasets, with the best model selected using the F1 score from validation set 2. The same set of models were used to control for the possibility of performance differences coming from better or worse hyperparameter random search results, rather than the data augmentation procedure.

A second experiment was also conducted, augmenting the fraud class only at synthetic proportions varying from 1% to 66%, or synthetic:real ratios of 1:99 to 2:1. By augmenting the fraud class only, less down-sampling of the non-fraud transactions is required in order to achieve a fraud rate of 1%. This combination of over-sampling the minority class and down-sampling the majority class is the general approach outlined in Chawla et al. (2002). The model selection procedure remained the same as in experiment 1, with the experiment conducted using both local and transfer augmentation.

Using the notation from Section 3, it is possible that by including more samples of the majority class from p_{data} rather than a noisy p_{synth}, the resulting p_{aug} from experiment 2 will be a better approximation to p_X. Conversely, if synthetic non-fraud samples from p_{synth} add previously unobserved samples that could be generated by our random variable X, then experiment 1 could achieve better results.

For both experiments, once the best performing LGBM models were selected, the test sets were used to obtain estimates of out-of-sample model performance for each of the three models – the real data model benchmark, the local augmentation-based model, and the transfer augmentation-based model. For each bank, the test set results are reported as training plots, showing the change in out-of-sample performance through each boosting round of the LGBM models on the training set. These plots are presented both on a bank's own test set, referred to as its local performance, and also on its partner bank's test set as a way to examine data augmentation's effect on a bank's ability to generalise its model, referred to as its generalisation performance. For example, the benchmark and augmented models for Bank A/55/45 were trained on Bank A/55/45's own data (with added synthetic samples in the augmented cases), with performance being measured on both the Bank A/55/45 test set (local performance) and the Bank B/45/55 test set (generalisation performance).

A third scenario was also examined whereby a bank's non-fraud customer distributions were considered stationary, while fraud data from both bank test sets were pooled and sampled at varying rates. In other words, each bank was evaluated using its own test set or legitimate customer base for non-fraud transactions, while fraudsters were considered "mobile" and free to circulate, not tied to a particular distribution but instead able to "test" their fraud techniques at different institutions. This experiment was conducted using the best performing augmentation rate-model combination for each augmentation scenario, based on validation F1 score.

As a comparison to GAN-based data augmentation techniques, each dataset was also augmented using a SMOTE variant from the Imbalanced-learn package in Python (Lemaître et al., 2017) that is capable of working with mixed continuous and categorical data, SMOTE-NC. A similar model selection process was performed using the SMOTE-augmented datasets, and out-of-sample performance estimates from the best validation SMOTE-NC models are presented alongside benchmark and GAN-augmented model performance. In subsequent sections, the benchmark or baseline model, local GAN model, partner GAN model, and SMOTE-NC model refer to the LGBM fraud classification model obtained from training on the corresponding augmented dataset.

5. Results

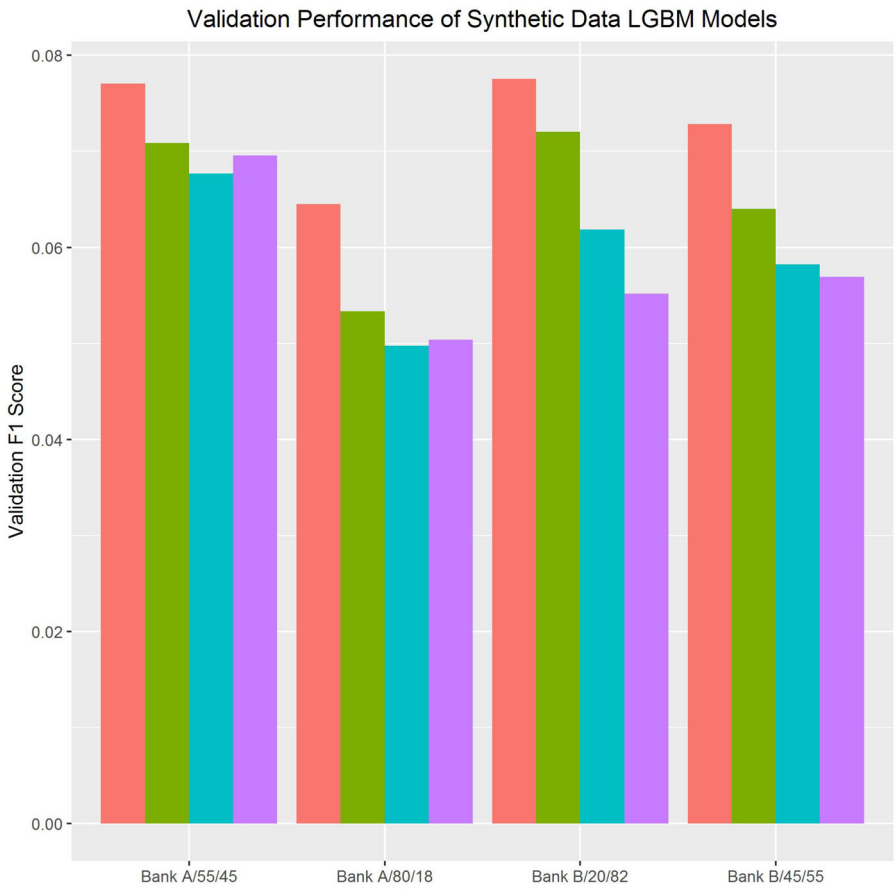
5.1. Validation results – benchmarks and GAN architecture selection

Figure 6(a) shows the validation performance of the best benchmark real data LGBM models at various fraud rates after 500 boosting rounds. In all scenarios, model performance drops off precipitously for fraud rates larger than 1%, and so data augmentation experiments were conducted at a 1% fraud rate.

Figure 6(b) shows the validation results of the best performing candidate synthetic data models after 500 boosting rounds. As in the real data case, a fraud rate of 1% yields the best F1 score on the validation set. The validation performance after training on synthetic data only is also considerably below the performance of the real data-trained models, suggesting that the GANs are able to capture some distributional information, although with a considerable amount of noise. From a privacy standpoint this may be beneficial, assuming the synthetic data remains useful, as it suggests there is a lower level of information leakage since the GAN is not perfectly modelling the customer distributions.

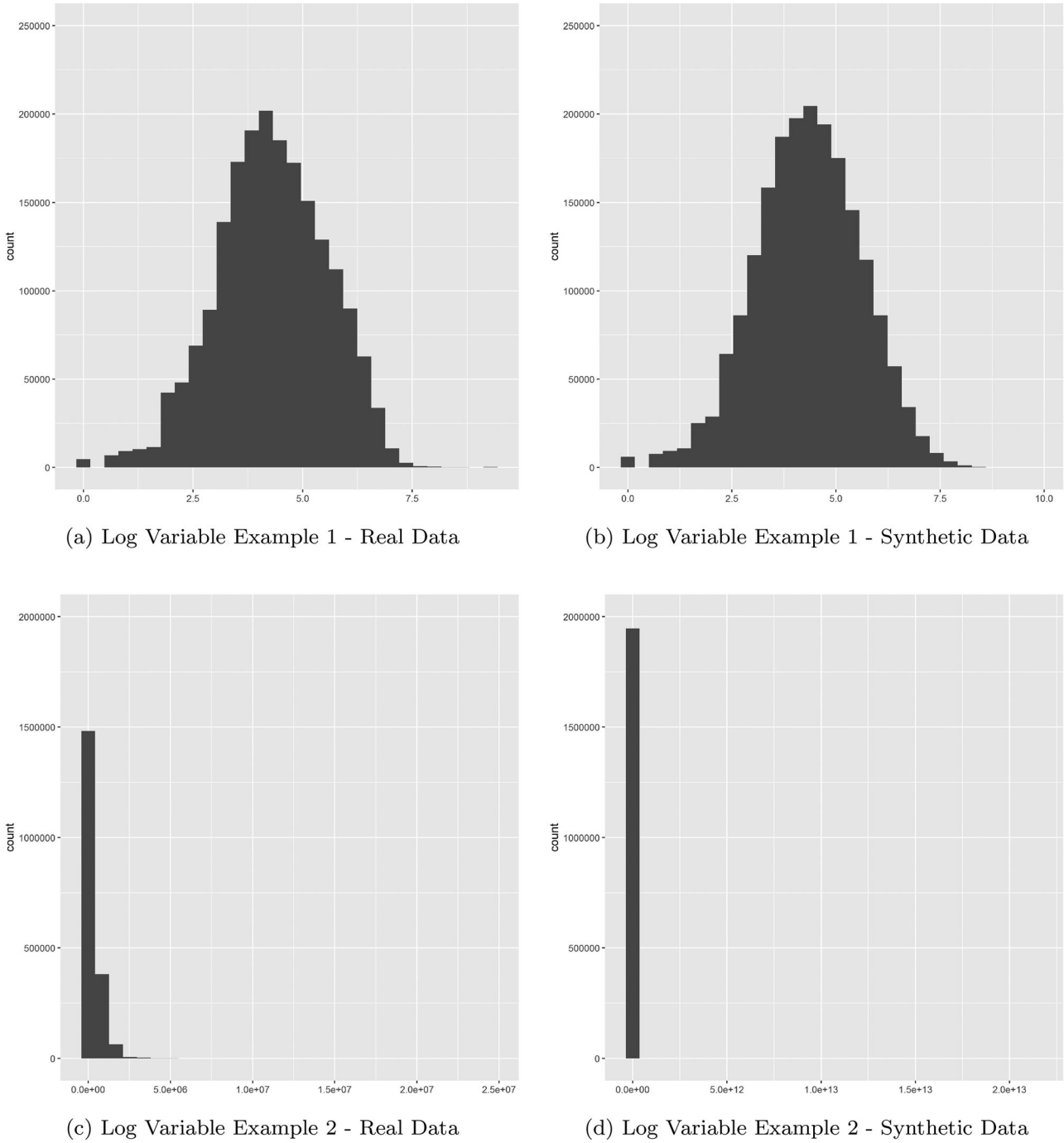
Across all four distribution scenarios, the best performing GAN models were cases where both the generator and critic were transferred to the fraud data using the trained weights and architecture of the non-fraud GAN, or with the final layer of the critic model re-initialised to random weights. Attempting to train a fraud GAN with random weight initialisations, or resetting any portion of the generator weights produced considerably worse

(a) Best Performing Real Data Benchmark Models



(b) Best Performing Synthetic Data Models

Figure 6. Best performing LGBM models after 500 boosting rounds.



(a) Log Variable Example 1 - Real Data

(b) Log Variable Example 1 - Synthetic Data

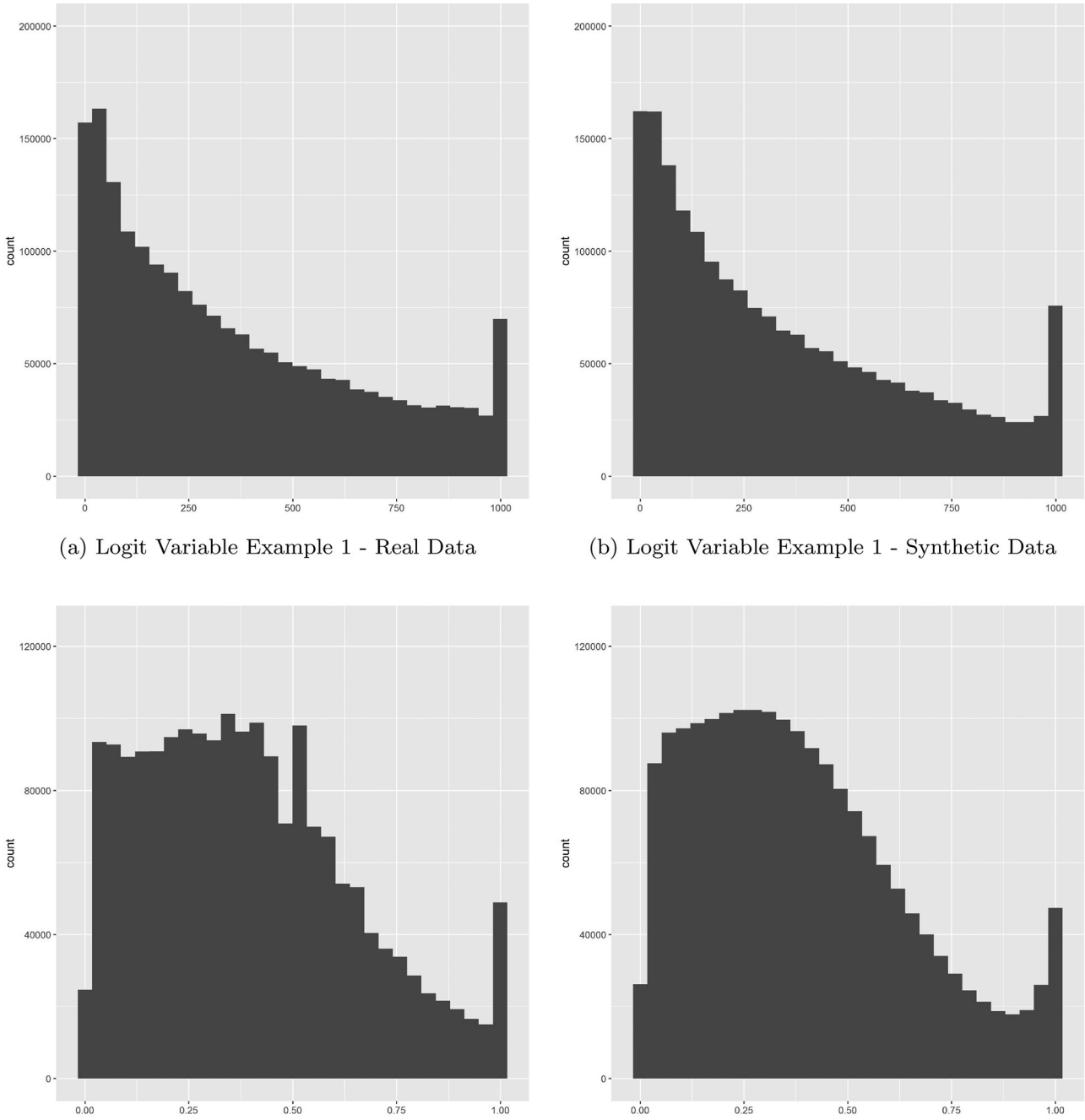
(c) Log Variable Example 2 - Real Data

(d) Log Variable Example 2 - Synthetic Data

Figure 7. Bank A/55/45 synthetic data univariate analysis - log transformations.

distributional approximations for the fraud class. This provides some evidence that the majority class GAN is able to capture additional distributional information about the minority class that would otherwise be missed by attempting to train a model from scratch directly on the minority class.

Also notable, the best performing GANs across all scenarios included the proposed log and logit data transformations – Figures 7 and 8 provide some side-by-side univariate comparisons. Figure 7(a,b) show an example of where the generators were able to approximately capture the univariate distribution of the log transformed variable. Figure 7(c,d) show an example of where the generators had difficulty in modelling the log transformed features on a univariate basis. It appears that where the log transformed variables included a long tail of extreme values, the generators struggled to model these tails, which resulted in some synthetic samples showing extreme values that were off by several orders of magnitude relative to the real data. In general, the univariate distributions of the logit transformed variables show that the GAN generators were able to capture the overall univariate structure of the features, including at the feature boundaries where the additional noise step was required – Figure 8 presents some examples. We further discuss implications of the log transformed results in Section 6.



(a) Logit Variable Example 1 - Real Data

(b) Logit Variable Example 1 - Synthetic Data

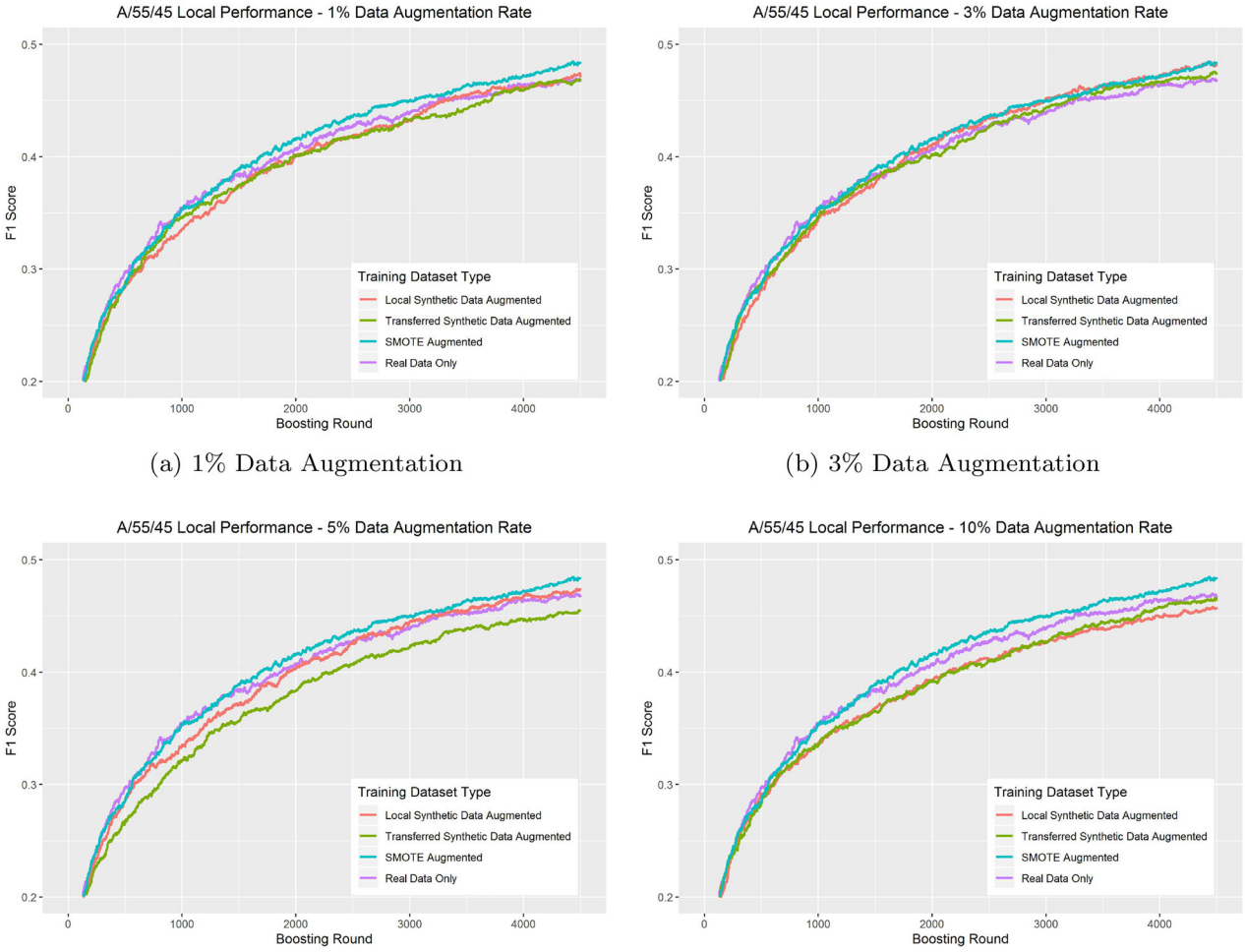
(c) Logit Variable Example 2 - Real Data

(d) Logit Variable Example 2 - Synthetic Data

Figure 8. Bank A/55/45 synthetic data univariate analysis - logit transformations.

5.2. Data augmentation experiments – Bank A/55/45

First looking at the similar bank scenario, the out-of-sample performance results of the mildly sub-prime bank's selected LGBM models are presented in Figure 9. This figure presents the model performance on Bank A/55/45's own (local) test set, as the models are progressively trained on either of the augmented datasets or Bank A/55/45's real data only, through 4500 boosting rounds. The best result for Bank A in the similar bank scenario was achieved in experiment 1 at a data augmentation rate of 3% for both fraud and non-fraud samples. At an augmentation rate of 3% using Bank A's own GAN, the difference between the local GAN model and baseline is 1.4 points of F1 score, or an F1 score of 0.482 for the local GAN model vs. 0.468 for the benchmark. Since F1 score $\in [0, 1]$, we also discuss results in terms of points, i.e. F1 score $\times 100$, as a matter of convenience. Note that this uplift in performance is also persistent – the local GAN model records a stronger test set F1 score for the majority of training, averaging 1.3 points over the final 500 boosting rounds. The local GAN model performs roughly on par with the SMOTE-NC model, which reaches an F1 score of 48.4 points by boosting round 4500, marginally outperforming the local GAN model, although for the latter half of the



(a) 1% Data Augmentation

(b) 3% Data Augmentation

(c) 5% Data Augmentation

(d) 10% Data Augmentation

Figure 9. A/55/45 local performance - experiment 1.

training rounds the two models show very similar performance.

At a 3% augmentation rate, the partner GAN model also outperforms the real data benchmark at round 4500, with a difference in F1 score of 0.5 points, however, when looking at the training plots, the baseline and partner GAN models exhibit a crossing pattern, often achieving similar test set performance, making it unclear whether the partner GAN model outperforms the benchmark. The results are also highly sensitive to the augmentation rate – once the data augmentation rate reaches 5%, the performance gains disappear, with performance worsening considerably as the proportion of synthetic data is increased further. At a 10% augmentation rate for example, the local GAN model performs worse than baseline for the majority of model training, and by boosting round 4500 the difference is 1.1 points of F1 score. At a 25% augmentation rate, this difference is 3.0 points, and by 50% the local synthetic data model performs 11.2 points worse than benchmark (10.4 point difference for the partner GAN model). It is worth noting that Bank A/55/45 was the only bank to show a clear benefit from augmenting the non-fraud transactions, the focus from this point is on the results from experiment 2.

Figure 10 shows Bank A/55/45's most noteworthy local performance result from experiment 2. At an augmentation rate of 1%, the partner GAN model displays a clear F1 score improvement through the boosting rounds relative to the baseline model. At round 4500, the partner GAN data model achieves a test set F1 score of 47.9 points compared to a benchmark of 46.8 points. With an F1 score of 48.4 points at round 4500 the SMOTE-NC model does look to offer even further gains, although as can be seen in the training plot both SMOTE-NC and partner GAN models often exhibit a crossing pattern in test set performance, and so the difference between augmented data models could be sensitive to an early stopping criterion, which was not considered in validating fraud classifiers. The average difference between the 1% partner GAN model and SMOTE-NC model over the final 500 boosting rounds was 0.1 points.

The next two scenarios attempt to examine how data augmentation impacts a model's ability to

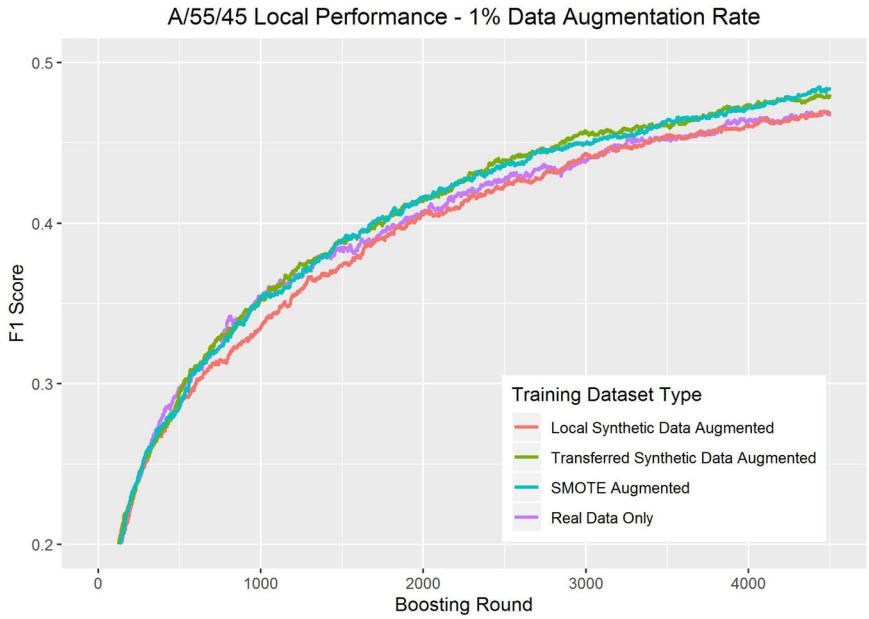
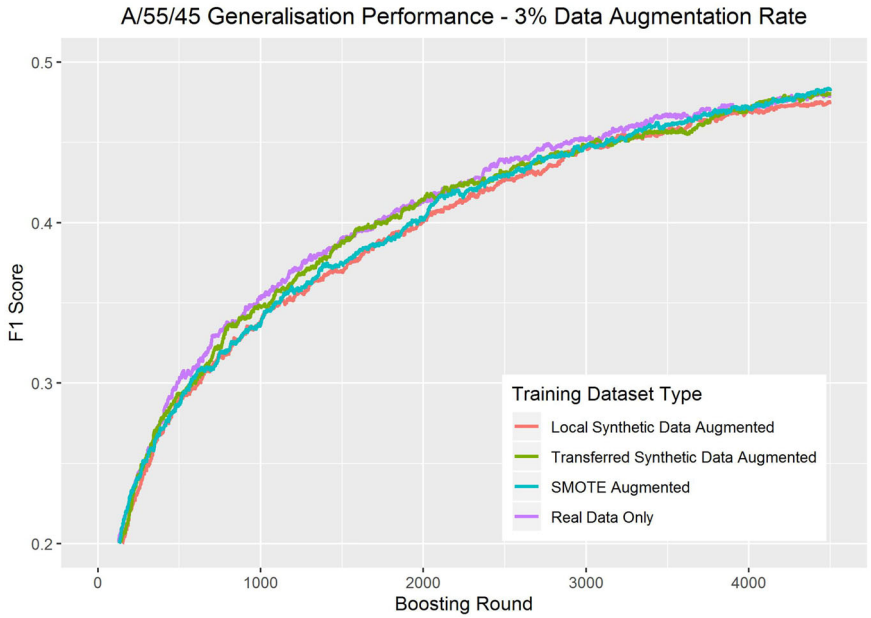


Figure 10. Bank A/55/45 local performance - experiment 2.

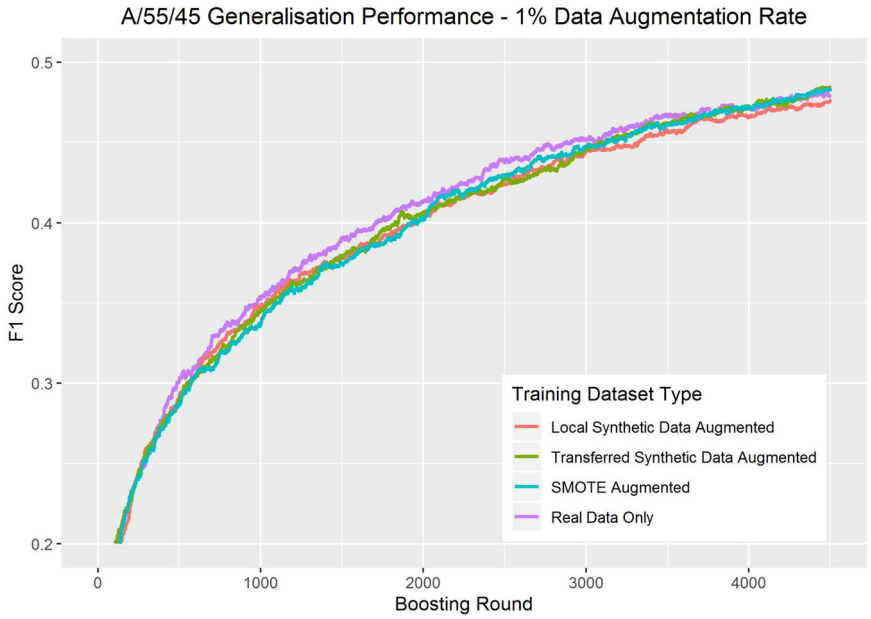
generalise to new fraud settings. Figure 11 displays Bank A/55/45 model performance on Bank B/45/55's test set, i.e. the ability of Bank A's model to generalise to other customer distributions. Figures 11a and 11b display the generalisation performance of the strongest validation models for the local GAN and partner GAN, respectively. In other words, these are the LGBM models that would be selected in a full validation process. In this case at a 1% fraud sample augmentation rate, the partner GAN model performs best, although roughly on par with SMOTE-NC – the partner GAN model achieves an F1 score of 48.4 points compared to 48.3 points for SMOTE-NC and 47.9 points for the baseline model. Similar to the local test set, generalisation performance of the GAN augmented data models decreases markedly as the proportion of synthetic data increases. In the case where the dataset is augmented with 3% fraud and non-fraud data, the local GAN model reaches an F1 score of 0.476, 0.3 points below baseline, and 0.7 points below the SMOTE-NC model.

In the pooled fraud experiment, we left the non-fraud test set samples unchanged, under the assumption that a bank's distribution of legitimate customers remains stationary, while the fraud cases were pooled from both Bank A and B test sets, and sampled at the original fraud ratio. For example, if Bank A had a fraud rate of 0.1% in a test set of 2 million samples, then from the combined fraud pool, 2000 fraud samples were drawn at random and joined with Bank A's non-fraud test set observations before obtaining the model's F1 score. This experiment attempts to simulate a more realistic scenario where a bank's customer distribution remains largely static in the short term, while fraudsters are able to freely circulate amongst the banks to try their luck. Figure 12 displays the performance of the best validation augmentation models compared to the baseline model, based on an average of 100 random draws from the combined fraud pool. The percentages on the horizontal axis represent the proportion of samples drawn from the partner bank's fraud pool – Bank B/45/55 in this case. Higher percentages attempt to estimate the Bank's model performance with higher rates of "new" or "unobserved" fraud behaviours or typologies, i.e. a high velocity amongst the fraudsters.

In this pooled fraud scenario, the baseline model's performance stays static at an F1 score of around 0.468 regardless of proportion of new fraud. At a 50% new fraud proportion, the SMOTE-NC augmented data model slightly outperforms the partner GAN model 0.483 vs. 0.481. At 60% new fraud cases, the performance is the same at 0.482, and past 60% new fraud cases the performance of the partner GAN model continues to improve, reaching 0.485 at close to 100% new fraud, compared to the SMOTE-NC model performance of 0.482 and a baseline model F1 score of 0.469, a difference of 0.3 points and 1.6 points, respectively. The selected local GAN model performs roughly on par with the SMOTE-NC model through the different sampling proportions, recording an F1 score of 0.481 at 50–80% new fraud cases, and 0.482 beyond that. Table 1 summarises the selected GAN models that offered the best validation performance for each bank.



(a) Experiment 1: 3% Data Augmentation



(b) Experiment 2 : 1% Data Augmentation

Figure 11. Bank A/55/45 generalisation performance - best validation models.

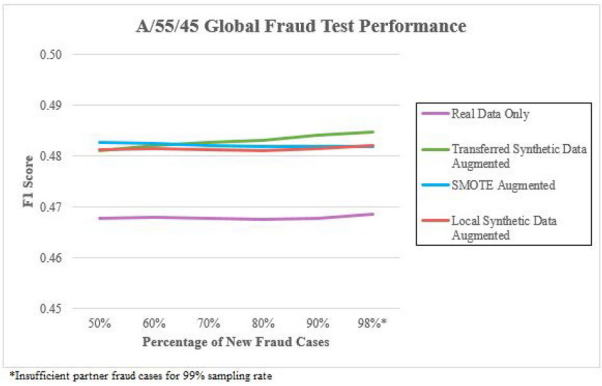
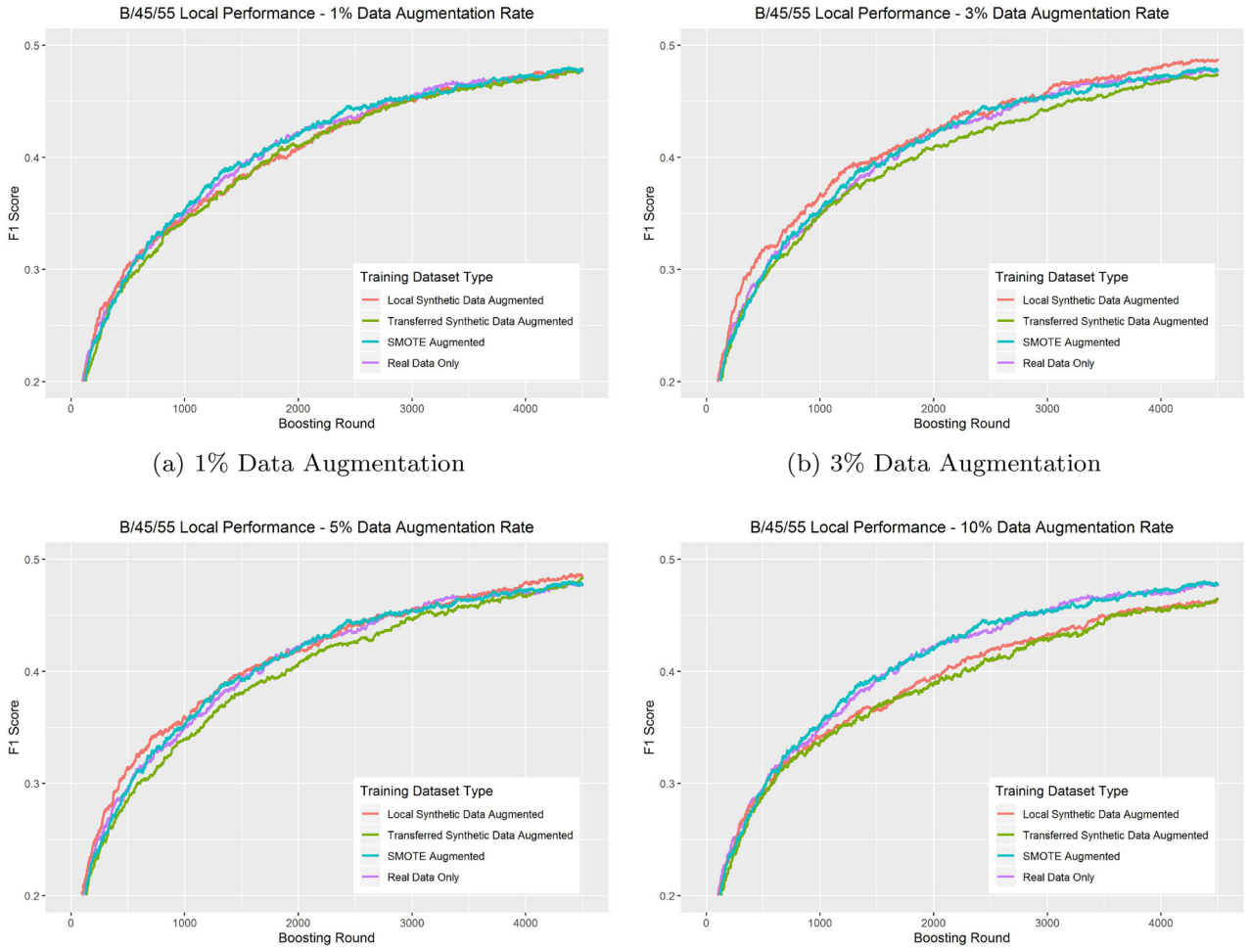


Figure 12. Bank A/55/45 realistic performance.

Table 1. Validation best GAN model summary performance.

Bank	Source GAN	Classes augmented	Augmentation rate	Baseline local performance	GAN model difference (local perf)	Baseline generalisation performance	GAN model difference (gen perf)
A/55/45	Local	Fraud & Non-Fraud	3%	0.468	1.4	0.479	−0.3
	Partner	Fraud Only	1%		1.1		0.5
B/45/55	Local	Fraud Only	3%	0.478	1.0	0.472	1.1
	Partner	Fraud & Non-Fraud	1%		0.3		−0.5
A/80/18	Local	Fraud Only	3%	0.491	−0.3	0.399	−0.5
	Partner	Fraud Only	1%		0.5		−0.5
B/20/82	Local	Fraud Only	1%	0.552	0.6	0.367	1.0
	Partner	Fraud Only	3%		−0.7		0.5



(a) 1% Data Augmentation

(b) 3% Data Augmentation

(c) 5% Data Augmentation

(d) 10% Data Augmentation

Figure 13. B/45/55 local performance - experiment 2.

5.3. Data augmentation experiments – Bank B/45/55

For Bank B/45/55, the mildly prime bank, the most significant local results are presented in Figure 13. When augmenting the fraudulent transactions with local synthetic data at a 3% rate, Bank B's local GAN model records an F1 score of 0.488 vs. a score of 0.478 for the benchmark and SMOTE-NC models. At a 3% augmentation rate the partner GAN model performs on par with the benchmark. At an augmentation rate of 5% both GAN models obtain an F1 score of 0.484 by the final boosting round, although in the case of the partner GAN model, this performance improvement over benchmark was not necessarily persistent – the gain over the benchmark model occurred in the final 100 or so boosting rounds, before which the partner GAN model consistently underperformed benchmark. In experiment 1, the GAN-based models performed worse than benchmark past a 1% augmentation rate.

As for the ability to generalise to new fraud cases and customer distributions, the local GAN model shows a clear and reasonably persistent improvement over the others. Figure 14 shows generalisation performance when augmenting the dataset with 3% synthetic fraud samples – using synthetic samples from Bank B/45/55's own GAN yields an F1 score

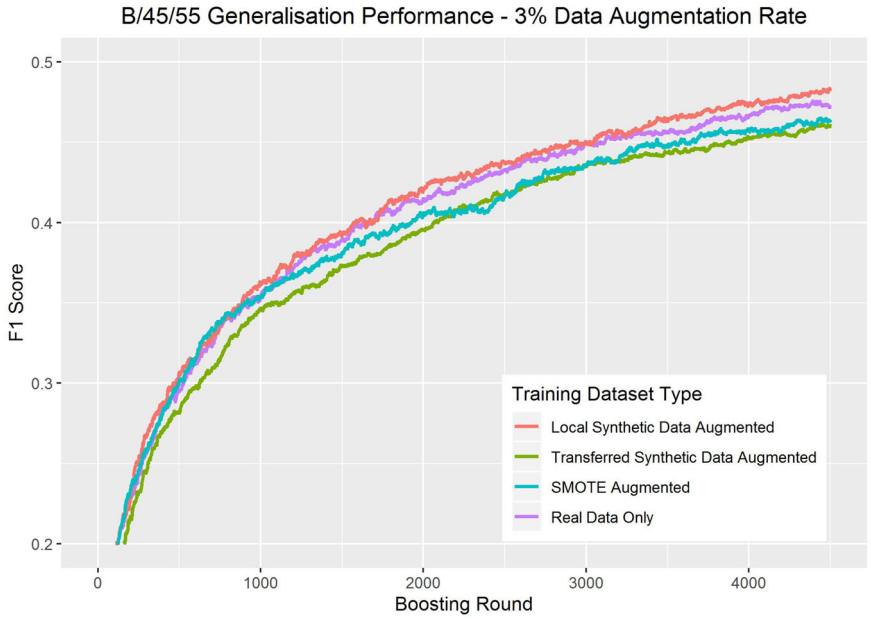


Figure 14. Bank B/45/55 generalisation performance - experiment 2.

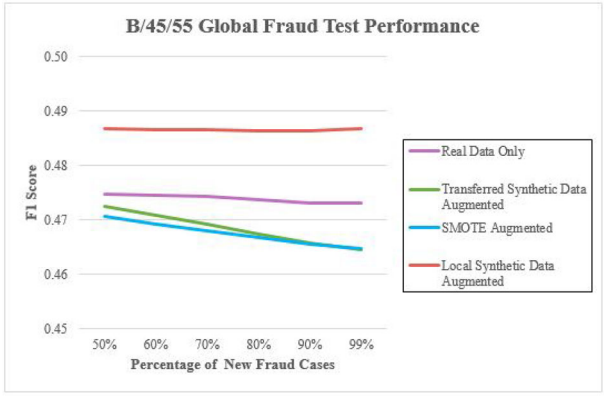


Figure 15. Bank B/55/45 realistic performance.

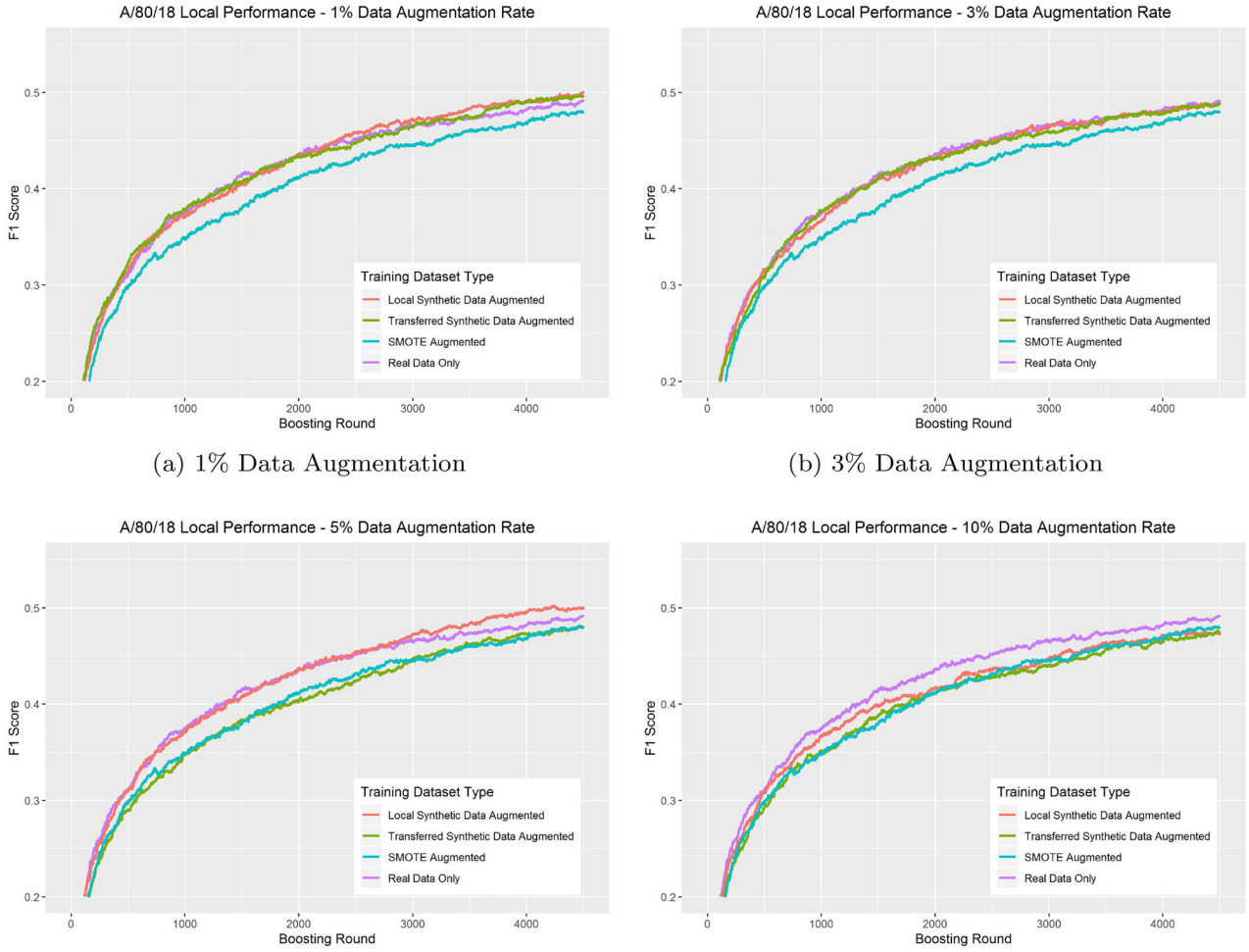
of 0.483, a 1.1 point improvement over the baseline model and 1.9 point gain compared to the SMOTE-NC model. Results from the pooled fraud scenario appear in Figure 15, again showing the local GAN model outperforming other methods across all sampling proportions. Across the various new fraud sampling rates, the local GAN model has average F1 scores of 0.486–0.487 compared to a baseline average of 0.475 at 50% new case sampling, decreasing to 0.473 at 99% new case sampling. The local GAN model selected for this experiment was augmented with 3% synthetic fraud samples, which is the same model that displayed the strongest local and generalisation performance. Both the partner GAN and SMOTE-NC models show declining performance as the proportion of new fraud cases is increased, from averages of 0.472 and 0.471, respectively, at a 50% new fraud rate, to 0.464 and 0.465, respectively, at a 99% sampling rate. The best validation partner GAN model in this instance had local performance comparable to the benchmark model, and showed worse than baseline generalisation performance.

5.4. Data augmentation experiments – Bank A/80/18

Looking at the local performance results from experiment 2 in Figure 16, the local GAN model shows the strongest test set performance – at a 1% augmentation rate the F1 score at boosting round 4500 is 0.501, a 1.0 point improvement over baseline. The partner GAN model also offers an improvement over the benchmark model with an F1 score of 0.496, while the SMOTE-NC model performs worst at 0.480.

Figure 17 displays the generalisation performance of fraud data augmented models compared to the benchmark model. In all cases, the models trained using augmented data performed worse than the benchmark model. The baseline had a generalisation F1 score (on Bank B/20/82's test set) of 0.399, with the SMOTE-NC model reaching a score of 0.379 by boosting round 4500. The validation best partner GAN model had an F1 score of 0.394 – a loss of 0.5 points compared to baseline – at a 1% augmentation rate. The best performing local GAN model obtained an F1 score of 0.395 at a 5% augmentation rate, again a loss compared to benchmark.

In the pooled fraud scenario all models saw a decline in performance as the sampling proportion of new fraud cases increases, as shown in Figure 18. The baseline model performs better than or equal to the selected augmented data models through the various sampling proportions. At a 50% new fraud case sampling rate the baseline model has an average F1 score of 0.460, declining to 0.428 at a 99% new fraud case proportion. The partner GAN model holds up best among the augmented data models, matching baseline at a 50% sampling rate, performing 0.1 points worse at 60% and 70% new fraud sampling rates, and 0.4 points worse at the 99%



(a) 1% Data Augmentation

(b) 3% Data Augmentation

(c) 5% Data Augmentation

(d) 10% Data Augmentation

Figure 16. A/80/18 local performance - experiment 2.

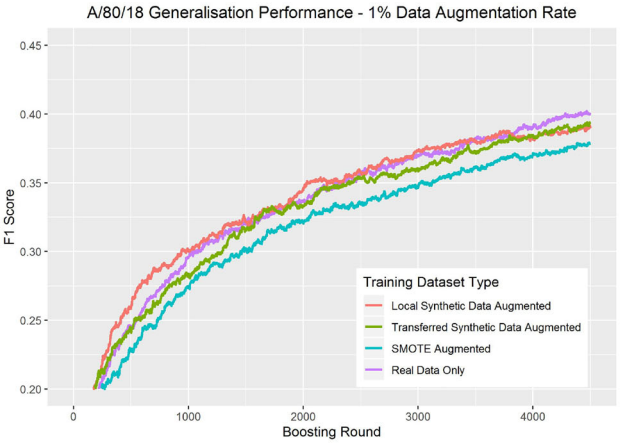
rate. The local GAN augmented model consistently performs 0.2 points worse than baseline, while the SMOTE-NC model has an F1 score 1.4–1.8 points below baseline.

5.5. Data augmentation experiments – Bank B/20/82

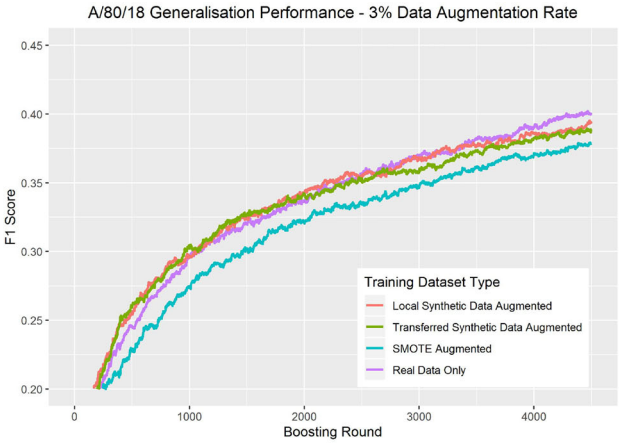
The gains in local performance for the strongly prime Bank B/20/82 were relatively more subdued compared to the other banks – Figure 19 displays the test set results for the only augmentation rate that showed a clear performance improvement over baseline. By augmenting the real fraud data with 1% synthetic samples from B/20/82's own GAN, test set F1 score improves to 0.558, or 0.6 points over baseline. The best validation SMOTE-NC model recorded a test set F1 score of 0.525, 2.7 points below baseline, while the partner GAN model never beat benchmark. The gains from the local GAN augmentation appear fairly persistent through the LGBM model training process, averaging 0.4 points over the final 500 boosting rounds, and 0.5 points over the final 1000.

The generalisation performance of Bank B's models on the heavily subprime bank's test set is shown in Figure 20. The local GAN model outperforms baseline at several different augmentation rates. At a 1% augmentation rate, the local GAN model yields an F1 score of 0.377 compared to a benchmark score of 0.367, and a SMOTE-NC score of 0.363. At 3% and 5% augmentation rates, the local GAN model reaches an F1 score of 0.381, an improvement of 1.4 points over benchmark, with gains disappearing at higher augmentation rates. The partner GAN model also outperforms benchmark at 3% and 5% augmentation rates, by 0.5 points and 1.0 points, respectively.

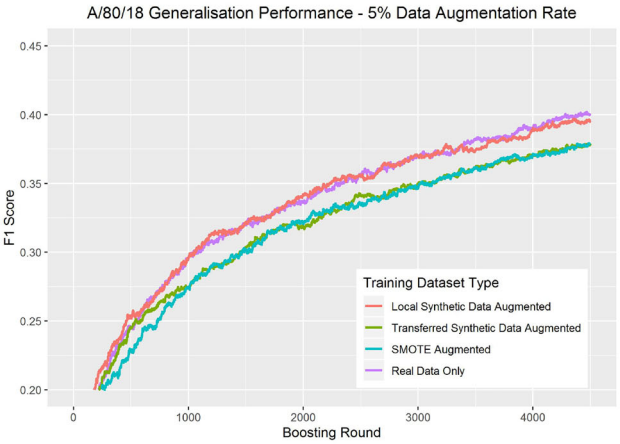
Finally, Figure 21 presents the results of the pooled fraud sampling scenario. All models see declining performance as the sampling proportion from Bank A/20/82's fraud pool is increased, with the baseline model recording a test set F1 score of 0.466 at a 50% new fraud sampling rate, decreasing to 0.370 at a 99% sampling rate. The validation-best local GAN model outperforms all others through the range of sampling proportions, beating baseline by 0.8 points at the 50% new fraud case rate, increasing to a 1.2 point difference at the 99%



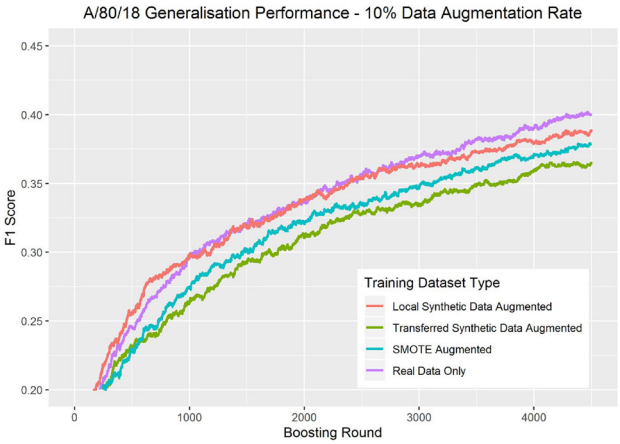
(a) 1% Data Augmentation



(b) 3% Data Augmentation



(c) 5% Data Augmentation



(d) 10% Data Augmentation

Figure 17. A/80/18 generalisation performance - experiment 2.

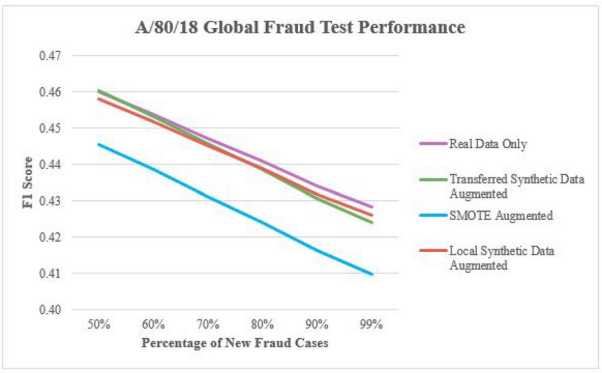


Figure 18. Bank A/80/18 realistic performance.

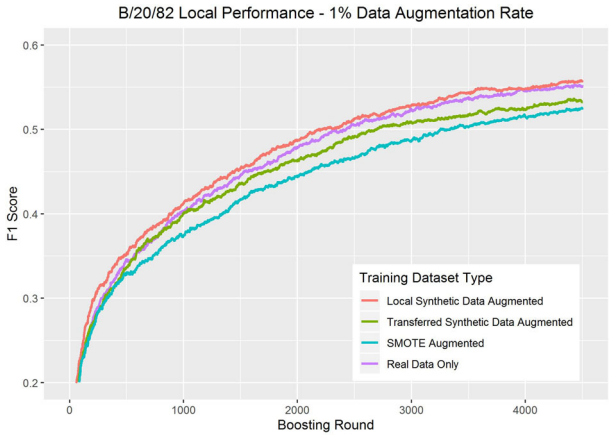


Figure 19. Bank B/20/82 local performance - experiment 2.

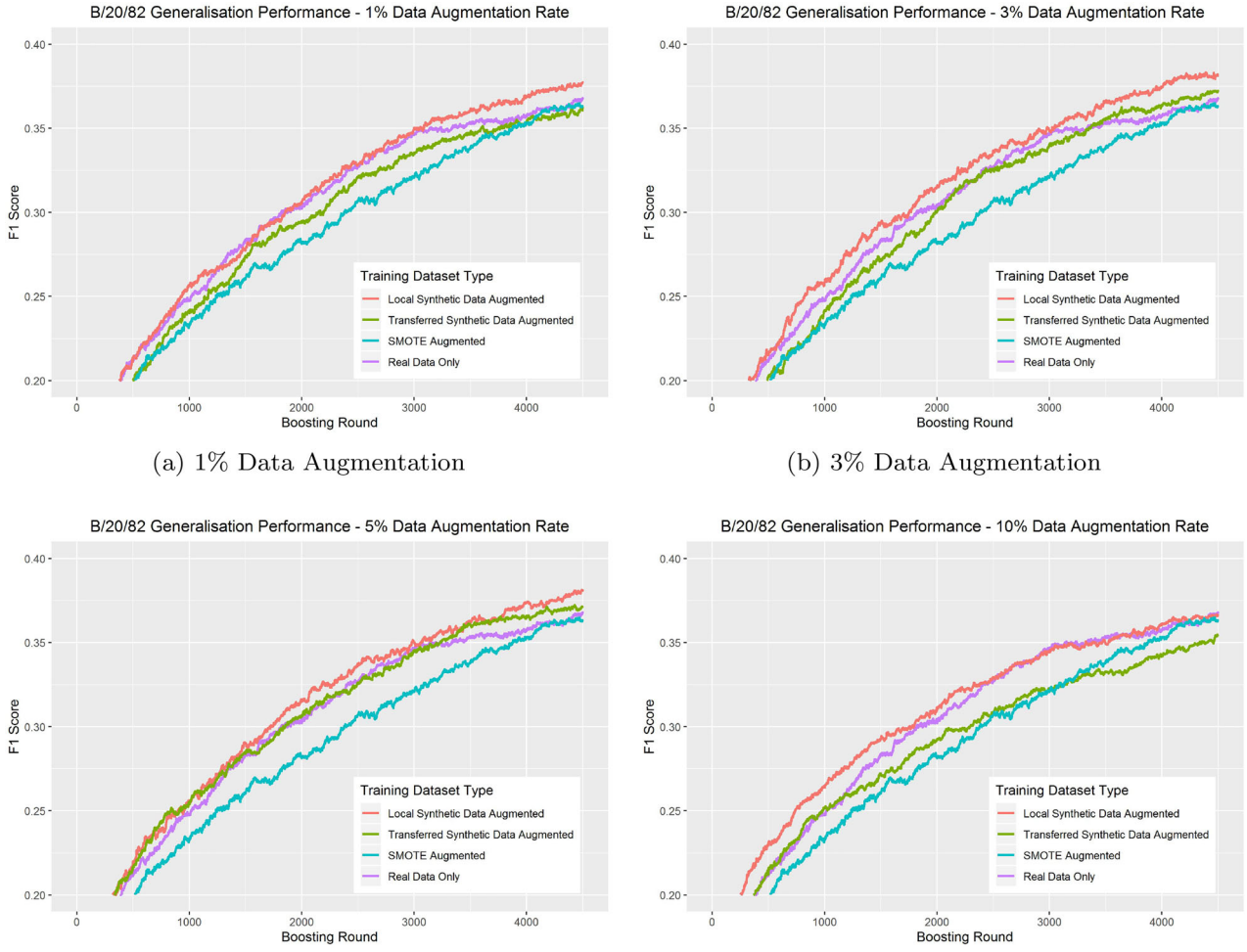
extreme. The validation-best partner GAN model also outperforms the benchmark model at each sampling rate, although by a smaller margin, while the SMOTE-NC model underperforms at each point in the test achieving an F1 score of 0.449 at the 50% sampling rate, decreasing to 0.367 at 99%.

6. Discussion

6.1. General remarks

The main conclusion that we draw from the results is that using GAN-generated synthetic data in small amounts to augment training sets for fraud detection has the potential to improve model performance, although how synthetic data affects performance is sensitive to the underlying customer distributions and the source of the data.

The gains from GAN-based augmentation for the banks which skew towards prime customers is fairly definitive, while for the subprime banks we have some ambiguity as to which model ultimately



(a) 1% Data Augmentation

(b) 3% Data Augmentation

(c) 5% Data Augmentation

(d) 10% Data Augmentation

Figure 20. B/20/82 generalisation performance - experiment 2

performs best. For the mildly subprime A/55/45, arguably the best validation model was obtained by augmenting fraud data only with the partner bank GAN. Both the partner GAN and SMOTE-NC models clearly outperform the baseline model, and so the question is which augmentation method is preferable for this particular distribution. For Bank A/80/18, however, the SMOTE-NC augmentation method is at an obvious disadvantage, and so the choice is between the baseline model and partner GAN model, with the generalisation performance being perhaps less relevant to the decision in the skewed bank scenarios, since it is unlikely that a subprime lender will be competing for customers with a private bank, for instance.

For the subprime banks, the decision to utilise GANs depends on the level of new fraud cases or typologies to which the bank expects to be exposed. Some studies (Tang et al., 2014; Van den Poel & Lariviere, 2004) have found that financial services customers with lower financial and/or social status tend to have higher rates of attrition, or churn. Higher churn for subprime banks, all else being equal, means more chances for fraud at the application stage, and more limited transaction histories with which to identify anomalous spending behaviour.

Another interesting result is how the various banks benefit from GAN augmentation and transfer. Both of the prime banks have a clear performance gain from utilising local GAN-augmented fraud data, while the evidence of performance gains from the partner GAN are at best mixed. Conversely, the best performing validation GAN models for the subprime banks are arguably the partner GAN models. Until further investigation is conducted, we can only provide an educated guess as to the reason for this asymmetry.

Our hypothesis is that the observed model performance with and without synthetic data augmentation is due to a fundamental difference in spending behaviours between customer types. Perhaps due to increased customer churn, we believe that the subprime population exhibits more heterogeneous or diverse spending patterns compared to the prime population, and at least partially encompasses the spending patterns of the prime population. Consider for example a student who leaves university and enters the workforce in a high-paying position. Due to limited or non-existent

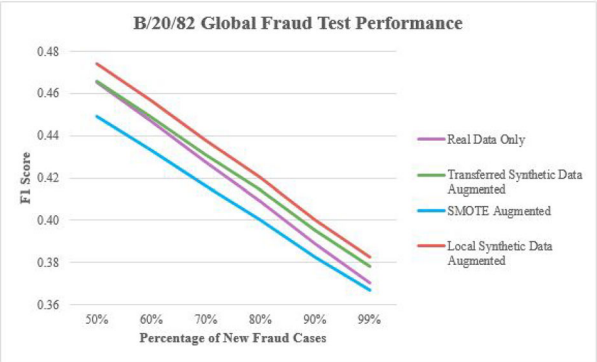


Figure 21. Bank B/20/82 realistic performance - experiment 2.

credit history, he or she may be initially classed as a subprime customer, later becoming a prime customer as his or her credit history develops, without there necessarily being any change in spending patterns.

In the event subprime customers exhibit greater volatility in spending patterns, while partially overlapping with prime customers, we would expect that transferring synthetic samples from Bank B to Bank A would result in a higher ratio of relevant to irrelevant or noisy samples, and vice versa from Bank A to Bank B. This could explain why Bank B sees less utility from Bank A's data – the number of irrelevant or noisy samples causes an impediment to training relative to Bank B's own synthetic samples. In contrast, although the prime bank may produce more homogeneous samples, a greater number fall within the distribution of the subprime bank, which could explain the performance of the validation-best partner GAN model for Bank A/80/18 – a slight boost to local performance with constant or slightly worse performance in terms of generalisation to new cases and distributions, a sign of overfit. This observed result could have also been in part a function of data quality, as the synthetic data validation results in Figure 6(b) show, the best performing fraud detection model trained on only synthetic samples for Bank A/80/18 did relatively worse than for the other banks. This in turn suggests that the local synthetic samples provided to Bank A/80/18 were a less accurate representation of p_{data}, which may also speak to the greater diversity and complexity of the distribution, making it more difficult to model.

Differences in volatility could explain the observed model performance of both the non-augmented model benchmarks, as well as the synthetic data-augmented models. Referring back to Table 1, more diverse or "noisier" spending behaviours would be consistent with greater difficulty in detecting credit card fraud, which is what we observe. In both the similar and skewed distribution scenarios, Bank A's model performs worse on its own customer base than does Bank B, suggesting that the higher credit quality customers exhibit more homogeneous spending patterns where it is easier to detect deviations from those norms. Note as well that the skewed distribution models perform better than the similar distribution models, suggesting that there are some distinct behaviours in either population that are easier to discern in isolation, and confound when the customer mix becomes more evenly split between the two customer groups.

The observed generalisation performance also lends support to our hypothesis. In both distribution scenarios the Bank A benchmark has better generalisation performance on Bank B's customer base than vice versa, a pattern which also mostly holds true for the pooled fraud experiments in the skewed distribution scenario. The fact that Bank B/20/82 sees relatively marginal gains on its local model performance from synthetic data augmentation, but does clearly benefit in terms of generalisation would tend to support this view, to the extent that augmentation increases sample diversity as opposed to simply altering model training behaviour.

In most cases when augmenting training sets with synthetic data, the best results were achieved with augmentation using synthetic fraud data only. When combined with down-sampling this allows for the inclusion of a greater number of real non-fraud observations. In the case of Bank A/55/45, however, the strongest local GAN results occurred with the inclusion of both synthetic non-fraud and fraud data, suggesting there can be circumstances where the inclusion of synthetic data from the majority class is warranted, and so should not be ruled out of the model development process prematurely.

There is a concern as to whether the magnitude of the gains observed from data augmentation with GANs are significant from both a statistical and business investment standpoint. From a statistical perspective, the sheer size of the datasets involved instills a high degree of confidence in the obtained results. Consider the Agnostic Test Set Bound from prediction theory (see e.g. Langford (2005) and the references therein for an overview) which permits us to upper bound the difference, specifically the KL divergence, between the error rate of the test set, and the true, unobservable error rate of a given model with a minimum probability. Since the test set for each bank is in the range of 1.9–2 million samples, these bounds can made remarkably tight. In the case of Bank B/45/55 for example with 1.95 million test set observations, the KL divergence between the observed and true model error rates is upper bounded at 5.9×10^{-6} with probability at least 99.999%. Relaxing the probability threshold

permits an even tighter bound on the error rate difference.

As to whether these gains, although statistically meaningful, are worth the resource investment depends both on the customer distribution, as well as bank-specific fraud characteristics that drive fraud-related losses. Again consider Bank B/55/45, which witnessed a 1.0 point local improvement in F1 score from the GAN-augmented data model compared to both benchmark and SMOTE-augmented data models. Analysing the F1 score components shows this improvement to come from a 13.9% reduction in false positives (relative to benchmark), partially offset by a slight 1.2% decrease in detection of fraudulent transactions, i.e. true positives. In the analysis of fraud losses mentioned in Subsection 4.4.1, the report takes a false positive:true positive ratio of 10:1 as typical in financial fraud detection. Assuming a bank starts with this 10:1 ratio as baseline, then implementing the GAN-augmented fraud model will shift this ratio to 8.61:0.99, or equivalently 8.72:1. Assume for simplicity this bank incurs £100 of fraud-related costs, split roughly equally between direct fraud and false positive costs, and in a given year catches 1 true fraud with 10 false positives. Before implementing the GAN-based model, a false positive costs the bank $£50/10 = £5$ per instance. After implementing the GAN-augmented data model, the bank's direct fraud costs increase to $£50 \cdot 1.012 = £50.63$, while costs linked to false positives decrease to $£5 \cdot 8.72 = £43.59$. The 1.0 point increase in F1 score for this bank then translates to a reduction in fraud-related expenses, from £100 to £94.22 – a 5.8% cost savings. Larger financial institutions can process hundreds of billions of pounds in credit and debit card payments per year, and smaller institutions will still process substantial volumes. Even if fraud-related expenses represent a small fraction of total payment value, given the volumes involved a 5.8% fraud cost reduction can lead to significant savings.

Finally, while not a direct comparison to Chawla et al. (2002) and the results obtained by Fiore et al. (2019), we note that our most definitive benefits from the addition of synthetic data were obtained at augmentation rates of 1–5% and in fact saw considerable deterioration in model performance past that point. In contrast, Fiore et al. (2019) report their strongest F1 score results at a minority class sampling ratio synthetic:real of 2:1, while the best reported sampling ratios in Chawla et al. (2002) ranged from 1:2 up to 5:1, which corresponds to augmentation rates of 33% to 83%. At augmentation rates in this range, we witnessed model performance losses relative to the benchmark model in all cases, with losses ranging from 2.2 to 14.2 points of F1 score for experiment 2. This could be due to several factors including peculiarities of our dataset or experiment design, and we also note that these two studies were experimenting with datasets numbering in the hundreds or thousands of samples, whereas we had access to a training set numbering in the millions. We leave it to future work to explore a potential relationship between dataset size and the effectiveness of data augmentation.

6.2. Limitations & future work

In designing the fraud detection experiments, we sought to examine the effects of synthetic data augmentation and transfer under multiple transfer scenarios and potential customer distributions, to be able to condition the results on the fact that each financial institution has a unique customer mix, as well as attain a degree of generality in our conclusions. Despite these efforts, there may be some peculiarities to our dataset which affect the generality of our results. In order to confirm or disprove our findings and hypotheses, in particular surrounding the behaviours of subprime and prime customers, replication across additional datasets is required, although to the best of our knowledge no publicly available credit card fraud datasets of similar size exist – as mentioned in the previous subsection, our conclusions regarding the optimal level of augmentation may be sensitive to dataset size.

Additionally, while we sought to give as broad a consideration as possible to alternative procedures at each stage in the experiments, time and resource constraints limited the scope of our work. For example, we attempted to exhaustively consider the potential data augmentation scenarios by including multiple hypothetical institutions with varying distributions, multiple rates of data augmentation of both fraud and non-fraud data, as well as a full fraud classifier model selection procedure for both the local and transfer augmentation cases. Had we been able to narrow the focus beforehand to a subset of these scenarios, a more in-depth hyperparameter search in the distribution modelling and fraud classifier validation stages may have affected the results. In our results we had also discovered that certain customer distributions (Bank A/55/45) could potentially benefit from synthetic data augmentation from different sources, i.e. local GAN non-fraud samples and partner GAN fraud samples – an experiment we did not consider in this study which presents a possible avenue for future research. Our study design also specifically controlled for differences in features between the hypothetical banks, in order to isolate the effects of distributional differences on transfer learning with synthetic data. Based on our findings, in future work we plan to relax

this assumption of homogeneous feature sets and develop techniques for feature transfer while controlling for distributional differences, to further explore the potential gains from synthetic data transfer.

Also, while we were able to establish that our proposed data transformation and GAN transfer framework does yield results that are an improvement to some alternative procedures, for similar reasons as above we were not able to give consideration to all published alternatives. In future work, we plan to evaluate our proposed minority class GAN training methods and data transformations against a wider range of alternatives, for example cGANs and autoencoders. In particular, cGANs have been investigated in imbalanced data settings (Douzas & Bacao, 2018; Fiore et al., 2019), although with datasets much smaller in scale than our case study. In Subsection 4.3, we hypothesised that sample size could play a role in the effectiveness of cGANs for imbalanced data modelling, and this would be the area of focus in future method comparison research.

As was mentioned in Section 5, while our proposed data transformations did offer improved performance compared to some alternative GAN generator architectures, the generator did appear to struggle in modelling log transformed variables in some circumstances, in particular where the univariate histograms display a long tail of extreme values. We suspect that this was due in part to the bounded variables being sampled, and hence error functions calculated, in log space which will reduce the impact of extreme positive observations on the resulting gradient calculations. We plan to examine alternative GAN architectures in future work that better account for this fact, by adding an inverse transformation to the generator output layer for example, or by altering the sampling of the noise distribution such that the log and logit transformations are included in the computational graph of the generator.

Lastly, we mentioned in Section 1 that there is the possibility to combine synthetic data generation with privacy mechanisms such as differential privacy. In the distribution modelling stage we did initially attempt to train a GAN under the stricter ϵ-differential privacy, although did not obtain any usable models. In future work, we plan to examine alternative differential privacy definitions and extensions, including those which seek to address the issue of differential privacy in the presence of correlated data (Kifer & Machanavajjhala, 2011).

Acknowledgements

The authors acknowledge Research Computing at The University of Virginia for providing computational resources and technical support that have contributed to the results reported within this publication. URL: https://rc.virginia.edu

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This material is based upon work supported by the National Science Foundation under Grant No. CNS: 1650512. This work was conducted in the Center for Visual and Decision Informatics, a National Science Foundation Industry/University Cooperative Research Center.

ORCID

Stephen Adams http://orcid.org/0000-0002-1207-4504
Peter Beling http://orcid.org/0000-0003-2196-6982

References

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2015). Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and Its Applications*, *7*(3), 177–203.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *Proceedings of the 34th International Conference on Machine Learning* (vol. 70, pp. 214–223). Sydney, Australia: PMLR.

Beaulieu-Jones, B. K., Wu, Z. S., Williams, C., Lee, R., Bhavnani, S. P., Byrd, J. B., & Greene, C. S. (2019). Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, *12*(7), e005122. https://doi.org/10.1161/CIRCOUTCOMES.118.005122

Camino, R., Hammerschmidt, C., & State, R. (2018). Generating multi-categorical samples with generative adversarial networks. *Proceedings of the 2018 ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*. Stockholm, Sweden: PMLR.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). Xgboost: Extreme gradient boosting. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA (pp. 785–794). New York, NY, USA: Association for Computing Machinery. .

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., & Sun, J. (2017). Generating multi-label discrete patient records using generative adversarial networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace and Jenna Wiens (Eds.), *Proceedings of the 2nd Machine Learning for Healthcare Conference*, 18–19 Aug, Boston, Massachusetts (vol. 68, pp. 286–305). PMLR.

Cody, T., Adams, S., & Beling, P. A. (2018). *A utilitarian approach to adversarial learning in credit card fraud detection* [Paper presentation]. 2018 Systems and Information Engineering Design Symposium (SIEDS) (pp. 237–242). Charlottesville, VA, USA: IEEE. https://doi.org/10.1109/SIEDS.2018.8374743

Crossfield, J. L., & Griffin, A. (2017). *Card fraud costs to banks increase to $40 billion*. (Tech. Rep.). https://www.feature-space.com/wp-content/uploads/Cost-of-Card-Fraud-to-Banks-January-2017.pdf

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240). Pittsburgh, PA, USA: Association for Computing Machinery. https://doi.org/10.1145/1143844.1143874

Douzas, G., & Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications*, *91*, 464–471. https://doi.org/10.1016/j.eswa.2017.09.030

Dwork, C. (2011). Differential privacy. In *Encyclopedia of cryptography and security* (pp. 338–340). US: Springer.

Federal Trade Commission. (2019). *The consumer sentinel network data book 2018*. (Tech. Rep.). https://www.ftc.gov/enforcement/consumer-sentinel-network/reports

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, *479*, 448–455. https://doi.org/10.1016/j.ins.2017.12.030

Flamary, R., & Courty, N. (2017). *POT: Python optimal transport library*. https://pythonot.github.io/

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, Montreal, Canada, Vol. 27, 2672–2680. Curran Associates, Inc.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, Long Beach, California, USA, Vol. 30, 5767–5777. Curran Associates Inc.

Hittmeir, M., Ekelhart, A., & Mayer, R. (2019). On the utility of synthetic data: An empirical evaluation on machine learning tasks. *Proceedings of the 14th International Conference on Availability, Reliability and Security*. Canterbury, United Kingdom: Association for Computing Machinery. https://doi.org/10.1145/3339252.3339281

Hjelm, R. D., Jacob, A. P., Trischler, A., Che, G., Cho, K., & Bengio, Y. (2018). Boundary seeking GANs. *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada. OpenReview.net. https://openreview.net/forum?id=rkTS8lZAb

HSN Consultants Inc. (2019). *The Nilson Report*. (Tech. Rep.). https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1164.pdf

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel-Softmax. *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

Jordon, J., Yoon, J., & M. van der, S. (2019). *PATE-GAN: Generating synthetic data with differential privacy guarantees* [Paper presentation]. 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA. OpenReview.net. https://openreview.net/forum?id=S1zk9iRqF7

Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, *52*(4), Article No. 79. https://doi.org/10.1145/3343440.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 3149–3157). Curran Associates Inc.

Kifer, D., & Machanavajjhala, A. (2011). No free lunch in data privacy. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (pp. 193–204). Athens, Greece: Association for Computing Machinery. https://doi.org/10.1145/1989323.1989345

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014*, Banff, AB, Canada. OpenReview.net. http://arxiv.org/abs/1312.6114

Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y., & Zhou, X. (2020). Generative adversarial networks and its applications in biomedical informatics. *Frontiers in Public Health*, *8*, 164. https://doi.org/10.3389/fpubh.2020.00164

Langford, J. (2005). Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, *6*, 273–306.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, *5*(1), 42. https://doi.org/10.1186/s40537-018-0151-6

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1–5. http://jmlr.org/papers/v18/16-365.html

Li, H., Xiong, L., & Jiang, X. (2014). Differentially private synthesization of multi-dimensional data using copula functions. *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014* (pp. 475–486), Athens, Greece. OpenProceedings.org. https://doi.org/10.5441/002/edbt.2014.43

Liu, X., Zou, Y., Kong, L., Diao, Z., Yan, J., Wang, J., Li, S., Jia, P., & You, J. (2018). *Data augmentation via latent space interpolation for image classification* [Paper presentation]. 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 728–733). Beijing, China: IEEE. https://doi.org/10.1109/ICPR.2018.8545506

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. Y. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017* (pp. 1273–1282), Fort Lauderdale, FL, USA: PMLR. http://proceedings.mlr.press/v54/mcmahan17a.html

Mead, A., Lewris, T., Prasanth, S., Adams, S., Alonzi, P., & Beling, P. (2018). *Detecting fraud in adversarial environments: A reinforcement learning approach* [Paper presentation]. 2018 Systems and Information Engineering Design Symposium (SIEDS) (pp. 118–122). Charlottesville, VA, USA: IEEE. https://doi.org/10.1109/SIEDS.2018.8374720

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

R Core Team. (2018). *R: A language and environment for statistical computing [Computer software manual]*. https://www.R-project.org/

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: A review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4), 1–29. http://www.researchmanuscripts.com/ijcbr/index.php/vol-5-issue-4-july-2014

Shao, S., Wang, P., & Yan, R. (2019). Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry*, *106*, 85–93. https://doi.org/10.1016/j.compind.2019.01.001

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1–42. https://doi.org/10.1186/s40537-019-0197-0

Sun, Y., Cuesta-Infante, A., & Veeramachaneni, K. (2019–February 1). *Learning vine copula models for synthetic data generation* [Paper presentation]. The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019 (pp. 5049–5057), Honolulu, Hawaii, USA: AAAI Press. https://doi.org/10.1609/aaai.v33i01.33015049

Tang, L., Thomas, L., Fletcher, M., Pan, J., & Marshall, A. (2014). Assessing the impact of derived behavior information on customer attrition in the financial service industry. *European Journal of Operational Research*, *236*(2), 624–633. https://doi.org/10.1016/j.ejor.2014.01.004

UK Finance. (2019). *Fraud the facts 2019: The definitive overview of payment industry fraud*. (Tech. Rep.). https://www.ukfinance.org.uk

Van den Poel, D., & Lariviere, B. (2004). Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, *157*(1), 196–217. https://doi.org/10.1016/S0377-2217(03)00069-9

Wang, Y., Adams, S., Beling, P., Greenspan, S., Rajagopalan, S., Velez-Rojas, M., Mankovski, S., Boker, S., & Brown, D. (2018). Privacy preserving distributed deep learning and its application in credit card fraud detection. *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)* (pp. 1070–1078). New York, NY, USA: IEEE.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag. https://ggplot2.tidyverse.org

Zeager, M. F., Sridhar, A., Fogal, N., Adams, S., Brown, D. E., & Beling, P. A. (2017). *Adversarial learning in credit card fraud detection* [Paper presentation]. 2017 Systems and Information Engineering Design Symposium (SIEDS) (pp. 112–116). Charlottesville, VA, USA: IEEE. https://doi.org/10.1109/SIEDS.2017.7937699

Zhang, J., Wang, J., Zhao, Y., & Chen, B. (2019). An efficient federated learning scheme with differential privacy in mobile edge computing. In X. B. Zhai, B. Chen, & K. Zhu (Eds.), *Machine learning and intelligent communications* (pp. 538–550). Nanjing, China: Springer International Publishing.

Appendix A

Experimental details

In terms of computing environment for the experiments, feature selection, experiments with the "xgboost" algorithm (Chen et al., 2016), and most data visualisations (Wickham, 2016) were conducted in R (R Core Team, 2018), while the remaining experimental steps were performed in Python 3.6/3.7, primarily using the University of Virginia's high-performance computing cluster.

Of the 50 features selected for the experiments, the categorical variables were day of the week (7 features) and month of the year (8 features), as well as the merchant category code (15 features). Numeric features included account credit limit, account-level information on balances outstanding and authorised transactions, total number of transactions on the card, length of time since the credit card was issued, transaction time, and distance of the transaction from the cardholder's home ZIP code, along with several derived features. Examples of the derived features include the length of time between a card holder's transactions, and the proportion of the card holder's transactions that fall within a given hour of day, day of the week, etc.

Feature selection was conducted on a hold out set of 50,000 random samples – 5000 fraud transactions and 45,000 non-fraud transactions – using the feature importances obtained from an xgboost gradient boosted decision tree model. Feature selection was performed in an iterative fashion, training a model then culling the least important feature before re-training.

Regarding the proposed bounded feature transformations, where these features had values at the boundaries, meaning the log and logit functions are undefined, random noise was added such that the values fell in between the boundary and the next lowest (or highest) value in the dataset. The GAN models were trained on these noisy values, while for model evaluation and fraud detection experiments, any values produced within these margins were rounded to the boundary.

In Subsection 4.2, it was mentioned that sampling was adjusted to equalise expected fraud instance between the partner banks. This was done to control for the possibility of differences in GAN quality between the banks being caused by differences in fraud data availability. The fraud cases were fairly evenly distributed across credit limits, meaning little over/under sampling was required to generate an approximately balanced division of the fraud cases among the banks. In each scenario, both banks were left with $36-38$ million transactions, including circa 50,000 fraudulent ones. During the fraud classifier model selection phase, 2500 fraud samples were assigned to the two validation sets, and merged with random samplings of non-fraud transactions such that the fraud rates matched the test set.

During GAN training and architecture selection, similar to Camino et al. (2018) categorical features were branched off separately for an additional $0-2$ layers, before applying a Gumbel – Softmax activation and concatenating the output to the generated continuous features. Included in the hyperparameter tuning was the testing of different data transformations for the bounded variables. In addition to the proposed transformations, for scaled variables that were both upper and lower bounded a sigmoid activation of the generator output layer was tested, as well as threshold function activations – essentially ReLU activations with arbitrary threshold(s) – for all bounded features.

As for the GAN validation metrics, due to the computational complexity of the calculation, EMD distance was not calculated on the entire validation set, but rather on random subsets of the validation data and then averaged. Through trial and error it was found that averaging 40 subsamples of size 2000 provided reasonably stable distance estimates with an acceptable computation time.

Once the data augmentation stage of the experiments were reached, initial tests were conducted with various machine learning algorithms for fraud classifier training including xgboost, LGBM, logistic regression, K-Nearest Neighbours, and artifical neural networks, with LGBM models showing the most promise in terms of F1 score.

Appendix B

Key GAN hyperparameters

Table B1. Key hyperparameters across non-fraud and transferred fraud GANs.

Hyperparameter	Bank A/55/45	Bank B/45/55	Bank A/80/18	Bank B/20/82
GAN Training Framework	WGAN-GP	WGAN-GP	WGAN-GP	WGAN-GP
Optimiser	Adam	Adam	Adam	Adam
Fraud Weight Re-initialisations - Generator	None	None	None	None
Fraud Weight Re-initialisations - Critic	None	None	None	Final Layer
Noise Distribution	N(0,1)	Unif(−1,1)	N(0,1)	Unif(−1,1)
Nodes per Hidden Layer - Generator	[171,63,128,186]	[50,50]	[50,50,50]	[124,124]
Residual (Shortcut) Connections	Y	Y	Y	Y
Generator Activations	ReLU	Tanh	ReLU	ReLU
Additional Categorical Hidden Layers	2	0	0	2
Categorical Layer Activations	Tanh	ReLU	Softsign	Tanh
Generator Batch Norm	N	Y	Y	Y
Nodes per Hidden Layer - Critic	[96]	[83,134]	[186,168,178,38,56]	[143,184,147,145,33,59]
Critic Activations	Softsign	Leaky ReLU	Softsign	Sigmoid
Critic Batch Norm	N	N	N	N