Special Topics in DS Project

University of Jordan

King Abdullah II School for Information Technology

Special Topics in Course

Semester 2025/2026

Supervisor: Dr. Yousef Sanjalawe

Table of Contents

Problem statement

Detecting hate speech on social media is an important yet challenging text classification task. One of the main difficulties lies in the highly imbalanced nature of the data, where non-hate content significantly outnumbers hate speech samples. This imbalance makes the classification process particularly challenging and affects the reliability of standard evaluation metrics.

As a result of this skewed distribution, classification models often achieve high accuracy while failing to correctly identify hate speech instances. This leads to misleading performance results, especially reflected in poor recall and F1-score values for the minority class.

To address this issue, this project employs Generative Adversarial Networks (GANs) to generate synthetic hate speech samples and balance the dataset. Both a Vanilla GAN and a Conditional GAN (CGAN) were implemented to augment the minority class. The effectiveness of this approach was evaluated by comparing classification performance on the original imbalanced dataset and the balanced datasets.

Description of Dataset & Imbalance Analysis

Dataset Name: Hate Speech and Offensive Language Dataset

Source: Kaggle – Hate Speech and Offensive Language Dataset

Number of Records: 24,783 labeled text samples collected from social media.

Class Labels (Binary Classification):

Not Hate: 23,353 samples (offensive language + neither)

Hate: 1,435 samples

Imbalance Description:

The dataset exhibits a severe class imbalance, where hate speech samples form a small minority compared to non-hate samples. This significant imbalance impacts classification performance and motivates the use of GAN-based data augmentation techniques.

Imbalance Visualization:

This clear imbalance is visualized using a class distribution plot shown below:

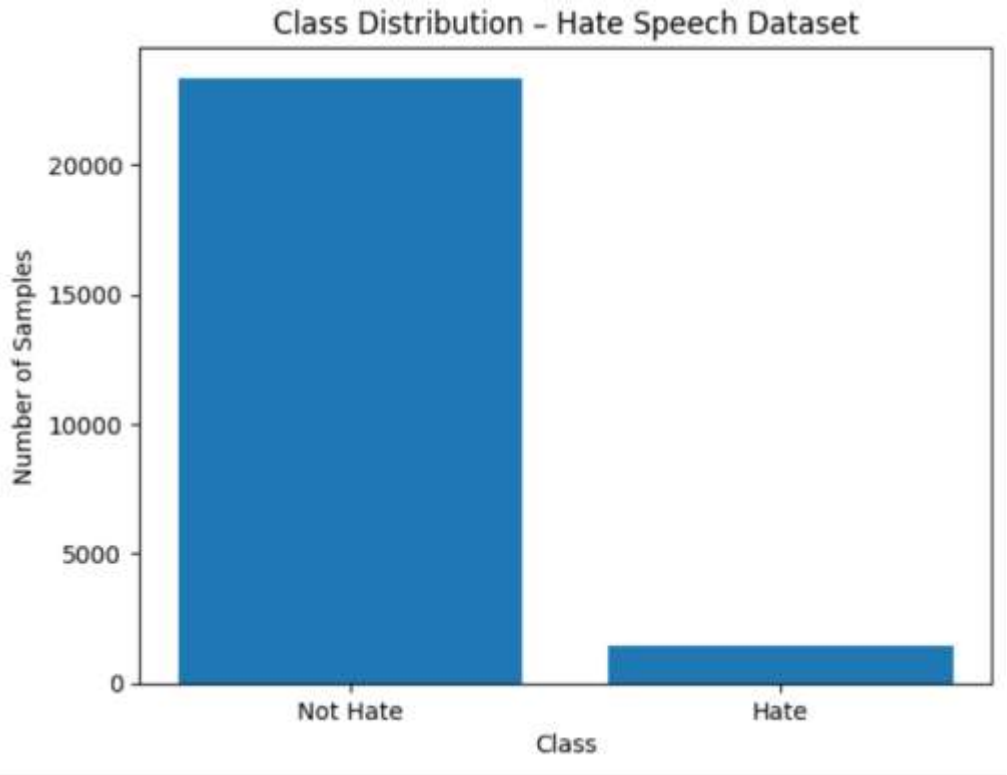

Figure 1 Data distribution

Details of GAN Architectures & Training

Two types of Generative Adversarial Networks (GANs) were implemented in this project: **Vanilla GAN** and **Conditional GAN (CGAN)**. Both models were designed to generate synthetic samples for the minority class (hate speech) in order to address class imbalance.

Vanilla GAN

The Vanilla GAN architecture consists of two neural networks: a generator and a discriminator. The generator learns to produce synthetic hate speech feature vectors, while the discriminator attempts to distinguish between real and generated samples. Both networks are trained in a minimax game, where the generator minimizes its loss by fooling the discriminator, and the discriminator maximizes its ability to distinguish real samples from generated ones.

The generator is implemented using two fully connected layers (64 hidden units followed by a sigmoid output layer), while the discriminator also consists of two fully connected layers (64 hidden units followed by a sigmoid output neuron for binary classification).

```
# Generator
def build_generator():
    model = tf.keras.Sequential([
```

```
        layers.Input(shape=(latent_dim,)),
        layers.Dense(64, activation='relu'),
        layers.Dense(input_dim, activation='sigmoid')
    ])
    return model

# Discriminator
def build_discriminator():
    model = tf.keras.Sequential([
        layers.Input(shape=(input_dim,)),
        layers.Dense(64, activation='relu'),
        layers.Dense(1, activation='sigmoid')  # real vs fake
    ])
    return model
```

- Generator: Two dense layers (64 units + sigmoid)

- Discriminator: Two dense layers (64 units + sigmoid)

Minimax game:

The Generator attempts to deceive the Discriminator by producing fake data that appears real, thus **minimizing** the loss.

The Discriminator attempts to distinguish real data from fake data, thus **maximizing** the Generator's loss.

In addition to Vanilla GAN and CGAN, a **Wasserstein GAN (WGAN)** was implemented as an alternative GAN variant to improve training stability. Unlike the standard GAN, WGAN replaces the binary cross-entropy loss with the **Wasserstein distance**, which provides smoother gradients and reduces common training issues such as mode collapse.

In WGAN, the discriminator is replaced by a **critic** that outputs a real-valued score instead of a probability. The critic is trained to maximize the Wasserstein distance between real and generated samples, while the generator aims to minimize this distance. Weight clipping was applied to enforce the Lipschitz constraint required by the Wasserstein formulation.

The WGAN model was trained on the minority class (hate speech) using the same TF-IDF feature representation and training settings as the other GAN models.

HYPERPARAMETERS	DESCRIPTION	VALUE
BATCH SIZE	Number of samples per training step	32
LEARNING RATE	Controls the weight update speed	0.0002
EPOCHS	Number of training iterations	1000

OPTIMIZER	Optimization algorithm	Adam
LOSS FUNCTION (VANILLA GAN & CGAN)	Adversarial loss used for GAN training	Binary Crossentropy
LOSS FUNCTION (WGAN)	Wasserstein loss	Wasserstein Distance
WEIGHT CLIPPING	Lipschitz constraint	0.01

Table 1 Hyperparameters

Training Strategy:

Both GAN models were trained using TF-IDF feature representations of the text data. The generator receives a random noise vector as input and outputs a TF-IDF–like feature vector representing synthetic hate speech samples.

Classifier Setup and Evaluation

An MLP classifier was applied to determine the impact of GAN-based data augmentation on hate speech detection. The text samples were represented by means of TF-IDF features and the dataset was divided into the training set and the testing set following the 80/20 ratio.

The classifier was trained and tested in three different situations: first, the original imbalanced dataset, second, the dataset balanced through Vanilla GAN, and third, the dataset balanced through CGAN. To measure performance, various metrics such as Accuracy, Precision, Recall, F1-score, AUC-ROC, and Confusion Matrix were used, with the last one particularly focused on the improvement in minority class detection.

The classifier was trained and evaluated under the three previously described scenarios, and the results are shown below:

	DATASET	ACCURACY	PRECISION	RECALL	F1-SCORE	AUC-ROC	CONFUSION MATRIX
0	Original Imbalanced	0.935243	0.266667	0.069930	0.110803	0.733358	[[4616, 55], [266, 20]]
1	Vanilla GAN Balanced	0.964140	0.982421	0.945194	0.963448	0.983392	[[4592, 79], [256, 4415]]
2	CGAN Balanced	0.963498	0.981111	0.945194	0.962818	0.983703	[[4586, 85], [256, 4415]]

Table 2 Results for three scenarios

Here are some images of the confusion matrix in three scenarios:

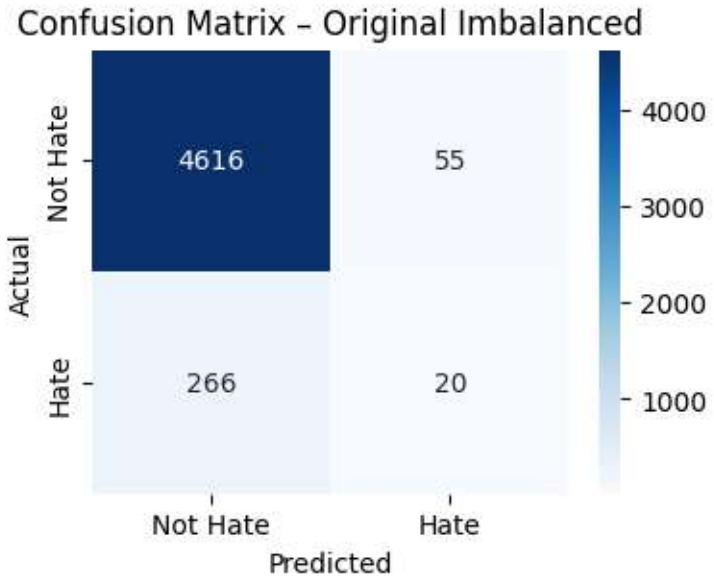

Figure 2 Confusion Matrix – Original Imbalanced

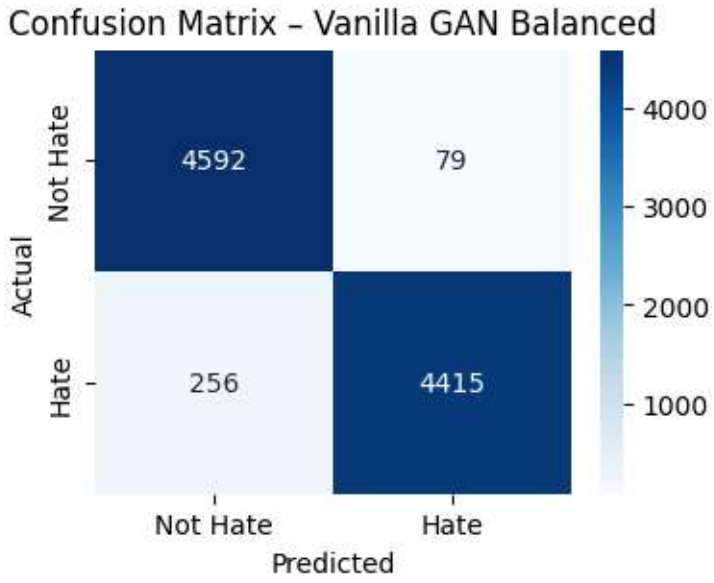

Figure 3 Confusion Matrix – Vanilla GAN Balanced

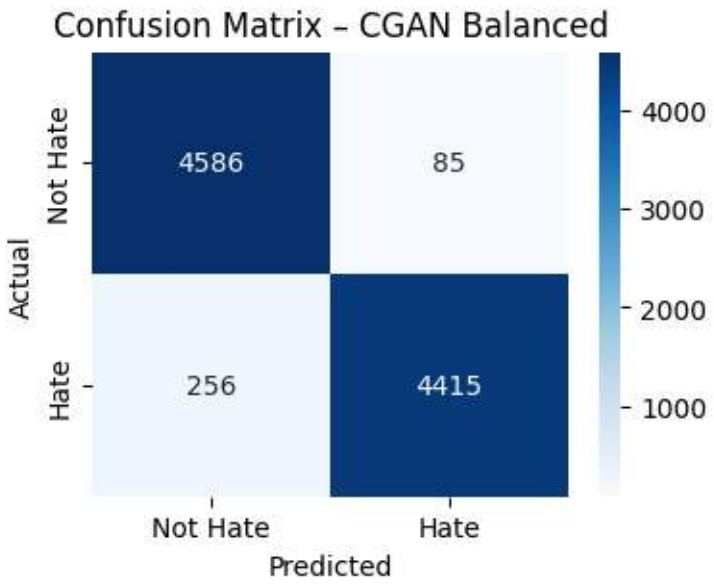

Figure 4 Confusion Matrix – CGAN Balanced

Comparison of classification performance:

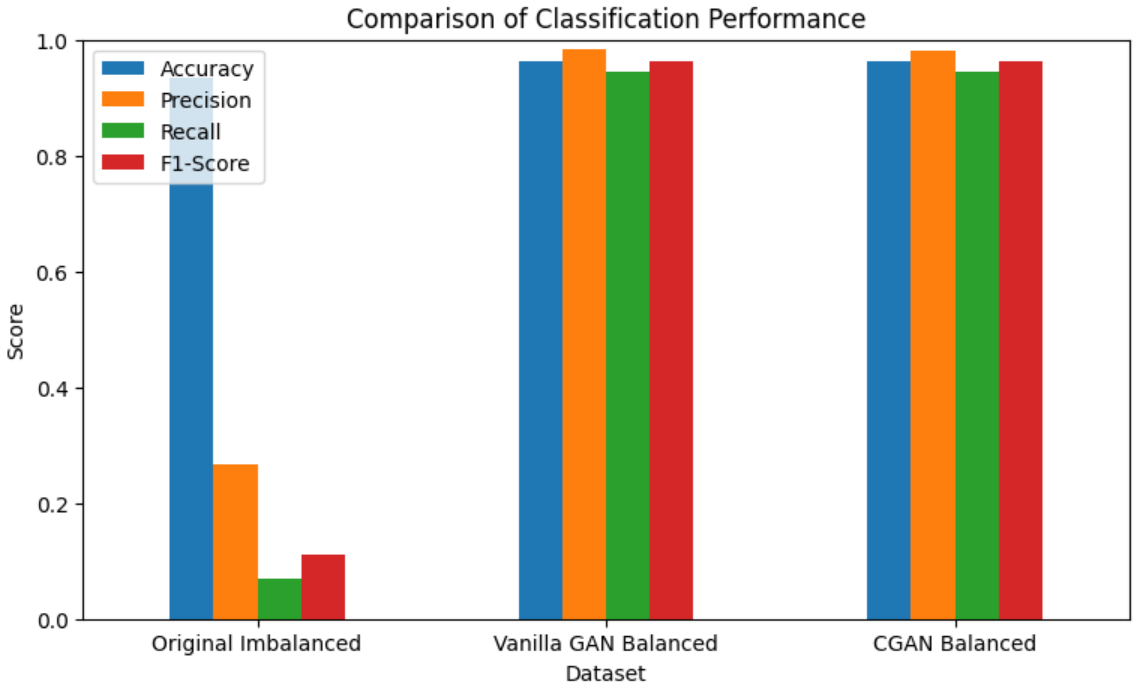

Figure 5 Comparison of the three scenarios

Observations and Conclusions

Despite the high accuracy achieved on the original dataset, class imbalance significantly limited the classifier's ability to detect hate speech. GAN-based data augmentation substantially improved recall and F1-score, confirming the effectiveness of GANs for imbalanced text classification tasks.