GAN Variants for Balancing Imbalanced Datasets

1. Problem Statement

Text classification is commonly used in tasks such as sentiment analysis and emotion detection , where a model learns to assign a label to a given piece of text. These models depend heavily on labeled datasets to perform well. However, in real-world scenarios, text datasets are often imbalanced, meaning that some classes appear much more frequently than others.

This imbalance creates a major challenge during training. Machine learning models tend to focus on the majority class because it dominates the dataset. As a result, the model may show good overall accuracy while failing to correctly classify minority classes. This is especially problematic in emotion and sentiment analysis, where minority emotions may be important but underrepresented.

Due to class imbalance, the classifier struggles to learn enough meaningful patterns from minority class samples. This usually leads to poor performance on these classes, such as low recall and F1-score, and a high number of misclassifications. In such cases, accuracy alone does not provide a reliable measure of model performance.

The main problem addressed in this project is the negative impact of class imbalance on text classification models, particularly how it reduces the model's ability to correctly identify minority classes. Addressing this problem is essential to build more reliable and fair text classification systems.

2. Dataset Description & Imbalance Analysis

2.1 Dataset Description

The dataset used in this study is a text-based emotion classification dataset consisting of short textual statements paired with class sentiments and labels. Each instance contains:

- **Text**: a short sentence expressing an emotional state.
- **Sentiment** : emotion category.
- **Label**: a numerical class identifier representing the emotion category.

The dataset was obtained from Kaggle , and contains 6 emotion categories. Due to natural data collection biases, the distribution of classes is highly imbalanced.

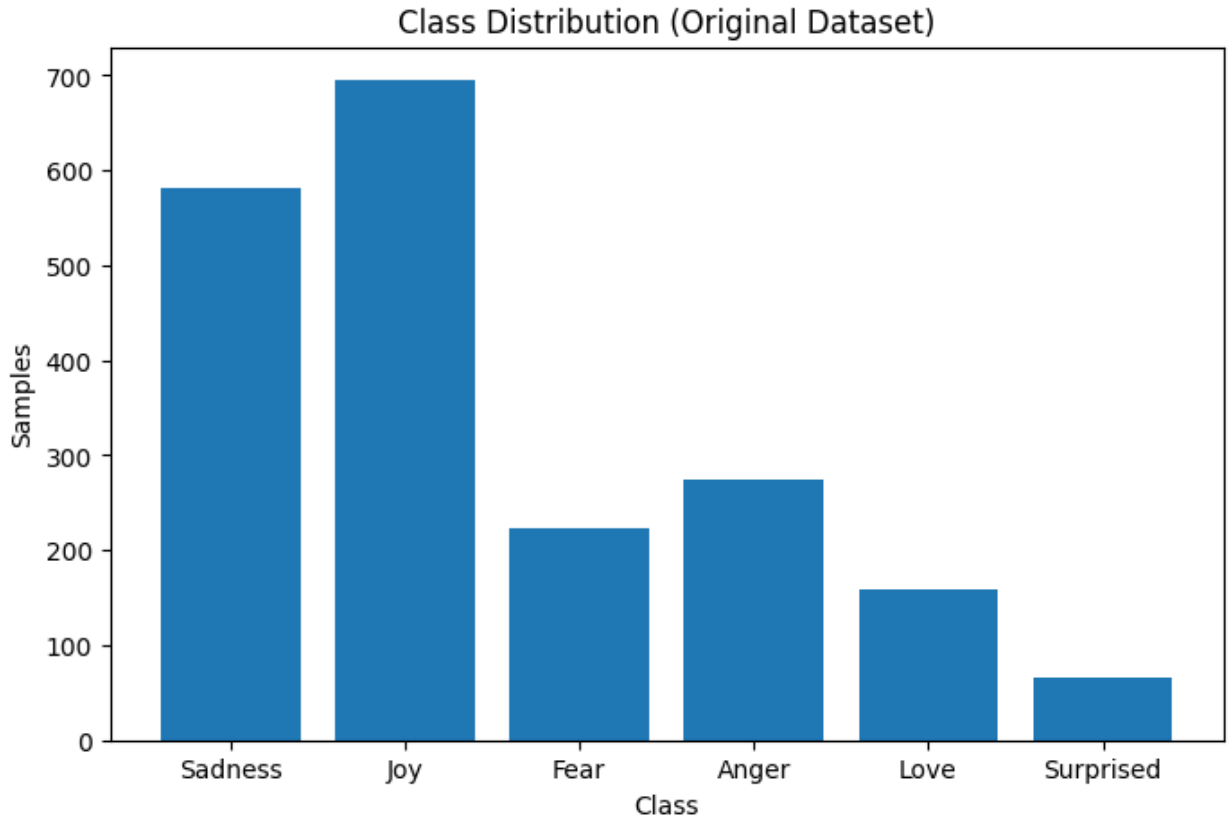
2.2 Preprocessing

preprocessing was performed using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. TF-IDF converts raw text into continuous numerical feature vectors that takes the importance of words while reducing the impact of commonly occurring terms.

2.3 Imbalance Analysis

An analysis of class frequencies revealed a significant imbalance between majority and minority classes. The minority class contains substantially fewer samples than the majority class, making it difficult for standard classifiers to learn representative decision boundaries.

A bar chart visualization of the class distribution of te original dataset clearly shows this imbalance, motivating the need for synthetic data generation. The imbalance ratio directly impacts recall and F1-score for the minority class, which this project aims to improve using GAN-based augmentation.



3. GAN Architectures & Training

To address the class imbalance problem, three GAN-based models were employed to generate synthetic samples for the minority class. All GANs were trained ONLYon minority-class samples to ensure targeted augmentation.

3.1 Vanilla GAN

The Vanilla GAN consists of two fully connected neural networks:

- **Generator**: takes random noise as input and generates synthetic TF-IDF feature vectors.
- **Discriminator**: attempts to distinguish between real and generated samples.

The generator and discriminator are trained adversarially using binary cross-entropy loss. The discriminator learns to classify real versus fake samples, while the generator learns to fool the discriminator by producing realistic minority-class representations.

Despite its simplicity, the Vanilla GAN provides a baseline for understanding GAN-based augmentation. However, it may suffer from training instability and mode collapse.

3.2 Conditional GAN (CGAN)

The Conditional GAN extends the Vanilla GAN by incorporating class label information into both the generator and discriminator. In this implementation, class labels are represented using one-hot encoding and concatenated with the input noise vector and feature vectors.

By conditioning the generation process on class labels, CGAN improves control over the generated samples and encourages better alignment with the minority-class distribution. This typically results in higher-quality synthetic samples and improved classifier performance compared to Vanilla GAN.

3.3 Wasserstein GAN (WGAN)

To further improve training stability, an adapted implementation of the Wasserstein GAN (WGAN) was employed. Unlike traditional GANs, WGAN replaces the discriminator with a critic that estimates the Wasserstein distance between real and generated data distributions.

Key features of WGAN include:

- Removal of the sigmoid activation in the critic

- Use of Wasserstein loss instead of binary cross-entropy
- Weight clipping to enforce Lipschitz continuity

The WGAN implementation was adapted from standard reference architectures and trained exclusively on minority-class samples. This approach provides more stable convergence and smoother gradients during training.

4. Classifier Setup and Evaluation

4.1 Classification Model

A Multi-Layer Perceptron (MLP) classifier was used to evaluate the effectiveness of data augmentation. The classifier consists of:

- An input layer matching the TF-IDF feature dimension
- One hidden layer with ReLU activation
- An output layer with softmax activation for multi-class prediction

4.2 Training Scenarios

To evaluate the effect of data augmentation, two classifiers were employed. A PyTorch-based neural network was used as the primary classifier to allow full control over training and evaluation. Additionally, a Scikit-learn MLP classifier was used as a baseline model to verify the consistency of the results across different learning frameworks.

PyTorch provides flexibility and full control over the training process, including model architecture design, loss computation, optimization, and GPU acceleration, so it was used as the primary deep learning framework

The classifier was trained and evaluated under three distinct scenarios:

1. Original Imbalanced Dataset
2. Dataset Balanced Using Vanilla GAN
3. Dataset Balanced Using CGAN
4. Dataset Balanced Using WGAN

Each model was trained using identical hyperparameters to ensure a fair comparison.

4.3 Evaluation Metrics

Performance was evaluated using multiple metrics:

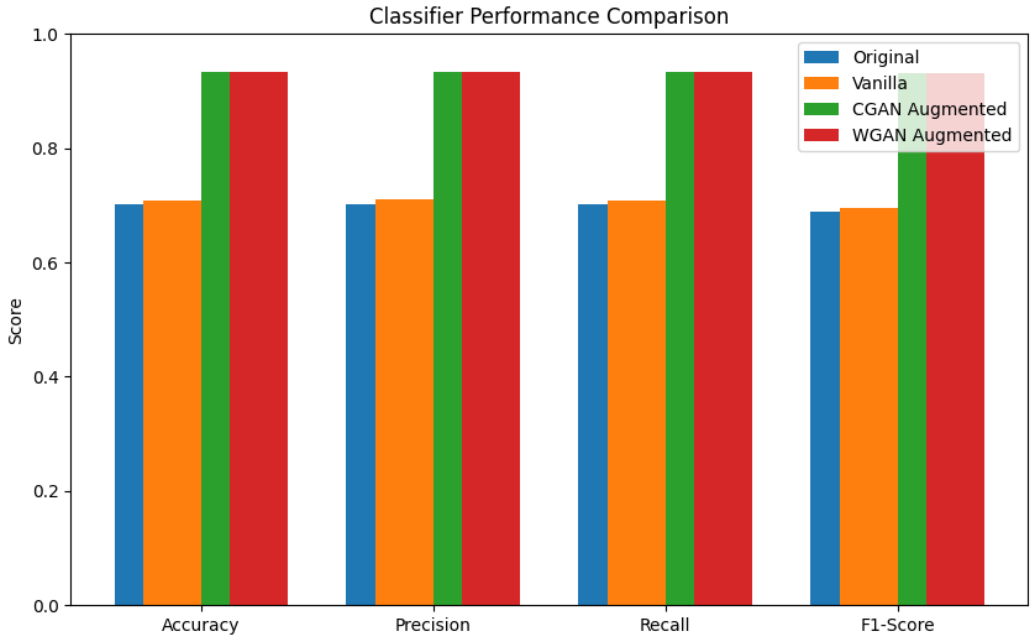
- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

These metrics provide a comprehensive assessment of classifier performance, particularly for the minority class.

5. Results & Comparisons

Experimental results demonstrate that GAN-based augmentation significantly improves classification performance on the minority class.

- The imbalanced dataset showed high accuracy but poor recall and F1-score for the minority class.
- Vanilla GAN augmentation improved minority recall but exhibited moderate instability.
- CGAN and WGAN augmentation achieved the best overall performance.



Confusion matrix visualizations further confirm that GAN-based augmentation reduces misclassification of minority-class samples. Among all approaches, WGAN provided the most stable and consistent improvement due to its robust training dynamics.

6. Observations and Conclusions

This study demonstrates that GAN-based data augmentation is an effective strategy for addressing class imbalance in text classification tasks. Training GANs exclusively on minority-class samples enables the generation of realistic synthetic data that enhances classifier learning.

Key observations include:

- Vanilla GAN provides a simple but limited baseline.
- CGAN improves sample quality through label conditioning.
- WGAN offers superior training stability and performance.

Overall, GAN-based augmentation significantly improves minority-class recall and F1-score without sacrificing overall accuracy. Future work may explore transformer-based text representations or diffusion models for further performance gains.