ISSN 2278-3091
Volume 9 No.2, March -April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse26922020.pdf
https://doi.org/10.30534/ijatcse/2020/26922020

Imbalanced Data Classification Using Auxiliary Classifier Generative Adversarial Networks

Indrajani Sutedja[1], Yaya Heryadi[2], Lili Ayu Wulandhari[3], Bachtiar Abbas[4]
[1]Bina Nusantara University, Indonesia, indrajani@binus.ac.id
[2] Bina Nusantara University, Indonesia, yayaheryadi@binus.edu
[3] Bina Nusantara University, Indonesia, lili.wulandhari@binus.ac.id
Bina Nusantara University, Indonesia, bachtiars@binus.edu.

ABSTRACT

This paper explores an Auxiliary Classifier Generative Adversarial Networks (AC-GAN) model to address imbalanced data classifications. More specifically, debit card transaction data from a private Indonesian bank were categorized into fraudulent and non-fraudulent transactions. Training and testing datasets were then analyzed. Due to its architectural similarity, this study explores the learning performance of an AC-GAN Discriminator model and compares it with the learning performance of a Convolutional Neural Network (CNN) model, as a baseline model. The experimental results illustrate that the training and testing accuracy of the AC-GAN discriminator model (i.e., 81.1% and 93.0%, respectively) outperformed the CNN model (i.e., 68.1% and 67.7%). Given the similar structures of AC-GAN discriminators and CNN models, it is hypothesized that AC-GAN discriminators can perform better than CNNs. This is because the effect of the adversarial training approach embedded up-sampling techniques into the model-training process.

Key words : adversarial training, Auxiliary Classifier Generative Adversarial Networks, imbalanced data classification..

1. INTRODUCTION

Imbalanced data classification is an interesting problem in the machine learning research field. Imbalanced data is a term which refers to the skewed number of class samples over other class samples. Learning under imbalanced datasets results in problems. These problems are not discovered when the classifier models are trained on relatively balanced data. The models learn more patterns from majority class samples than minority class samples. As such, the trained classifier tends to exhibit bias towards the majority class samples. That is, the imbalanced data may make the trained classifier unable to recognize the minority class.

Imbalanced data classification problems can be found in various research areas, ranging from engineering, bioinformatics, and banking transactions to the field of medicine. With the wide breadth of potential applications, finding solutions to imbalanced data classification problems has gained extensive attention from various research communities. Over the past decade, a plethora of methods have been proposed to address imbalanced data classification problems.

Reference [1] broadly categorized the methods for addressing imbalanced data into two categories: data manipulation methods and algorithm-oriented methods [1]. Data manipulation methods rely on sampling-based techniques (e.g., duplicating minority class samples, creating new samples by corrupting existing samples with artificial noise, up-sampling (i.e., Synthetic Minority Over-sampling Technique (SMOTE) [2]).

One technique is to eliminate over-sized class samples at random until its size matches the size of the other class techniques. This technique is used to reduce the disparity between the number of samples among the classes and the down-sampling [3]. The drawback of this approach is that it either leads to over-fitting or it ignores the majority class, resulting in a loss of information.

Algorithm oriented methods focus on developing learning algorithms which are insensitive to a class samples distribution in the training dataset [4]-[6].

In the banking industry, recognizing fraudulent debit card transactions is a well-known, but challenging, problem. This problem is due to the imbalanced data in nature [7]. In this context, the term ‘fraudulent transactions’ refers to a class of illegal transactions made by someone who impersonates a debit card holder. Fraudulent transactions cause financial losses and erode the prudent image of the card issuer banks.

To address this problem, many models have been proposed to solve the imbalanced data classification problem. These models include extended decision tree models with resampling techniques [8], the combination of machine learning and deep learning models with resampling techniques [1][9]-[10], and game theoretic-based models [6].

The primary purpose of this paper is to address imbalanced data classifications using the adversarial learning algorithm that unifies data treatment with learning algorithms. To achieve this objective, this research explores Auxiliary

Classifier Generative Adversarial Network (AC-GANs) models to solve an imbalanced binary data classification. The classification is comprised of two classes: fraudulent and non-fraudulent transactions.

This investigation adopts an adversarial training approach for training classifiers that is similar to the model proposed in [6]. Our model differs from the Zeager [6] model in two primary ways. Firstly, the former research proposed logistic regression models as classifiers, which were trained in adversarial ways. This research proposes an AC-GAN model. Secondly, for the data dimensional reduction, the former study implemented Gaussian Mixture Models (GMMs). This research uses a stack of convolution layers, rectified linear units, and pooling.

2. LITERATURE REVIEW

2.1 Imbalanced data classification

The imbalanced data classification problem has drawn extensive interest from researchers. This has resulted in a plethora of proposed methods that can be used to address this problem. The imbalanced data might only have a small impact on classifications when the data is linearly separable. However, in general, the trained classifier tends to be biased towards the majority classes [11]. This effect is not favorable for data classifications in various domains.

Many models have been proposed to solve the imbalanced data classification problem. These models include: k-NN [8] [12], SVM [13], SVM and HMM Hybrids [14]-[15], HMMs [16] [8]-[9], Neural Networks and CNN models [1] [17].

Over the past ten years, many research studies have reported various successful results in relation to adopting deep learning models to address classification problems. These models are based on the premise that the training dataset is balanced [18]-[23]. Most of the proposed models in these prominent research articles are extended Convolutional Neural networks (CNN) using a variety of techniques. These techniques include increasing the number of layers [24], increasing the layer size [24] [25], introducing a dropout layer [26], and combining discriminators and generator models [27]-[28].

A recent study by Zeager et al. [6] reports on imbalanced data classification methods. The authors propose a learning algorithm based on an adversarial game theoretical approach. In the proposed method, an adversarial training algorithm is designed using a logistic regression, Gaussian Mixture Models (GMM) and a Synthetic Minority Oversampling Technique (SMOTE). The empirical results show that the proposed approach is promising in addressing the imbalanced data classification problem.

2.2 Convolutional Neural Networks

Convolutional neural networks (CNN) are a class of neural network models with standard structures. LeNet-5 proposed by [29] is the first well known CNN model.

The CNN architecture is comprised of the following layers:

1. Convolutional layer: A layer to capture the local dependencies in the original image by preserving the spatial relationship between the pixels.
2. Non-linearity layer: A layer to introduce non-linearity in CNN.
3. Pooling or subsampling layer: A layer aimed at reducing the dimensionality of each feature map, but retaining the most important information.
4. Classification or fully connected layer: One or several layers used for classifying the input image into various classes, based on the training dataset.

A sample of the CNN architecture is presented in Figure 1.

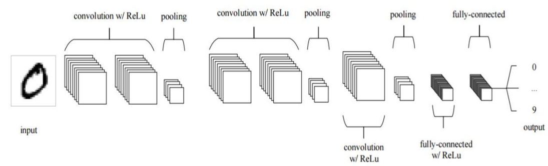

Figure 1: General Configuration of a CNN Model [30].

The objective function L, of the CNN model training, is formulated as follows:

$$L = \frac{1}{2}\sum_{i=1}^{b}\sum_{j=1}^{b_i}\left(t_{ij} - o_{ij}\right)^2 \qquad (1)$$

where: t_{ij} is the actual class of the j^{th} sample of the i^{th} training batch and o_{ij} is the predicted class of the j^{th} sample of the i^{th} training batch. The performance of the trained CNN model is measured using an accuracy metric formulated as follows:

$$Accuracy = \frac{TF+TN}{N} \qquad (2)$$

where: TF is the true positive; TN is the true negative; and N is the total number of training or testing samples.

Over the past two decades, a vast number of studies in various domains have established CNN as a robust class of models. These models are used to address various pattern recognition problems (e.g., identifying faces and objects, traffic sign detection and recognition, image segmentation, image retrieval). Other problems include large video datasets for video classifications [31], the fusion of several image modalities from a large image dataset for pedestrian recognition [32], and a large text dataset for sentence classification and modeling [33]-[34]. These studies resulted in many variations of CNN models, which have become ubiquitous, those are not only in the image/video classification literature. Surprisingly, these models illustrate outstanding perform for classifications using several benchmark datasets (e.g., MNIST, CIFAR, ImageNet datasets).

The popularity of ImageNet challenges has motivated many researchers to develop variations in the CNN architecture to solve classification problems using large-scale datasets. Many of these models have been categorized as top image classifiers in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (e.g., AlexNet[19], GoogleNet [24], VGG [35], and ResNet [36]).

Karpathy, Toderici, Shetty, Leung, Sukthankar, and Fei-Fei [31] argue that the high performance of CNN-based models for image classification is mainly due to its architecture. This architecture enables researchers to build neural networks with many parameters to learn from a given input dataset. This study uses the CNN model in Karpathy, Toderici, Shetty, Leung, Sukthankar, & Fei-Fei [31], a baseline model for recognizing patterns from imbalanced datasets. The CNN model is then compared with an AC-GAN discriminator model that has a similar structure.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were proposed for the first time in Goodfellow et al. [37]-[38]. GANs were initially designed as an image synthesis model which was trained using an adversarial technique. Some studies illustrate evidence that GAN models perform well in generating considerable image samples on datasets with low variability and low resolutions [39]-[40].

GAN architecture (Figure 2a) is comprised of two neural networks acting as a generator (G) and a discriminator (D). The discriminator model of a GAN is trained to recognize data from the original training dataset and the synthesized data generated by the generator. To achieve that objective, the two neural networks are trained in opposition to one another (the adversarial way) as follows:

The generator G takes, as an input, a random noise vector z. It then outputs an image X_{fake}=G(z). Hence, X_{fake} is an image produced by the generator G with a random noise vector z as an input.

The discriminator D receives an input, either as a training image or a synthesized image from the generator. The discriminator then outputs a probability distribution P(S|X)=D(X), over possible image sources, where P(S|X) is the probability of the given input as a fake/non-fake image.

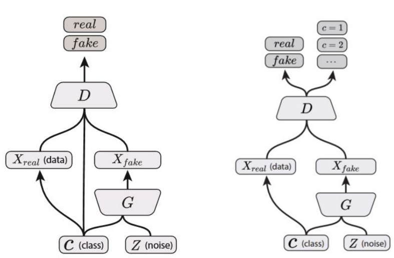

Figure 2: (a) General Architecture of the GAN Model and (b) the AC-GAN Model [26]

In adversarial training, the discriminator is trained under supervision to maximize a log-likelihood. On the other hand, the generator is trained to minimize that same quantity. The lost function for a GAN model is formulated as:

$$L = E[\log P(S=real \mid X_{real})] + E[\log P(S=fake|X_{fake})] \quad (3)$$

where: P(S|X) is the probability distribution over the source image, given the X image as an input. The input images can be training images or synthesized images. The results of the GANs training process are trained generators (G) and trained discriminators (D).

2.4 Auxiliary Classifier Generative Adversarial Networks

The Auxiliary Classifier Generative Adversary Network (AC-GAN), proposed by [26], is a variant of the GAN model. Like previous GANs, the AC-GAN model is comprised of discriminator and generator models (Figure 2b). These models are trained with conflicting learning objectives or adversarial ways.

In contrast to previous GANs, in which the discriminator is only designed to categorize the input data as a fake/non-fake class, the AC-GAN discriminator model is extended to become a general classifier. Given original and synthesis data as inputs, the discriminator, as an auxiliary classifier, is designed to categorize input into: a fake/non-fake class and a label of the data class. Whilst the theoretical aspect of AC-GAN continues to gain research attention [39], some experiments illustrate that the AC-GAN model is powerful enough to solve classification problems in many areas.

In AC-GANs, every generated sample has a corresponding class label c, with a distribution function p_c, c~p_c, in addition to the noise z.

1. The generator G model uses class labels and noise to generate images X_{fake}=G(c,z).
2. The discriminator D gives both a probability distribution over sources P(S|X) and a probability distribution over the class labels P(C|X)=D(X). The objective function has two parts: the log-likelihood of the correct source (L_S) and the log-likelihood of the correct class (L_C), where:

$$L_S= E[\log P(S = real \mid X_{real})] + E[\log P(S=fake \mid X_{fake})] \quad (4)$$

$$L_C= E[\log P(C = c \mid X_{real})]+E[\log P(C=c \mid X_{fake})] \quad (5)$$

The adversarial training of AC-GAN trains the D model to maximize L_S+L_C and trains the G model to maximize L_C-L_S. With such adversarial learning objectives, we hypothesized that the discriminator of AC-GAN achieved stronger learning performance than CNN, which is trained separately.

In recent years, researchers have been experiencing some successful results in generating close-to-natural synthetic images, due to the improvement in AC-GANs. One common challenge is improving the performance of globally synthesizing coherent and high-resolution image samples from datasets with high variability as an input. Odena, Olah, and Shlens [26] argue that adding more structure to the AC-GAN latent space and modifier cost function will improve the quality of its generated samples.

Despite many studies adopting the strength of AC-GAN models in addressing classification problems, to the best of our knowledge, little has been said on the other strength of this model in addressing imbalanced data classifications. Imbalanced data treatment by AC-GAN can be described as follows. As part of the discriminator training process, the generator keeps producing synthetic images. Due to the added noise, these images are different from the original data. Therefore, producing synthetic images can be viewed as an up-sampling process.

3. METHODS

3.1 Research Framework

The research process blocks of this study are represented in the following diagram (Figure 3). As can be seen in Figure 3, both classifier models in this investigation are trained under supervision using a variation of the stochastic gradient descent algorithms. However, the discriminator of the AC-GAN model is trained in adversarial ways, as compared to its generator model counterpart.

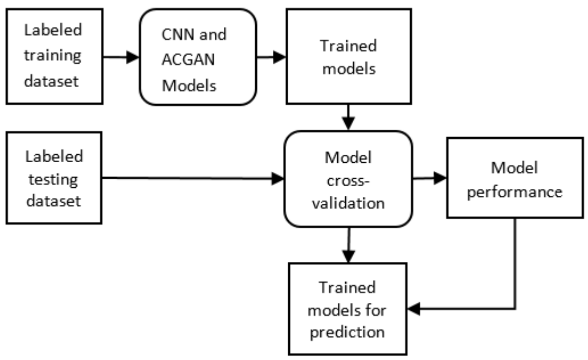

Figure 3: Research Framework.

3.2 Dataset

Data for debit card transactions collected during the period of 2016-2017 were analyzed. The data were obtained from the data warehouse of a local Indonesian bank. Each transaction was labeled by the bank as either fraudulent or non-fraudulent. Permission for the limited use of the data for this research was granted by the management of the bank. For banking confidentiality reasons, confidential data (e.g., cardholder identities, place and time of transaction, device information, and bank-added codes) are excluded from the dataset.

This research represents debit card transactions by a sequence of daily transactions over a monthly period as transaction features [41]-[45]. In this study, we added 9 derivative features (Figure 4). In this way, a data unit is comprised of 40 features. To simplify the processing of the data, each month is assumed to have 31 days. Zero transaction amounts are added as padding for the transactions in a calendar month with fewer than 31 days (e.g., February, April, and November).

For active debit cards, a unit of data with the NF (non-fraudulent) label is represented as a list of the past 31-day transactions. In contrast, for a blocked debit card, its last 31-day transaction (cross calendar month) is used to form the unit data with the F (fraudulent) label. For unit data with a fraudulent label, the last daily transaction is the fraudulent transaction. In this research, it is assumed that the previous history of credit card transactions is non-fraudulent, unless deemed so by the bank.

The following are the additional features and labels used to capture short and middle range transaction patterns. These were all averages: 2-days (2d-avg), 3-days (3d-avg), 4-days (4d-avg), 5-days (5d-avg), 6-days (6d-avg), 7-days (1w-avg), 2-weeks (2w-avg), 3-weeks (3w-avg), and 1-month (1m-avg). The final transaction feature can be represented using the following figure.

Sample No	1d	2d	3d	...	2d-avg	3d-avg	..	1w-avg	2w-avg	3w-avg	1m-avg
1											
2											
3											
..											
..											
N											

Figure 4: The Debit Card Transaction Features

The dataset was comprised of 22,683 data unit transactions represented by a 22,683×40 matrix, $T=[T_{ij}]$, where i represents the index of the debit card samples and j is the j^{th} transaction feature. A 22,683×1 transaction label $R=[R_i]$ represents the category of the T_{ij} transaction. Each raw matrix element is an integer ranging from 0 to 100,000,000 (maximum transaction allowed by the bank). The data pre-processing (data normalization) transforms the raw data x using the following steps: transformation, $x^{(i)} = \log(x_i+1)$, followed by $x_n^{(i)} = \frac{x^{(i)}}{max(x^{(i)})}$ where x is the raw data and $x_n^{(i)}$ is the ith normalized data (i=1,2,..,N). The normalized data ranges from [0, 1].

The observations reveal that the dataset has the following characteristics:

1. The sample distribution is skewed. The dataset for this research is comprised of: (1) non-fraudulent (13,405 (59%) samples); and (2) fraudulent data (9,278 (41%) samples). That is, there are more non-fraudulent samples than fraudulent samples.
2. The transaction matrix is highly sparse. The data illustrates that most debit card holders do not make transactions every day. However, the amount of the transaction varies significantly.

3.3 Module Training and Cross-Validation

The models explored in this study include Auxiliary Classifier Generating Adversarial Networks (AC-GAN) [26]. Convolutional neural networks (CNN) [28] were used as the baseline model (Figure 1).

In this research, all data were represented as 1×40 vectors. The architectural components of the CNN, as well as the AC-GAN discriminator, were a stack of several layers.

- Convolutional layer: 100×1×4
- Convolutional layer: 100×1×4
- Non-linearity layer: LeakReLU function
- Pooling layer: maxpooling function
- Dropout layer
- Convolutional layer: 50×1×4
- Non-linearity layer: LeakReLU function
- Pooling layer: maxpooling function
- Dropout layer
- Convolutional layer: 30×1×4
- Pooling layer: maxpooling function

- Dropout layer
- Fully connected layer: dense layer of 500 nodes and sigmoid activation function.

The dropout layers in the above structure were added to handle the over-fitting model (model regularization).

Following [26], the general architecture of the AC-GAN follows the structure illustrated in Figure 2b. The main components of an AC-GAN model include the generator (G) and the discriminator (D) models. In this research, the structure of the generator can be summarized as follows:

- Dense layer: 1024 nodes
- Non-linearity layer: ReLU function
- Dense layer: 128 ×1×5 nodes
- Non-linearity layer: ReLU function
- Up-sampling layer: 1×3.
- Dense Layer: 256 nodes
- Non-linearity layer: ReLU function
- Up-sampling layer: 1×2 factor.
- Fully connected layers: dense layer of 128 nodes, ReLU layer, and tanh activation function.

Both the CNN and AC-GAN models were trained using a labeled dataset of two classes: fraudulent and non-fraudulent. A random uniform function between the intervals of [-1,1] is used to generate random noise for generating synthetic data or simulating the up-sampling process.

In this research, AC-GAN models are trained using an adversarial training technique. The AC-GAN and CNN models are optimized using the Adam algorithm. In this study, Adam's hyper parameter values include: learning rate (lr) = 0.0001 and beta1 = 0.1. The training process is implemented by stopping early. That is, the training process is stopped whenever the value of the validation loss is not decreasing.

To compare the performance of the AC-GAN discriminator model and the CNN, a loss function is measured during AC-GAN's discriminator training. This function is formulated equally with equation number 1. Finally, the performance of the AC-GAN discriminator model for the classification is measured using the accuracy metric, which is formulated equally with in equation number 2. Both the CNN and AC-GAN model training are cross-validated using the leave-one out technique. In this way, the dataset is divided into 18,146 samples (80%), as a training dataset, and 3,537 samples (20%), as the testing dataset. For simplification reasons, the performance metrics in this experiment were training loss and accuracy, as well as testing loss and accuracy.

Due to its complete function libraries available for the deep learning model implementation, the models in this research were implemented using Keras [43] and TensorFlow [13].

4. RESULTS AND DISCUSSIONS

The training and testing results of the CNN and AC-GAN discriminator models are described in the following figures. As can be seen in Figure 5, the training and testing of the CNN model produces a convergent training and testing loss.

Similarly, the training and testing accuracy of the CNN model increases gradually by the training and testing epochs (Figure 6).

In contrast to the training and testing loss of the CNN model, the training and testing loss of the AC-GAN discriminator model needs more time to converge (Figure 7). This is the result of the random noise in generating the synthetic data by the generator. However, the high testing loss of the CNN model shows that the trained CNN model is less capable of generalizing the patterns learned from the training dataset, rather than that of the AC-GAN discriminator model.

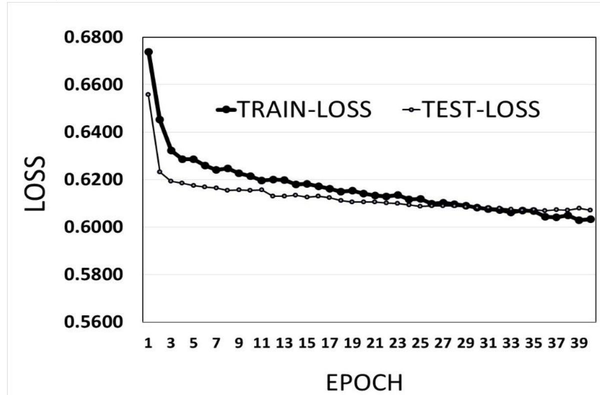

Figure 5: Training and Testing Loss of the CNN Model

In contrast to the training and testing loss of the CNN model, the training and testing loss of the AC-GAN discriminator model needs more time to converge (Figure 7). This is the result of the random noise in generating the synthetic data by the generator. However, the high testing loss of the CNN model shows that the trained CNN model is less capable of generalizing the patterns learned from the training dataset, rather than that of the AC-GAN discriminator model.

The high training and testing accuracy of the AC-GAN discriminator model (Figure 8) illustrates that the model has strong performance in generalizing patterns from its training dataset.

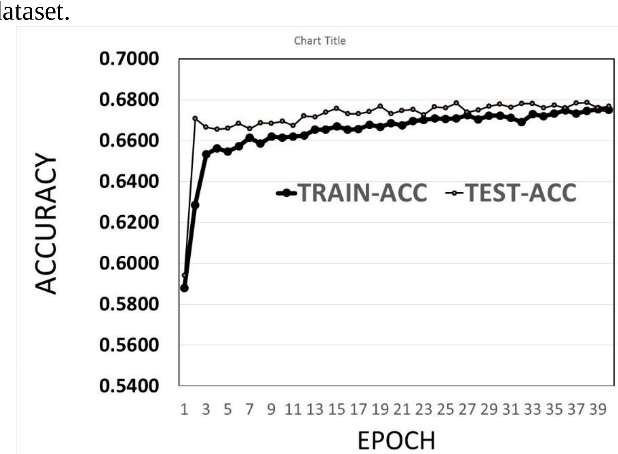

Figure 6: Training and Testing Accuracy of the CNN Model

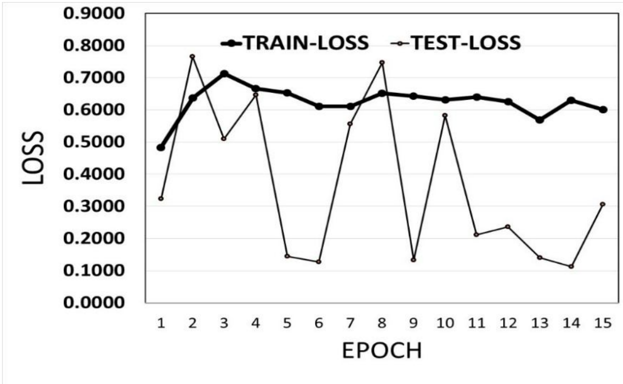

Figure 7: Training and Testing Loss of the AC-GAN Discriminator Model

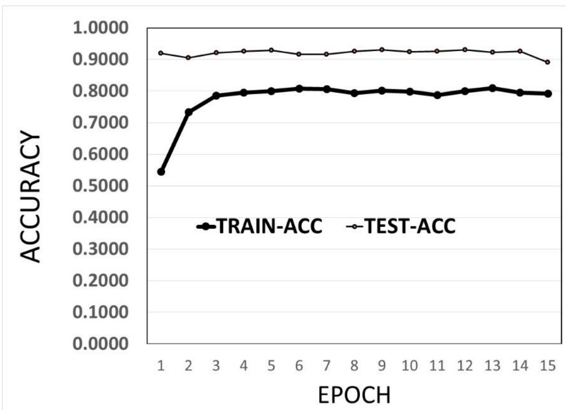

Figure 8: Training and Testing Accuracy of the AC-GAN Discriminator Model

Both results are summarized in Table 1. Surprisingly, as can be seen in Table 1, the AC-GAN discriminator model achieves higher training and testing accuracies than the CNN model. It is speculated that the high accuracy of the training and testing of the AC-GAN is partially due to the adversarial training algorithm used during the AC-GAN model training. In contrast to CNN, which is trained to learn patterns from the imbalanced data as an input, the AC-GAN discriminator model is trained to learn patterns from both the original and synthetic datasets in an adversarial way. As a result, the AC-GAN discriminator model learned a larger number of richer patterns to generalize the patterns from the training dataset.

Loss and Accuracy Comparison of the CNN and AC-GAN Discriminator Models

Table 1: Loss and Accuracy Comparison of the CNN and AC-GAN Discriminator Models

Model	*Training*		*Testing*	
	Loss (%)	*Accuracy (%)*	*Loss (%)*	*Accuracy (%)*
CNN	59.57	68.13	60.71	67.69
AC-GAN Discriminator	59.57	81.06	18.02	93.01

In addition, the adversarial training algorithm has contributed to the reduction effect of the imbalanced dataset during the AC-GAN training and testing process. As such, it reduces the bias towards the majority class.

5. CONCLUSION

The performance of a classifier to solve imbalanced data classification problems can highly benefit from the learning performance of the classifier or the capability of a classifier to generalize patterns from samples. Imbalanced data, however, can become a hindering factor for a classifier, in terms of achieving high learning performance from samples, as the training classifier will be biased to the pattern of the majority classes. Our experiment's results show that the AC-GAN discriminator model, which is trained in an adversarial way with the AC-GAN generator model counterpart, can achieve higher learning performance than the CNN with a similar architecture, but is trained separately.

Different models and different techniques can be used to address the imbalance in the data. However, this research finding validates the claim by Zeager et al. (2017)Zeager, Sridhar, Fogal, Adams, Brown, & Beling (2017), in that learning algorithms, based on the adversarial or game theory approach, have the potential to leverage the performance of deep learning models for a classification. The adversarial learning algorithm of the AC-GAN model enriches the learning patterns that will be learned by the discriminator model by introducing new patterns from the synthetic datasets. In contrast, the CNN training algorithm only learns patterns from the original data. In addition, the synthetic data added into the training dataset by the generator model in the AC-GAN reduces the bias in the trained discriminator model from only recognizing the majority class.
The experimentation results are only preliminary results used to address the imbalanced data classification further. Directions to improve this research include: exploring different performance metrics, increasing the number of samples, and exploring the different GAN models.

REFERENCES

1. Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen. **Deep learning for imbalanced multimedia data classification**, in IEEE International Symposium on Multimedia (ISM), 2015, pp. 483–488. https://doi.org/10.1109/ISM.2015.126
2. N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. **SMOTE: synthetic minority over-sampling technique**, J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002. https://doi.org/10.1613/jair.953
3. M. Kubat and S. Matwin. **Addressing the Curse of Imbalanced Training Sets: One Sided Selection**, in Proceedings of the Fourteenth International Conference on Machine Learning (ICML), vol. 97, pp. 179–186, 1997.
4. M. Kubat, R. C. Holte, and S. Matwin. **Machine learning for the detection of oil spills in satellite radar images**, Mach. Learn., vol. 30, no. 2–3, pp. 195–215, 1998.

5. F. Provost. **Machine learning from imbalanced data sets 101**, in Proceedings of the AAAI 2000 workshop on imbalanced data sets, 2000, pp. 1–3.
6. M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling. **Adversarial learning in credit card fraud detection**, in Systems and Information Engineering Design Symposium (SIEDS), 2017, pp. 112–116. https://doi.org/10.1109/SIEDS.2017.7937699
7. Indrajani, H. Prabowo, and Meyliana. **Learning Fraud Detection from Big Data in Online Banking Transactions: A Systematic Literature Review**, J. Telecommun. Electron. Comput. Eng., vol. 8, no. 3, pp. 127–131, 2016.
8. M. Anderka, T. Klerx, S. Priesterjahn, and H. K. Büning. **Automatic ATM Fraud Detection as a Sequence-based Anomaly Detection Problem**, in Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2014, pp. 759–764.
9. V. V Jog and N. R. Pardeshi. **Advanced Security Model for Detecting Frauds in ATM Transaction**, Int. J. Comput. Appl., vol. 95, no. 15, pp. 47–50, 2014.
10. S. Guan, M. Chen, H.-Y. Ha, S.-C. Chen, M.-L. Shyu, and C. Zhang. **Deep learning with MCA-based instance selection and bootstrapping for imbalanced data classification**, in Proceedings of the IEEE conference on Collaboration and Internet Computing (CIC), 2015, pp. 288–295.
11. N. Japkowicz. **Learning from imbalanced data sets: a comparison of various strategies**, in AAAI workshop on learning from imbalanced data sets, 2000, vol. 68, pp. 10–15.
12. M. Beckmann, N. F. F. Ebecken, and B. S. L. P. de Lima. **A KNN undersampling approach for data balancing**, J. Intell. Learn. Syst. Appl., vol. 7, no. 04, p. 104, 2015. https://doi.org/10.4236/jilsa.2015.74010
13. Sanjay KS, Ajit Danti. **Online Fake Review Identification based on Decision Rules**, International Journal of Advanced Trends in Computer Science and Engineering,Vol. 8, No.2, pp. 140-143, 2019. https://doi.org/10.30534/ijatcse/2019/07822019
14. J. Ni, T. Lambrou, and X. Ye. **Learning from life-logging data by hybrid HMM: a case study on active states prediction**, in Proceedings of the 12th IASTED International Conference Biomedical Engineering (BioMed 2016), 2016, pp. 70–74.
15. M. Abadi. **TensorFlow: A System for Large-Scale Machine Learning**, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, vol. 16, pp. 265–283.
16. S. S. Dhok and G. R. Bamnote. **Credit card fraud detection using hidden markov model**, Int. J. Adv. Res. Comput. Sci., vol. 3, no. 3, pp. 816–820, 2012.
17. Velasco, J. A **Smartphone-Based Skin Disease Classification Using MobileNet CNN**. International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 5, pp. 2632–2637, 2019. https://doi.org/10.30534/ijatcse/2019/116852019
18. D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. **Deep neural networks segment neuronal membranes in electron microscopy images**, in Advances in neural information processing systems, 2012, pp. 2843–2851.
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton. **ImageNet Classification with Deep Convolutional Neural Networks**, in Advances In Neural Information Processing Systems, 2012, pp. 1–9.
20. R. Girshick, J. Donahue, T. Darrell, and J. Malik. **Rich feature hierarchies for accurate object detection and semantic segmentation**, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
21. A. S. Razavian. **CNN Features off-the-shelf: an Astounding Baseline for Recognition**, CVPRW ’14 Proc. 2014 IEEE Conf. Comput. Vis. Pattern Recognit. Work, pp. 512–519, 2014. https://doi.org/10.1109/CVPRW.2014.131
22. C. Farabet, C. Couprie, L. Najman, and Y. Lecun. **Learning Hierarchical Features for scence labeling**, Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1915–1929, 2013.
23. Swetha Allam, Dr. Srinivas Kudipudi. **Churn Prediction using Attention Based Autoencoder Network**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 3, pp. 725-729, 2019. https://doi.org/10.30534/ijatcse/2019/60832019
24. C. Szegedy. **Going deeper with convolutions**, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
25. M. D. Zeiler and R. Fergus. **Visualizing and understanding convolutional networks**, in European conference on computer vision, 2014, pp. 818–833.
26. J. Schmidhuber. **Deep Learning in neural networks: An overview**, Neural Networks, vol. 61, pp. 85–117, 2015. https://doi.org/10.1016/j.neunet.2014.09.003
27. H. V Yamazaki and T. Yamaguchi. **Depth and Complexity of Deep Generative Adversarial Networks**, in The 31st Annual Conference of the Japanese Society for Artificial Intelligence, 2017, pp. 1–4.
28. A. Odena, C. Olah, and J. Shlens. **Conditional Image Synthesis With Auxiliary Classifier GANs**, IICLR 2017, pp. 1–16, 2017.
29. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. **Gradient-based learning applied to document recognition**, Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
30. H. Maghrebi, T. Portigliatti, and E. Prouff. **Breaking Cryptographic Implementations Using Deep Learning Techniques**, in International Conference on

Security, Privacy, and Applied Cryptography Engineering, 2016, pp. 3–26.

31. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. **Large-Scale Video Classification with Convolutional Neural Networks**, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725–1732. https://doi.org/10.1109/CVPR.2014.223
32. D. O. Pop, A. Rogozan, F. Nashashibi, and A. Bensrhair. **Fusion of stereo vision for pedestrian recognition using convolutional neural networks**, in ESANN 2017-25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2017.
33. N. Kalchbrenner, E. Grefenstette, and P. Blunsom. **A convolutional neural network for modelling sentences**, in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.
34. Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li. **Character-level Convolutional Networks for Text Classification**, in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2013, pp. 3057–3061.
35. Y. LeCun, Y. Bengio, and G. Hinton. **Deep learning**, Nature, vol. 521, no. 7553, pp. 436–444, 2015.
36. K. He, X. Zhang, S. Ren, and J. Sun. **Deep residual learning for image recognition**, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
37. I. J. Goodfellow. **Generative Adversarial Networks**, pp. 1–9, 2014.
38. I. Goodfellow. **Generative adversarial nets**, in Advances in neural information processing systems, 2014, pp. 2672–2680.
39. E. L. Denton, S. Chintala, R. Fergus, and others. **Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**, in Advances in neural information processing systems, 2015, pp. 1486–1494.
40. A. Radford, L. Metz, and S. Chintala, **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks**, ICLR 2016 UNSUPERVISED, pp. 1–16, 2016.
41. 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.
42. K. Fu, D. Cheng, Y. Tu, and L. Zhang. **Credit Card Fraud Detection Using Convolutional Neural Networks**, in International Conference on Neural Information Processing, 2016, pp. 483–490.
43. Abdullah, M., Agal, A., Alharthi, M., & Alrashidi, M. **Arabic handwriting recognition model based on neural network approach**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 1.1, pp. 253-258, 2019. https://doi.org/10.30534/ijatcse/2019/4581.120419
44. Kanika, Jimmy Singla. **Online Banking Fraud Detection System: A Review**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.3, pp. 959-962, 2019. https://doi.org/10.30534/ijatcse/2019/96832019
45. S. Tokui, K. Oono, S. Hido, and J. Clayton. **Chainer: a Next-Generation Open Source Framework for Deep Learning**, in Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015, pp. 1–6.