Semi-supervised GANs for Fraud Detection *

Charitos Charitou
Department of Computer Science
City, University of London
London, UK
charitos.charitou@city.ac.uk

Artur d'Avila Garcez
Department of Computer Science
City, University of London
London, UK
a.garcez@city.ac.uk

Simo Dragicevic
BetBuddy
Playtech Plc
London, UK
simo.dragicevic@playtech.com

***Abstract*—Over the years the online gambling industry has evolved into one of the most profitable industries on the Internet. At the same time, new stringent regulations have required the online industry to become a lot more vigilant. Although standards have improved, the methods used to process finance from illicit activities also evolved and became more sophisticated. Detecting these fraudulent activities in real life with high accuracy requires a learning system to be trained with balanced data sets of fraudulent and normal transactions. However, in the real-world, the number of fraudulent cases is significantly lower than normal cases. In this paper, to deal with data imbalance, we propose a novel generative adversarial framework based on semi-supervised learning of sparse auto-encoders for the detection of fraud in online gambling. Experimental results show that the proposed framework outperforms mainstream discriminative techniques such as logistic regression, random forest and multi-layer perceptron. We validate further the approach by applying it to other domains that suffer from the problem of class imbalance obtaining promising results.**

***Index Terms*—Fraud detection, Imbalanced data, Semi-supervised Generative Adversarial Networks, Sparse Auto-encoders.**

I. Introduction

Fraud detection refers to the identification of illegal activities occurring in numerous industries such as finance, gambling, insurance or cybersecurity. If fraudulent behaviour is not monitored and prevented then it can have catastrophic consequences such as the financing of terrorism. Many organizations have been interested in the immediate detection of illicit activities, aiming to prevent losses, while also ensuring the safety of their customers [1].

This research is part of a collaboration with a major gambling operator. The purpose of the research is to explore the use of deep learning to strengthen processes used in the detection of suspicious gambling behaviour, in particular money laundering. In the UK, gambling firms have paid over £40 million in fines and settlements since 2017 with all major cases involving failings in detecting money laundering.

Until recently, the gambling industry has tackled the identification of money laundering in online gambling primarily by using knowledge-based systems. Whilst capable of easily embedding regulatory requirements which have focused on simple thresholds, these systems are unable to adapt to new requirements to proactively monitor the activity of millions of online customers and a changing malicious behaviour related to criminal activity online.

In fraud detection problems, the fraudulent cases tend to be far fewer than the non-fraudulent ones (referred to in the literature as an 'imbalanced data set'), which leads to difficulties in the training of classification algorithms. In most cases, such algorithms seek to maximize accuracy and as a result become biased towards the majority class.

Classification models, such as logistic regression (LR), random forest (RF), multi-layer perceptron (MLP), are typically discriminative models, i.e. via the use of a certain feature set, they try to select the most appropriate class. This is, essentially, the root cause of the problem of the bias caused by the data imbalance, as the algorithm does not have a notion of 'how' the data are produced, yet it focuses on the objective measure of discrimination (e.g. accuracy). A way of alleviating this problem is to use models that aim to also understand the underlying generative process, as done for example by generative networks. Gaussian Mixture Models (GMMs) have formed the backbone of a variety of generative models, including Hidden Markov Models, employed with this objective [2], yet they come with Gaussian distribution assumptions and require much effort to be deployed in classification problems. Such models have been used together with clustering techniques to provide the required classification algorithm [3].

Recently, Generative Adversarial Networks (GANs) allowed for a more generic approach with the advantages of combining end-to-end both generative and discriminative techniques. By extending the traditional framework of GANs to allow for the discriminator to perform classification [4], semi-supervised GANs (SSGANs) have shown potential in the recent literature particularly at learning from unstructured data such as images or sound [5]. Nevertheless, research regarding the application of GANs to structured data has been very limited.

In this paper, we argue that semi-supervised GANs can provide a powerful and versatile framework for tackling supervised learning from imbalanced and sparse structured data. We validate this claim empirically by applying SSGANs to different domains suffering from the same data imbalance difficulty. We conduct experiments on the benchmark data sets for Credit Card Fraud, Breast Cancer Wisconsin and Pima Diabetes. Finally, we apply the proposed semi-supervised framework on a real-world Gambling Fraud Detection data set which is related with money laundering. We compare

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

our results with those of classical discriminative techniques, namely random forest, logistic regression and multi-layer perceptron, trained in conjunction with the synthetic minority oversampling technique (SMOTE) [6] and adaptive synthetic sampling technique (ADASYN) [7]. The results show that our framework outperforms the other models even when these are combined with elaborate oversampling methods such as SMOTE and ADASYN. We also note that during training, our model conveniently produces a generator for the production of synthetic data for this type of problem. This is useful at creating simulations which may be essential for the development and testing of automated systems such as a fraud detection system.

More specifically, in this paper we introduce a system architecture based on semi-supervised generative adversarial networks and sparse auto-encoders (SAE) and we apply it to a fraud detection system and other classification tasks with imbalanced data. During the training phase our approach is divided into two parts: first, the data are encoded into a latent representation (vector space) using the sparse auto-encoder. Then, that feature representation extracted from the auto-encoder is used to train the semi-supervised GAN. The contributions of this work are summarized as follows:

- We propose a new architecture for imbalanced data classification which does not require oversampling techniques to produce good classification results.
- Our results on the benchmark data sets are promising; our method outperforms logistic regression, random forest and multi-layer perceptron with improvements on F1 score for all the data sets that were examined.
- We apply the proposed architecture to a real-world problem of money laundering in online gambling, obtaining better classification results than an existing anti-money laundering detection system. The F1 score is improved by 3.64%.

The remainder of this paper is organised as follows: Section 2 discusses the related work. Section 3 describes the proposed semi-supervised GAN model in detail. Section 4 presents the experimental results. Section 5 discusses the application of the model to money laundering detection in gambling. Section 6 concludes the paper and discusses directions for future work.

II. Related Work

A. Fraud Detection and Imbalanced Data Classification

In recent years, there has been a considerable research effort at handling imbalanced data classification for fraud detection. In [8] the authors combined SMOTE and under-sampling to solve the class imbalance problem. Wu, Shen and Zhang [9] developed a fuzzy multi-class support vector machine algorithm for imbalanced data. Shukla and Bhownick [10] used K-Means algorithm to balance an imbalanced data set and then use SVM to classify that data set. In [11], the authors proposed different techniques to enhance the classification performance of random forest and logistic regression when dealing with imbalanced data sets. A more generative approach is followed by [12], describing a Gaussian Mixture under-sampling technique in order to solve the class imbalance problem that exists in many real world applications.

Both supervised and unsupervised techniques have been examined to address fraud related problems. Niu, Wang and Yang [13] conduct a comparison study for credit card fraud detection by evaluating ten machine learning algorithms both supervised and unsupervised. The research in [14] describes and reviews the challenges and different techniques and evaluation criteria that can be used in the mitigation of credit card fraud.

Zareapoor and Shamsolmoali [15] proposed a bagging classifier based on decision trees for the construction of a fraud detection model. In [16], the authors use a Convolutional Neural Network (CNN) to capture important patterns of fraud behaviour. Saraswathi, Kulkarni, Khali and Nigam [17] developed a clustering mechanism based on Self-Organizing Maps (SOM) for the detection of credit card extortion activities. Srivastava, Kundu, Sular and Majumdar [18] utilized Hidden Markov Models (HMM) to model the sequence of operations in credit card transaction processing and fraud detection. Perhaps closest to our model, Chen, Shen and Ali [19] combine a sparse auto-encoder with one-class adversarial networks to classify credit card transactions as fraudulent.

B. GANs for classification

Several studies have incorporated the concept of adversarial training for semi-supervised learning. Salimans et al. [4] introduced empirical techniques such as feature matching for stabilizing the training process of GANs. In [20], the author proposes categorical generative adversarial networks (Cat-GAN) to substitute the binary discriminator in standard GANs with a categorical classifier. Research in [5] proposed a bad generator in order to improve the classification results of semi-supervised GANs. The authors also showed that good classification performance and a good generator cannot be obtained together. Based on the work from [5], the study in [21] introduces One-Class Adversarial Nets for fraud detection. In their approach, the generator of the complementary GAN can generate benign user representations, while the discriminator is trained to distinguish the real and complementary benign users. Finally, Zhou, Liu, C. Zhou and Chen [22] investigated the application of SSGANs on structured imbalanced data sets.

III. SSGAN for fraud detection

A. Framework Description

The structure of the proposed framework is illustrated in Fig. 1. It consists of two parts: a sparse auto-encoder and a complementary generative adversarial network. In this architecture, the sparse auto-encoder includes two encoding layers and two decoding layers. During the encoding phase the input data are projected into a higher dimension, while in the decoding phase the network tries to reconstruct the input data from the sparse representations of the data. Mapping the data onto a higher dimension during encoding seeks to increase the distance between positive and negatives samples as Fig. 2a and Fig. 2b illustrate.

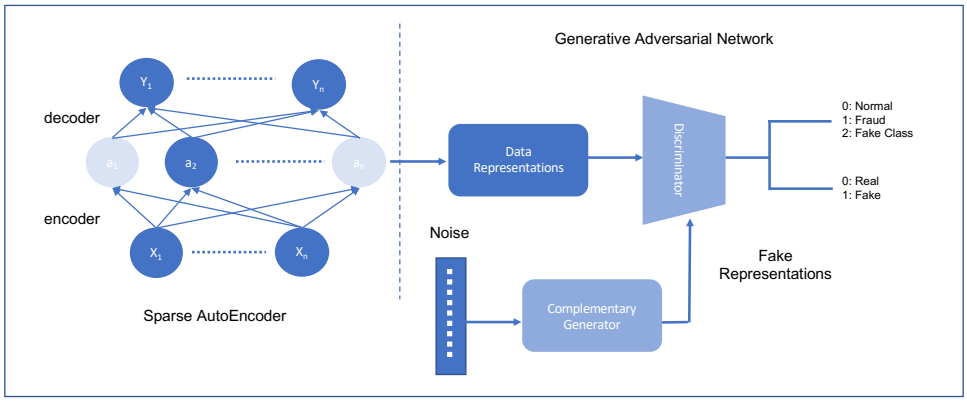


Fig. 1: Architecture of the proposed system (SSGAN) showing (on the left) a sparse auto-encoder mapping the data onto a higher-dimensional vector space. The output of the encoder is used as input to the generative adversarial network (on the right). After training, the discriminator of the GAN is able to classify the data as fraud or normal.

The data representations extracted from the SAE are used as input to the generative adversarial network. Our GAN adopts a complementary generator which tries to match the data representations from a Gaussian random noise in order to generate new complementary samples. Together with the real representations the generated samples are used to train a discriminator model. After training is complete, the discriminator is used to distinguish and detect the fraudulent cases.

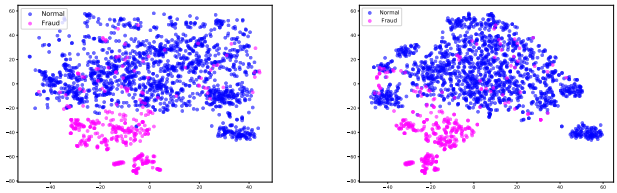
B. Sparse Auto-encoders for Latent Representation

The framework's sparse auto-encoder consists of a feedforward neural network whose hidden layer is larger than the size of the input layer and whose target output is by definition equal to the input vector [23]. The output of the hidden layer within the auto-encoder represents the encoding of the input x into a sparse latent feature representation. This type of neural network tries to learn a function $h_{W,b}(x) \approx x$ in order to reproduce an output x' that is similar to x [24].

Extending the idea of the original auto-encoder, a sparse auto-encoder incorporates to the reconstruction error a sparse penalty term $\Omega(h)$ w.r.t. the hidden layer h [25] [26]. This penalty on the activations of the units of a neural network seeks to make the representation sparse with the objective of producing more robust and generalized features [27]. The sparsity term can be imposed on the output layer of the encoder or on a hidden layer or bottleneck. In our sparse auto-encoder, we applied the L1 regularization which enforces sparsity by allowing some activations to become zero. The loss function of a sparse auto-encoder is defined in (1):

$$L(x, g(f(x))) + \Omega(h) \tag{1}$$

where $g(h)$ is the output of the decoder and $h = f(x)$ is the output of the encoder. The penalty term $\Omega(h)$ can be further expressed as $\Omega(h) = \lambda \sum_i \left| a_i^{(h)} \right|$. The loss function penalizes the absolute value of the vector of activation functions a in the hidden layer for an observation i, scaled by a tuning parameter λ.



(a) Original data (b) Representation data

Fig. 2: Fig. 2a and Fig. 2b show the original and representation training data distribution for the Credit Card Fraud data set in 2D space using t-SNE.

The choice of an sparse auto-encoder over the original auto-encoder is supported by [25]. In that paper, the authors suggest that using a sparse auto-encoder enables robust feature extraction from the input. In addition, projecting the data to higher dimensional spaces is more likely to result in an easier classification task [28]. In this paper, the data representations extracted from the hidden layers of the auto-encoder are denoted by $\tilde{x}$.

C. Theoretical analysis of GANs

Generative adversarial networks are generative models based on a game theoretic scenario in which a generator (G) network is competing against a discriminator (D) [29]. The generator having as input a noise variable Z, generates fake samples with distribution p_g which matches the true data distribution $p(data)$. On the other hand, the discriminator network is trained to distinguish the real samples (drawn form the training data) and fake samples generated from G.

Typically, the discriminative model D is trained to maximize its ability to distinguish the real input data from the fake data. The generator tries to fool the discriminator by producing better fake samples. Mathematically, the generator and discriminator play a min-max two player game with value function V(G,D) [29]:

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}}[log(D(x))] + E_{z\sim p_z}[1 - log(D(G(z)))] \quad (2)$$

where E is the expectation, $p(data)$ is the real data distribution and $p(z)$ is a noise distribution. The training of a generative adversarial network could be characterised as an optimization process for both generator and discriminator. The output of the generator is defined as p_g. As Eq. (2) suggests, GANs aim to minimize the Jensen–Shannon divergence between the data distribution p_{data} and the generative distribution p_g with perfect minimization reached when $p_g = p_{data}$. The optimization equations for the generator and the discriminator are defined respectively as:

$$\min_G E_{z\sim p_z}[1 - log(D(G(z)))] \quad (3)$$

$$\max_D E_{x\sim p_{data}}[log(D(x))] + E_{z\sim p_z}[1 - log(D(G(z)))] \quad (4)$$

Although GANs are very promising for new data generation, due to the vanishing gradient problem, training GANs could be really unstable. However, this can be improved when the model architecture and hyper-parameters are carefully selected [30].

GANs can be extended to semi-supervised learning by defining another output in the discriminator. The first output of the discriminator only classifies data as real or fake, while the second output classifies the data by the class that they belong. The idea is that whether the data are real or fake, the classifier has to determine whether it can be classified into the true classes. if it can then the data are probably real.

D. Training the complementary Generator of SSGAN

Inspired by the work of [21] and [5], we implement a complementary generator. Our generator is a two layer feed forward neural network that tries to learn the distribution of the representations (output of the encoder) and not the actual data distribution. The new generated samples have the same dimension as the latent representations and are defined by $n = G(z)$.

By following the approach of [5], the complementary generator with output p_g tries to learn the distribution $p^*(n)$ which is defined as:

$$p^*(n) = \begin{cases} \frac{1}{r} * \frac{1}{p(n)} & if\ p(n) > \tau\ and\ n \in B_{\tilde{x}} \\ C & if\ p(n) \leq \tau\ and\ n \in B_{\tilde{x}} \end{cases} \quad (5)$$

where r is a normalization term, and $B_{\tilde{x}}$ is the feature space of the extracted feature representations, C is a constant and τ is the threshold for separating low and high density data. As a result, the generator now is trained in order to converge its distribution (p_g) to the new complementary distribution p^*. Using the definition of the KL divergence:

$$KL_{(p_g \ ||\ p_g^*)} = -H(p_g) + E_{n\sim p_g} log\ p(n)\ \mathbb{I}[p(n) > \tau] + E_{n\sim p_g}(\mathbb{I}[p(n) > \tau]\ log\ r - \mathbb{I}[p(n) \leq \tau]\ log\ C) \quad (6)$$

In the above equation, $\mathbb{I}$ denotes the indicator function and H the entropy function. As it is stressed by [5] the final term of (6) would not add any further information and can be ignored. The generator also adapts the feature matching loss [4] in order to bring the generated representations closer to the real representations. The final objective function of the complementary generator is the following:

$$\min_G\ -H(p_g) + E_{n\sim p_g} log\ p_{data}(n)\mathbb{I}[p_{data}(n) > \tau] + \left\| E_{n\sim p_g f(n)} - E_{\tilde{x}\sim p_{data}} f(\tilde{x}) \right\|_2^2 \quad (7)$$

where the last term of (7) describes the feature matching loss and $f(\tilde{x})$ is defined as the output of the hidden layer in the discriminator.

One of the main disadvantages of training a generative adversarial network is the *mode collapse* scenario. It is also known as the problem that occurs when the generator learns to map several different input z values to the same output point [31]. This problem is directly related to the entropy distribution of generated features and is a sign of low entropy. Therefore, to improve further our generator, from [21] [5], we adopted a pulling away term (PT) which was introduced in [32] to increase the generator's entropy, defined as:

$$L_{PT} = \frac{1}{N-1} \sum_i^N \sum_{j\neq i}^N \left(\frac{f(n_i)^T f(n_j)}{\|f(n_i)\|\,\|f(n_j)\|}\right)^2 \quad (8)$$

where N is the size of the mini-batch and $f(n)$ is the output of the hidden layer of the discriminator.

E. Training the Discriminator

Following the architecture of [4], the output of the discriminator is mapped onto a softmax classifier. Assuming that there are K possible classes in the data, semi-supervised learning is performed by including the new (fake) samples from the generator in our data and labeling them with a new class K+1. The dimension of the discriminator output is increased to K+1. Moreover, an additional output is added to the soft-max classifier in order to distinguish the real and fake samples. Our discriminator loss function can be described as follows:

$$L = E_{\tilde{x},y\sim p_{data}(\tilde{x},y)}\ [log\ p_{model}(y|\tilde{x})] + E_{\tilde{x}\sim G}\ [log\ p_{model}(y = K+1|\tilde{x})] \quad (9)$$

where $p_{model}(y = K+1|\tilde{x})$ is defined as the probability that x is fake and $p_{model}(y|\tilde{x})$ as the probability that x belongs to a real class. The loss function in (9) is divided into supervised loss $L_{supervised}$ and unsupervised loss $L_{unsupervised}$:

$$L_{supervised} = E_{\tilde{x},y\sim p(\tilde{x},y)}\ [log\ p_{model}(y|\tilde{x}, y < K+1)] \quad (10)$$

$$L_{Unsupervised} = E_{x\sim p_{data}(\tilde{x})}\ [1 - log\ p_{model}(y = K + 1|\tilde{x})] + E_{\tilde{x}\sim G}[log\ p_{model}(y = K + 1|\tilde{x})] \quad (11)$$

where $L_{supervised}$ is the typical supervised loss and $L_{unsupervised}$ is the loss generated from the GAN.

A main contribution of this work is to highlight the classification ability of GANs in supervised learning tasks on structured data, including the imbalanced class problem, as discussed in the next section.

IV. Experimental Results

This research performs three different sets of experiments: (1) We compare the SSGAN framework with three popular classification machine learning algorithms, namely logistic regression (LR), random forest (RF) and multi-layer perceptron (MLP). We trained these methods with the imbalanced data sets of Breast Cancer and Diabetes. In addition, we generate results when these three algorithms are combined with oversampling techniques. (2) We investigate the effect of the sparse auto-encoder on the results by training the benchmark algorithms and our framework with the original data of Credit Card Fraud and with the latent representations. We also compare our method with a semi-supervised GAN trained with a regular generator. (3) We apply our framework to real-world data on Gambling Fraud and we demonstrate the value of the framework in that application domain with a comparison of results.

A. Data sets

Table I shows the three imbalanced data sets that are selected for the evaluation of the proposed method. The Breast Cancer data set includes 569 samples: 212 benign and 357 malignant samples. The Diabetes data set contains 768 patient samples, 268 of which have diabetes and 500 which do not. Finally, the Credit Card Fraud data set consists of 2,492 transactions: 2000 normal transactions and 492 fraudulent transactions.

TABLE I: Experimental Data sets

Data set	Sample Size	Positive samples	Negative Samples	Imbalance Ratio
Breast Cancer	569	212	357	1:1.68
Diabetes	768	268	500	1:1.87
Credit Card Fraud	2,492	492	2,000	1:4.07

In this research the minority class is described as positive while the majority class as negative. This scheme is consistent across all experiments for all data sets. In the classification of imbalanced data, the influence of the minority class on the accuracy is significantly smaller than that of the majority class. Due to the bias towards the majority class, using accuracy as the main criterion could lead to a low minority class identification rate. Therefore, alongside accuracy, we use other evaluation criteria, such as precision, recall and F1 score. The presented results illustrate the mean value and standard deviation for accuracy, recall, precision and F1 score on 10 different runs. In all the experiments the training and testing set ratio is set to 80% and 20% respectively.

B. Results and Comparison

Table II and Table III show the results obtained for the Breast Cancer and Diabetes data sets. It is evident that our framework achieves the best performance for both data sets with F1 scores 92.27% and 69.04% for the Breast Cancer and Diabetes data sets, respectively. Table II shows that the F1 value is increased by 3.88% when all the algorithms are trained with an imbalanced data set. Although, the discriminative models improved their performance when they were combined with oversampling techniques (c.f. increase in their recall score), still they are outperformed by our method by 2.92% on the F1 score.

For the Diabetes data set, the classifiers performed poorly due to the high intersection between the negative and positive samples. However, our method enhanced the F1 score by 5.65% on imbalanced training. Again, when ADASYN and SMOTE were combined with LR, RF and MLP, the recall value is increased significantly but precision is decreased. This suggests that when oversampling is used, classification

TABLE II: Breast Cancer detection results ($mean \pm std$): accuracy, recall, precision and F1 measure

Breast Cancer				
Method	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.8959 ± 0.0093	0.8053 ± 0.0345	0.9095 ± 0.0279	0.8534 ± 0.0185
LR + SMOTE	0.9114 ± 0.0175	0.8762 ± 0.0358	0.8813 ± 0.0387	0.8778 ± 0.0242
LR + ADASYN	0.9005 ± 0.0196	$\mathbf{0.9448 \pm 0.0300}$	0.8182 ± 0.0289	0.8766 ± 0.0238
RF	0.9134 ± 0.0110	0.8891 ± 0.0337	0.8802 ± 0.0230	0.8839 ± 0.0152
RF + SMOTE	0.9187 ± 0.0181	0.9232 ± 0.0353	0.8674 ± 0.0379	0.8935 ± 0.0239
RF + ADASYN	0.9052 ± 0.0245	0.9392 ± 0.0283	0.8230 ± 0.0316	0.8771 ± 0.0277
MLP	0.9157 ± 0.0309	0.8732 ± 0.0761	0.8889 ± 0.0528	0.8782 ± 0.0475
MLP + SMOTE	0.9093 ± 0.0294	0.9207 ± 0.0324	0.8468 ± 0.0599	0.8800 ± 0.0251
MLP + ADASYN	0.8871 ± 0.0264	0.8894 ± 0.0487	0.8361 ± 0.0739	0.8578 ± 0.0288
SSGAN-c+SAE	$\mathbf{0.9227 \pm 0.0193}$	0.9113 ± 0.0270	$\mathbf{0.93485 \pm 0.0238}$	$\mathbf{0.9227 \pm 0.0193}$

TABLE III: Diabetes detection results ($mean \pm std$): accuracy, recall, precision, F1 measure

Pima Diabetes				
Methods	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.7656 ± 0.0204	0.5074 ± 0.0598	$\mathbf{0.7455 \pm 0.0483}$	0.6013 ± 0.0440
LR+SMOTE	0.7604 ± 0.0298	0.6685 ± 0.0315	0.6598 ± 0.0601	0.6626 ± 0.0333
LR+ADASYN	0.7370 ± 0.0335	0.7444 ± 0.0785	0.6006 ± 0.0385	0.6638 ± 0.0406
RF	0.7688 ± 0.0193	0.5741 ± 0.0603	0.7145 ± 0.0405	0.6339 ± 0.0386
RF+SMOTE	0.7442 ± 0.0236	0.7037 ± 0.0530	0.6203 ± 0.0339	0.6582 ± 0.0324
RF+ADASYN	0.7357 ± 0.0363	0.7741 ± 0.0567	0.5965 ± 0.0451	0.6727 ± 0.0420
MLP	0.7513 ± 0.0409	0.5741 ± 0.0824	0.6748 ± 0.0780	0.6166 ± 0.0679
MLP+SMOTE	0.7591 ± 0.03551	0.7435 ± 0.0725	0.6357 ± 0.0483	0.6834 ± 0.0467
MLP+ADASYN	0.7351 ± 0.0476	$\mathbf{0.8000 \pm 0.0880}$	0.5907 ± 0.0531	0.6786 ± 0.0610
SSGAN-c+SAE	$\mathbf{0.79058 \pm 0.0321}$	0.6515 ± 0.0535	0.7381 ± 0.0428	$\mathbf{0.6904 \pm 0.0210}$

algorithms are able to identify better the minority class, still their performance related to the majority class is reduced.

We further evaluate the proposed method on the Credit Card Fraud data set. The algorithms are trained with the original data, data extracted using Principal Component Analysis (PCA) as a baseline, and representation data from the sparse auto-encoder. The results are reported in Table IV. The performance of SSGAN is improved significantly when the representations from the auto-encoder are used to train the model with an increase of 3.76% of recall and 2.31% of the F1 score. This validates our choice to use the extracted features from the sparse auto-encoder as input to the GAN framework. Table V shows the results of the discriminative models in combination with ADASYN and SMOTE for the Credit Card data set. Importantly, the SSGAN framework continues to achieve the best F1 score of 92.31%. In Table IV we also show the results when we train the SSGAN with a regular generator (SSGAN-r) as opposed to the complementary gen-

TABLE IV: Credit card fraud detection results ($mean \pm std$): accuracy, recall, precision and F1 measure

Credit Card Fraud					
Input	**Method**	**Accuracy**	**Recall**	**Precision**	**F1**
Original Data	SSGAN-c	0.9629 ± 0.0032	0.8297 ± 0.0230	0.9874 ± 0.0135	0.9005 ± 0.0096
	SSGAN-r	0.9424 ± 0.0367	0.8109 ± 0.0627	0.9041 ± 0.1105	0.8538 ± 0.0829
	Logistic Regression	0.9577 ± 0.0110	0.7972 ± 0.0523	$\mathbf{0.9954 \pm 0.0046}$	0.8853 ± 0.0328
	Random Forest	0.9667 ± 0.0013	0.8393 ± 0.0086	0.9924 ± 0.0032	0.9086 ± 0.0041
	MLP	0.9629 ± 0.0014	0.8264 ± 0.0259	0.9888 ± 0.0159	0.9000 ± 0.0087
PCA	SSGAN-c	0.9381 ± 0.0298	0.8113 ± 0.0353	0.8426 ± 0.1209	0.8426 ± 0.0642
	SSGAN-r	0.9577 ± 0.0054	0.8001 ± 0.0351	0.9821 ± 0.01380	0.8822 ± 0.0179
	Logistic Regression	0.9409 ± 0.0275	0.7737 ± 0.0565	0.9224 ± 0.1003	0.8400 ± 0.0696
	Random Forest	0.9519 ± 0.0102	0.7990 ± 0.0354	0.9513 ± 0.0319	0.8681 ± 0.0290
	MLP	0.9152 ± 0.0444	0.8000 ± 0.0356	0.8081 ± 0.1540	0.7968 ± 0.0842
Latent Representations	SSGAN-c	$\mathbf{0.9707 \pm 0.0019}$	0.8673 ± 0.0125	0.9869 ± 0.0165	$\mathbf{0.9231 \pm 0.0047}$
	SSGAN-r	0.9158 ± 0.0323	0.6404 ± 0.1886	0.9139 ± 0.0686	0.9158 ± 0.1551
	Logistic Regression	0.9232 ± 0.0124	0.6230 ± 0.0571	0.9911 ± 0.0117	0.7633 ± 0.0416
	Random Forest	0.9349 ± 0.0128	0.6970 ± 0.0682	0.9690 ± 0.0170	0.8089 ± 0.0488
	MLP	0.9619 ± 0.0016	0.8334 ± 0.0058	0.9706 ± 0.0066	0.9151 ± 0.0331

TABLE V: Credit Card Fraud detection results in conjunction with oversampling ($mean \pm std$): accuracy, recall, precision, F1 measure. Comparing with the results of Table IV, our proposed architecture achieves the highest F1 measure.

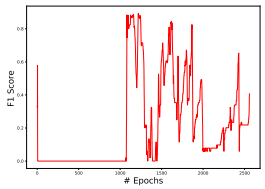
Methods	**Accuracy**	**Recall**	**Precision**	**F1**
LR + SMOTE	0.9691 ± 0.0055	0.8837 ± 0.0286	0.9577 ± 0.0151	0.9189 ± 0.0156
RF + SMOTE	0.9826 ± 0.0043	0.8620 ± 0.0346	0.9521 ± 0.0184	0.9045 ± 0.0245
MLP + SMOTE	0.9682 ± 0.0086	0.8858 ± 0.0283	0.7956 ± 0.0778	0.8353 ± 0.0376
LR + ADASYN	0.9154 ± 0.0147	0.9172 ± 0.0210	0.7272 ± 0.0465	0.8103 ± 0.0285
RF + ADASYN	0.9677 ± 0.0042	0.8561 ± 0.0279	0.9769 ± 0.0200	0.9120 ± 0.0136
MLP + ADASYN	0.8760 ± 0.0168	$\mathbf{0.9394 \pm 0.0262}$	0.6269 ± 0.0365	0.7517 ± 0.0328

erator (SSGAN-c) used in our framework. A regular SSGAN has the same architecture as the original GAN model with the addition of an extra output in the discriminator. Our framework improves consistently on the regular generator.

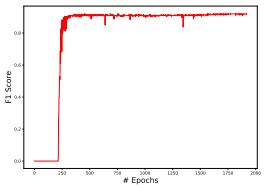
In order to tune the sparse auto-encoder in our experiments we altered the hidden dimension size from 20 up to 80 neurons. Precision had a small variation throughout all the different experiments due to the large number of non-fraudulent cases in the training set. On the other hand, recall had a significant increase when the dimension changed from 20 to 30 ($\approx 10\%$) and from 60 to 65 ($\approx 2\%$). Then, when dimension size changes from 65 to 80 a small decrease in recall was observed. Mapping the original data to a higher dimension allows data to be separated more easily. Nevertheless, if the dimension is too high it can lead to overfitting and information redundancy [19]. Finally, we also noted that the trend of F1 score follows the trend of recall as fluctuations occurred in the same examples for both metrics.

Focusing on the semi-supervised GANs, the complementary SSGAN has better performance compared to SSGAN-r as already mentioned as shown in Table IV. The discriminator of SSGAN-c, which is trained on real and complementary data, can classify more effectively the positive and negative cases since better recall and precision scores are achieved compare to SSGAN-r.

The training behaviour of the two models was further investigated and the progress of the F1 score on the Credit Card data set is presented in Fig. 3a and Fig. 3b. In Fig. 3, the regular SSGAN shows an inability to converge during training, while the SSGAN-c framework converges. The reason for this is that during the training phase the complementary GAN focuses on the classification task of predicting the correct class whilst the regular GAN focuses on generating better fake samples [21].



(a) F1 of SSGAN-r



(b) F1 of SSGAN-c framework

Fig. 3: Fig. 3a and Fig. 3b show the F1 score progress during training of a regular SSGAN and our complementary SSGAN framework.

V. Application of SSGAN to the Detection of Money Laundering in Online Gambling

We tested SSGAN-c on a real-world Gambling Fraud data set. Anonymized gambling data were collected over the period of one year starting 1st April 2018. The current anti-money laundering practice is composed of different monitoring levels. Our system targets to improve the first level of monitoring (identification rate of knowledge-based system). The labels provided in the data set represent the Internal Risk Reports (IRR) that were raised for high risk money laundering cases during the monitoring process.

The data set that is used in this experiment incorporates information about transactions and betting history of customers. It contains 4,700 samples, 3,500 of which are non-fraudulent players and the remaining 1,200 players are flagged for potential money laundering and further investigation. The F1 value of the knowledge-based system as this is calculated using the IRR labels and the detection flags of the system is 86.21%.

Table VI outlines the comparative results obtained for the Gambling Fraud data set. The SSGAN-c framework achieves F1 score of 89.85%, which yields a 3.64% ($\approx$ 20 cases) improvement on the company's current detection system and an 0.52% ($\approx$ 3 cases) improvement in comparison with the other methods. This is an indication that our fraud detection system can be applied to the detection of fraudulent behaviour in online gambling and improve the overall identification rate.

VI. Conclusion and Future Work

In this paper, we developed a GAN-based system architecture for detecting fraud in online gambling. Our implementation consists of a complementary generative adversarial

TABLE VI: Gambling Fraud detection results ($mean \pm std$): accuracy, recall, precision, F1 measure

Gambling Fraud				
Methods	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.8733 ± 0.0091	0.6103 ± 0.0296	0.8751 ± 0.0224	0.7187 ± 0.0234
LR+SMOTE	0.9089 ± 0.0103	0.8482 ± 0.0206	0.8164 ± 0.0225	0.8318 ± 0.0185
LR+ADASYN	0.9000 ± 0.0068	0.8960 ± 0.0152	0.7671 ± 0.0171	0.8264 ± 0.0104
RF	0.9424 ± 0.0059	0.9095 ± 0.0212	$\mathbf{0.8781 \pm 0.0115}$	0.8933 ± 0.0116
RF+SMOTE	0.9361 ± 0.0071	0.9458 ± 0.0062	0.8355 ± 0.0146	0.8872 ± 0.0106
RF+ADASYN	0.9347 ± 0.0057	$\mathbf{0.9569 \pm 0.0139}$	0.8256 ± 0.0165	0.8862 ± 0.0091
MLP	0.9194 ± 0.0100	0.8419 ± 0.0289	0.8534 ± 0.0245	0.8472 ± 0.0195
MLP+SMOTE	0.9203 ± 0.0060	0.9372 ± 0.0248	0.7984 ± 0.0167	0.8618 ± 0.0103
MLP+ADASYN	0.9219 ± 0.0039	0.9526 ± 0.0133	0.7946 ± 0.0118	0.8663 ± 0.0060
SSGAN-c+SAE	$\mathbf{0.9437 \pm 0.0051}$	0.9308 ± 0.0157	0.8672 ± 0.0170	$\mathbf{0.8985 \pm 0.0088}$

network and a sparse auto-encoder. First, we used the auto-encoder to extract new data representations which were then used to train our GAN model.

A series of experiments were performed to evaluate the proposed system architecture against popular discriminative models such as logistic regression and random forest, both on their own and in conjunction with data balancing SMOTE and ADASYN. Experiments were performed on three publicly available data sets and on a real-world gambling data set. We demonstrated that our system outperforms the other classification methods by achieving a higher F1 score. Overall, the results showed that complementary semi-supervised GANs can be a useful versatile framework for tackling supervised problems with imbalanced and sparse structured data. In future, results will be compared with other deep network models including with the use of sparse coding [33]. Further, we plan to test different sparse coding methods as well as use the generator of the SSGAN to produce synthetic data which we will use as part of a more extensive experiment with the objective of further improving system performance ahead of deployment.

ACKNOWLEDGMENT

We would like to thank Kindred Group plc for funding this research, providing data and supporting the evaluation of the results. We are also grateful to the reviewers for their comments and suggestions.

REFERENCES

[1] A. M. Mubarek and E. Adalı, "Multilayer perceptron neural network technique for fraud detection," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 383–387.

[2] B. Pal and M. K. Paul, "A gaussian mixture based boosted classification scheme for imbalanced and oversampled data," in *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2017, pp. 401–405.

[3] R. Ou, A. L. Young, and D. B. Dunson, "Clustering-enhanced stochastic gradient mcmc for hidden markov models with rare states," *arXiv preprint arXiv:1810.13431*, 2018.

[4] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[5] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Advances in neural information processing systems*, 2017, pp. 6510–6520.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[7] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[8] A. Hanskunatai, "A new hybrid sampling approach for classification of imbalanced datasets," in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2018, pp. 67–71.

[9] Y. Wu, L. Shen, and S. Zhang, "Fuzzy multiclass support vector machines for unbalanced data," in *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, 2017, pp. 2227–2231.

[10] P. Shukla and K. Bhowmick, "To improve classification of imbalanced datasets," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2017, pp. 1–5.

[11] H. Luo, X. Pan, Q. Wang, S. Ye, and Y. Qian, "Logistic regression and random forest for effective imbalanced classification," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 916–917.

[12] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "Gmm-based undersampling and its application for credit card fraud detection," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[13] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint arXiv:1904.10604*, 2019.

[14] Z. Zojaji, R. E. Atani, A. H. Monadjemi *et al.*, "A survey of credit card fraud detection techniques: data and technique oriented perspective," *arXiv preprint arXiv:1611.06439*, 2016.

[15] M. Zareapoor, P. Shamsolmoali *et al.*, "Application of credit card fraud detection: Based on bagging ensemble classifier," *Procedia computer science*, vol. 48, no. 2015, pp. 679–685, 2015.

[16] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 483–490.

[17] E. Saraswathi, P. Kulkarni, M. N. Khalil, and S. C. Nigam, "Credit card fraud prediction and detection using artificial neural network and self-organizing maps," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019, pp. 1124–1128.

[18] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[19] J. Chen, Y. Shen, and R. Ali, "Credit card fraud detection using sparse autoencoder and generative adversarial network," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 1054–1059.

[20] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[21] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-class adversarial nets for fraud detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1286–1293.

[22] T. Zhou, W. Liu, C. Zhou, and L. Chen, "Gan-based semi-supervised for imbalanced data classification," in *2018 4th International Conference on Information Management (ICIM)*. IEEE, 2018, pp. 17–21.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[24] A. Ng and S. Autoencoder, "Cs294a lecture notes," *Dosegljivo: https://web. stanford. edu/class/cs294a/sparseAutoencoder_2011new. pdf.[Dostopano 20. 7. 2016]*, 2011.

[25] C. Zhang, X. Cheng, J. Liu, J. He, and G. Liu, "Deep sparse autoencoder for feature extraction and diagnosis of locomotive adhesion status," *Journal of Control Science and Engineering*, vol. 2018, 2018.

[26] D. Yang, J. Lai, and L. Mei, "Deep representations based on sparse auto-encoder networks for face spoofing detection," in *Chinese Conference on Biometric Recognition*. Springer, 2016, pp. 620–627.

[27] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[28] M. Ranzato, Y.-L. Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in neural information processing systems*, 2008, pp. 1185–1192.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[32] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[33] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.