Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi
Pamukkale University Journal of Engineering Sciences

Performance comparison of data balancing techniques on hate speech detection in Turkish

Türkçe nefret söylemi tespitinde veri dengeleme tekniklerinin performans karşılaştırması

Habibe KARAYİĞİT[1], Ali AKDALI[*], Çiğdem ACI[3]

[1]Ministry of National Education, Adana, Turkey.
habibe_devrim@hotmail.com
[2]Department of Electrical and Electronics Engineering, Mersin University, Mersin, Turkey.
aliakdagli@gmail.com
[3]Department of Computer Engineering, Mersin University, Mersin, Turkey.
caci@mersin.edu.tr

Received/Geliş Tarihi: 13.03.2023 Revision/Düzeltme Tarihi: 17.08.2023 doi: 10.5505/pajes.2023.40072
Accepted/Kabul Tarihi: 10.10.2023 Research Article/Araştırma Makalesi

Abstract

Increasing hate speech on social media platforms causes psychological disorders and deep and negative effects. Automatic language classification models are needed to detect hate speech. When testing language models for hate speech, imbalanced datasets where one data class is represented much more frequently than the other can be a problem in language datasets. When the dataset is imbalanced, the classifier may be biased towards the majority class and may not perform well in the minority class. This can lead to incorrect or unreliable classification results. To solve this problem, data level balancing methods such as oversampling or undersampling are used to balance the class distribution before classifying the dataset. This study, it is aimed to achieve a successful classification model combination that detects hate speech by using data-level balancing methods. For this, a comprehensive study was carried out by applying the balancing method at eight data levels (random oversampling, Synthetic Minority Oversampling Technique (SMOTE), K-means SMOTE, Localized Random Affine Shadow Sample (LoRAS), Text-based Generative Adversarial Network (TextGAN), Nearmiss, Tomek Links ve Clustering-based) to the Abusive Turkish Comments (ATC) dataset, which has an imbalanced distribution of labels, obtained from Instagram. Classification performances of data level balancing methods were evaluated with Basic Machine Learning (BML) and Convolutional Neural Network (CNN) methods. It has been observed that the CBoW+CNN model based on the TextGAN data-level balancing method, as well as the Skip-gram CNN model, exhibited the best classification performance with a Macro-Averaged F1 score of 0.972.

Keywords: Data balancing, Social media, Machine learning, Deep learning, Natural language processing, Hate speech.

Öz

Sosyal medya platformlarında artan nefret söylemleri, psikolojik rahatsızlıklara, derin ve olumsuz etkilere neden olmaktadır. Nefret söylemlerini tespit etmek için otomatik dil sınıflandırma modellerine ihtiyaç vardır. Nefret söylemleri için dil modelleri test edilirken, bir veri sınıfının diğerinden çok daha sık temsil edildiği dengesiz veri kümeleri dil verilerinde sorun teşkil edebilir. Veri kümesi dengesiz dağılıma sahip olduğunda, sınıflandırıcı çoğunluk sınıfına yönelik önyargılı olabilir ve azınlık sınıfında iyi performans göstermeyebilir. Bu, yanlış veya güvenilmez sınıflandırma sonuçlarına yol açabilir. Bu sorunu çözmek için veri kümesi sınıflandırılmadan önce oversampling veya undersampling gibi veri düzeyi dengeleme yöntemleri ile veri sınıfları dengelenir. Bu çalışmada, veri düzeyi dengeleme yöntemleri kullanılarak nefret söylemini tespit eden başarılı bir sınıflandırma modeli kombinasyonu elde etmek amaçlanmaktadır. Bu amaçla, Instagram'dan elde edilmiş dengesiz etiket dağılımına sahip Abusive Turkish Comments (ATC) veri kümesine sekiz veri düzeyinde (rastgele oversampling, Synthetic Minority Oversampling Technique (SMOTE), K-means SMOTE, Localized Random Affine Shadow Sample (LoRAS), Text-based Generative Adversarial Network (TextGAN), Nearmiss, Tomek Links ve Clustering-based) dengeleme yöntemi uygulanarak kapsamlı bir çalışma yapılmıştır. Veri düzeyi dengeleme yöntemlerinin sınıflandırma performansları Basic Machine Learning (BML) ve Convolutional Neural Network (CNN) yöntemleriyle değerlendirilmiştir. TextGAN veri düzeyi dengeleme yöntemine dayalı CBoW+CNN modelinin ve Skip-gram CNN modelinin 0,972 Makro Ortalamalı F1 puanı ile en iyi sınıflandırma performansını sergilediği görülmüştür.

Anahtar kelimeler: Veri dengeleme, Sosyal medya, Makine öğrenmesi, Derin öğrenme, Doğal dil işleme, Nefret söylemi.

1 Introduction

People can express themselves freely on social media as long as they don't bother others. Utilizing social media frequently has made people lose measure in their expressions, which has led to an increase in hate speech. Speech that denigrates individuals and seeks to denigrate them in public is considered hate speech. The three targets of hate speech are race, religion, and gender. The detrimental impacts of hate speech on people's mental health make it a serious issue. People who are exposed to hate speech frequently experience trauma, sadness, worse academic achievement, and asocial behavior [1] . Hate speech is accepted as illegal, and several nations are attempting to enact legislation to combat it. Social media users have noted that these steps are taken but that they are insufficiently deterrent [2].

Finding methods to identify and evaluate hate speech content has become more crucial as hate speech on the internet and in social media has grown. However, it is known that most of the previous studies on the detection of hate speech are in English [3-6] and there are a limited number of Turkish studies [7-8] on this subject. To assess hate speech in Turkish and test balancing

*Corresponding author/Yazışılan Yazar

approaches in an imbalanced dataset, the Abusive Turkish Comments (ATC) dataset from Instagram was selected as the data source for this study. Instagram has been one of the most widely used social media platforms since October 2010 since it makes sharing simple for users. With more than one billion monthly active users globally, Instagram represents a sizeable share of the global population [9].

Large-scale data analysis using tested effective machine learning techniques can spot patterns and trends that point to hate speech in social media [6,10]. By receiving training on datasets that are precisely categorized according to various forms of hate speech, machine learning algorithms may accurately detect hate speech in uncategorized data. In recent years, it has been found that Deep Learning (DL) algorithms, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNN), are efficient at identifying hate speech because they can handle enormous amounts of unstructured text, like social media posts [6] . Thus, a greater spectrum of hate speech detection is possible thanks to these algorithms [11-12].

Although machine learning and deep learning-based algorithms can reach a high level of prediction accuracy, as a result of the algorithm's potential bias in favor of the majority class, classification results obtained from imbalanced datasets may not be reliable [13]. The approach may have trouble correctly identifying the minority class in imbalanced datasets when the number of samples in one class is significantly higher than the number of samples in the other classes. This can be explained by the fact that the algorithm will likely be trained mostly on samples from the majority class and may not have sufficient knowledge about the minority class to make reliable predictions.

Natural Language Processing (NLP) techniques frequently encounter imbalanced datasets, however, there are numerous ways to address this issue, including cost-sensitive learning and data-level balancing. These techniques have been applied in a number of NLP applications, including hate speech detection, and have been proven to increase classification accuracy in unstable datasets. Resampling, which involves oversampling the minority class or undersampling the majority class, is a common method for data-level balance. To achieve this, instances from the minority class may be repeated at random, or examples from the majority class may be dropped.

Synthetic Minority Oversampling Technique (SMOTE), one of the most often used oversampling techniques, creates synthetic samples for the minority class by interpolating between existing examples [14]. The decision threshold for the classifier or the cost function is both modified as part of the cost-sensitive learning strategy to account for class imbalance. A cost-sensitive decision tree, for instance, will place a higher cost on misclassifying samples from the minority class in order to motivate the classifier to pay closer attention to the minority class [15].

In this study, the effectiveness of several data-level balancing techniques on machine learning models' classification performance was examined as well as oversampling, undersampling, and SMOTE. The imbalanced ATC dataset was utilized which is made up of Instagram comments, with 33% of the categories being abusive and 66% of the categories being non-abusive. In addition to a CNN model, the classification performance of Basic Machine Learning (BML) models including Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest (RF) was examined. Random oversampling, SMOTE, K-means SMOTE, Localized Random Affine Shadow Sample (LoRAS), Text-based Generative Adversarial Network (TextGAN), Nearmiss, Tomek Links, and Clustering-based were used as the balancing techniques. BML classifiers use feature extraction techniques including Bag of Words (BoW), Term Frequency — Inverse Document Frequency (TF-IDF), and word n-grams with TF-IDF, whereas the CNN model uses Word2Vec (SkipGram and CBoW) word embedding algorithms.

The contributions of the study can be listed as follows; (1) the study is the first to examine the impact of balancing methods on various data levels on hate speech in Turkish. (2) The TextGAN oversampling strategy produced better classification outcomes for the identification of hate speech on social media was shown. (3) It was discovered that CNN-based classifiers provide models with more classification success when using data balancing techniques than BML-based classifiers.

The rest of the paper is structured as follows: Chapter 2 presents earlier research on the subject. The ATC dataset and the research techniques are detailed in Chapter 3. The experimental findings are reported and analyzed in Chapter 4. Chapter 5 concludes the study and provides a summary of its findings.

2 Related works

Hate speech detection in social media has become a very popular area of research in recent years. In a study using the social network Twitter was used to collect data and the data were divided into three groups (hate speech, offensive language, and neutral) [16]. Another dataset which consisted of 6,000 tweets across various hate categories (including religion, sexual orientation, gender, and ethnic minority) was studied in [17]. A study [18] presented a collection of approximately 30,000 tweets obtained from Twitter in which hate expressions were divided into three categories: Sexual orientation, gender, and ethnicity. In previous studies, the datasets collected for hate speech are primarily in English, but there are also datasets about hate speech in German [5,19], Italian [20], and Turkish [7] languages.

BML-based classifiers (i.e. NB, Logistic Regression (LR), SVM, and RF) with BoW, n-gram, syntactic, and linguistic techniques are widely utilized in the detection of hate speech [16,21-24]. Due to their superior performance over BML-based algorithms, DL-based algorithms are also employed in hate speech analysis. The usage of feature extraction techniques such as One-Hot Encoding, Word2Vec, Global Vectors for Word Representation (GloVe), Bidirectional Encoder Representations from Transformers (BERT) with Long Short-Term Memory Networks (LSTM), GRU, and Bidirectional LSTM (BiLSTM) are also popular [7, 12, 20, 26].

At the data level, various strategies have been put forth to enhance the classification performance of imbalanced datasets. Resampling algorithms, such as the Adaptive Synthetic Sampling Approach for Imbalanced Learning algorithm, which generates synthetic data in accordance with data density, were tested in a sentiment analysis study, and it was discovered that the success of classification increased by nearly 50% [27]. A study that uses SMOTE to identify cyberbullying on an imbalanced dataset acquired from Twitter boosted the amount of data by locating the n-nearest neighbors of the minority class samples in the train set [28]. The k-means SMOTE algorithm emerged as the most popular SMOTE technique in another investigation that included 85 SMOTE versions [29]. The TextGAN algorithm, which can generate realistic phrases for

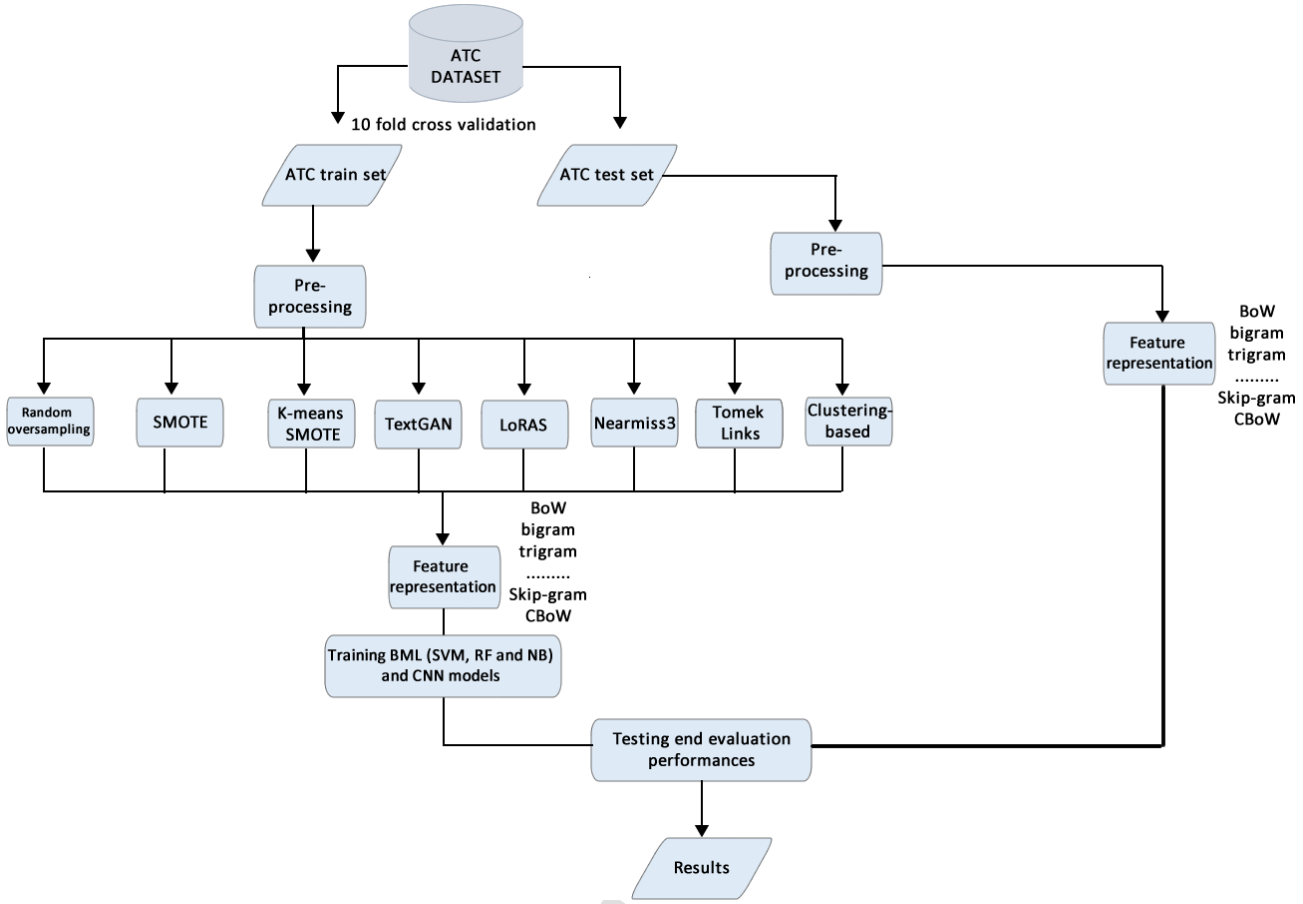

Figure 1. Block diagram of the proposed approach.

Şekil 1. Önerilen yaklaşımın blok diyagramı.

the oversampling approach of the samples in the dataset received from Twitter was used in a study that produced highly successful results [30]. Resampling combination strategies using DT and NB classifiers were found to improve classification performance more than other models in a study where undersampling, oversampling, and combination techniques were used to regulate the class distributions of imbalanced datasets [31]. In a different study, minority class URLs were used to train text generator competitor networks (TextGAN), and fictitious URLs were created as part of the train set [32].

In a study [33], in order to stop the majority class from misclassifying data with SMOTE, the minority class was oversampled with LoRAS, whose linear combination coefficients were chosen at random from a Dirichlet distribution. The imbalanced dataset utilized in a study for abusive speech analysis was balanced using the random oversampling method [7]. The idea of the density of the minority class is inversely correlated with the synthetic creation of data was found in a study utilizing Decision Tree (DT) and LR classifiers. It is shown that G-Means criteria recognition along with oversampling methods can enhance classification performance [34]. Cost-sensitive learning approaches with LSTM and BiLSTM performed better in classification tests of balancing methods based on data-level resampling techniques, cost-sensitive learning, and weight selection strategies than other model combinations on imbalanced datasets [13]. The effectiveness of the SMOTE approach on unbalanced text features was investigated for the purpose of detecting poisonous hate speech, and it was found that the RVVC model outperformed the SMOTE method on unbalanced labeled data [35]. Two state-of-the-art text creation GAN models, CatGAN and SentiGAN, were utilized to measure the impact of synthetic text production for sensitivity analysis. According to reports, the models' classification performance has significantly improved [36]. In order to detect spam using the unbalanced data set acquired from social media, a framework model was developed. Using imbalanced and balanced datasets, basic ML models, voting-based community models, and deep learning-based hybrid systems were ass essed using NearMiss and SmoteTomek methods [37]. Using a built generative contentious neural network, the minority class KNNGAN approach was used to produce numerous synthetic examples after the K-nearest neighbor (KNN) algorithm had identified noisy data and GAN [38].

3 Materials and methods

This study aims to select a successful hybrid model combination for the detection of hate speech in the imbalanced class-distributed ATC dataset utilizing text classification algorithms and data-level balancing algorithms. Figure 1 shows the block diagram of the approach. The following steps are taken to implement this approach: (1) Separate the ATC dataset into a train set and test set; (2) Pre-process the ATC train set and ATC test set; and (3) Apply different data-based balancing techniques (i.e. random oversampling, SMOTE, k-means SMOTE, LoRAS, TextGAN, Nearmiss, Tomek Links, and

Clustering-based), (4) Implement feature extraction (i.e. BoW, bi-gram, tri-gram, and Word2Vec) algorithms (5) Evaluation of classification results with Macro-Averaged F1 metric on balanced datasets using different classifiers (i.e. SVM, RF, NB, and CNN).

3.1 The Dataset

The family of languages known as the Altaic includes Turkish. Due to its structure and inflectional suffixes, the Turkish language has an agglutinative morphology [39]. With the suffixes it acquires, the word may grow into one or more words that differ from the meaning of the stem form. In our study, 10,528 abusive and 19,826 non-abusive comments in Turkish from Instagram between 2017 and 2019 were collected to form the ATC dataset [7]. According to the Turkish slang vocabulary, each sample in the ATC dataset was manually classified as either abusive or not abusive [40]. The number of comments gathered shows that the ATC dataset has an imbalanced structure when it comes to the distribution of categories.

Eight different data-level balancing techniques were individually used to the ATC train data to equalize the categories (abusive and non-abusive). The cross-validation method was then used ten times, with test data being utilized once for each of the ten subsamples [41]. Then, the overall result was an average of the ten results.

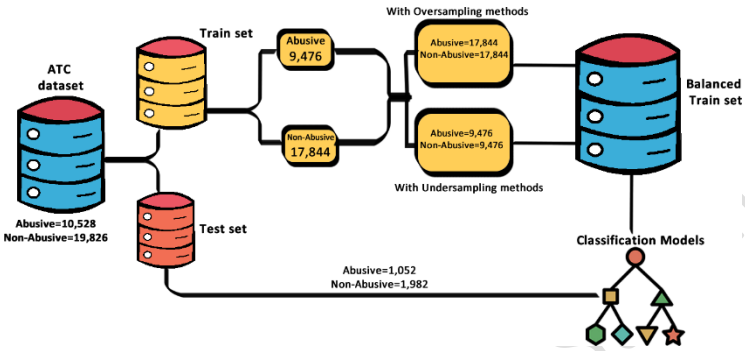

Figure 2. Balancing the number of comments in the ATC dataset through the application of sampling methods.

Şekil 2. ATC veri kümesindeki yorumların sayısını örnekleme yöntemleri uygulayarak dengeleme.

As seen in Figure 2, after dividing the ATC dataset with 10 cross-folds, the number of Abusive comments in the train set is 9,476 and the number of non-Abusive comments is 17,844. After applying oversampling techniques, the number of abusive comments is 17,844 and the number of non-Abusive comments is17,844. After the underampling techniques are applied, the number of Abusive comments is 9,476 and the number of non-Abusive comments is 9,476. In the test set, the number of Abusive comments is 1,052 and the number of non-Abusive comments is 1,982.

3.2 Pre-processing

Punctuations, hashtags, URLs, and other formattings were removed from the ATC dataset to make the dataset ready for tokenization and a lowercase data conversion. Turkish stop-words that were unnecessary in terms of meaning were removed from the ATC dataset at the last step.

3.3 Resampling algorithms

In datasets with an uneven distribution of classes, classifiers are sensitive to an equal number of classes and frequently misclassify examples of those classes [31]. Resampling the data is one way to get a good and realistic classifier performance for imbalanced datasets [42]. Resampling is the process of altering several samples in a dataset by either raising or decreasing the minority classes (oversampling) or the majority classes (undersampling). This study used Nearmiss, Tomek Links, and Clustering-based Undersampling methods from undersampling methods and Random Oversampling, SMOTE, K-means SMOTE, LoRAS, and TextGAN methods from oversampling methods.

3.3.1 Oversampling algorithms

The minority category in a dataset is increased via the oversampling method until it is equal to the majority category. The SMOTE method generates synthetic data to supplement the minority category data. The minority class's k-nearest neighbors are chosen using the SMOTE technique, and the synthetic minority class data points are increased along the line segments that connect them. The k-means SMOTE approach uses the k-means algorithm to cluster all of the categorical samples. It then filters out any clusters that have a sample from a minority class and uses the SMOTE method to artificially boost the new samples in the cluster in order to modify the data distribution. With affine linear combinations of instances from the minority class, LoRAS creates new instances. A generative and discriminative oversampling technique called GAN creates synthetic data that is realistic (has genuine value). By removing the issues with convergence and discrete inputs in GANs, TextGAN creates synthetic text. A short-term memory network and a convolutional network are used as the generator and allocator, respectively, in the paradigm suggested by TextGANs [30].

3.3.2 Undersampling algorithms

The dominant category in a dataset is reduced via the undersampling approach to match the minority category. Using the k-nearest neighbor (k-NN) algorithm, the near-miss approach, one of the undersampling techniques, seeks to stabilize the class distribution by randomly eliminating the majority of class samples. The study's Nearmiss-3 technique initially identifies each minority sample's closest neighbors. Second, the majority selects its samples based on the median distance between the neighbors it deems to be closest. Compared to other Nearmiss methods (i.e. Nearmiss-1, and Nearmiss-2), the Nearmiss-3 method is less impacted by noise.

The Tomek Links undersampling technique eliminates instances from the majority class that are noisy or on the edge of it. The Tomek Links approach in the train set reduces the size of the majority class by substituting cluster centers for the original data in the same groups. The clustering-based strategy generates centroids based on clustering techniques, which undersamples the data in the train set. In order to secure the data, this method groups the datasets based on how similar they are before undersampling.

3.3.3 Feature extraction algorithms

Feature extraction is a very important step in text mining, and it makes it easier to extract relevant information from datasets suitable for the desired result by ignoring unwanted information [43]. The dataset is handled by the BoW algorithm as a collection of randomly arranged words. Word frequency is typically used to sort utilizing the BoW representation. The set of countable words that make up a document (d) is represented by the BoW, which is determined as the sum of the single-word vectors that comprise the document. The resulting feature

vector generates the content of a document, but it grows fast in size with the size of the dictionary and ignores the semantics of the text [44].

The value obtained by dividing the chosen word by the total number of words in a phrase is known as Term Frequency (TF). The importance of the chosen term among all the comments in the dataset is depicted by the Inverse Document Frequency (IDF). It is calculated by dividing the logarithm of the total comments by the number of words that contain the term [37]. The TF and IDF results are multiplied to provide the TF-IDF value. In this study, TF-IDF and BoW were used to create BML classifiers.

The feature extraction methods that are often employed in language studies, such as sentiment analysis and the identification of hate speech, include word n-gram algorithms [37]. Before a word is analyzed, word n-gram models indicate whether n-1 words will be present. In this study, TF-IDF and word-based n-gram models, with n-word lengths ranging from 2 to 3, were applied to BML classifiers.

The use of words with other words is taken into account by word embedding techniques, which locate words in a vector space [45]. A neural network called the word embedding algorithm uses the data from the Word2Vec input layer to generate vector representations of the data as output. The Word2Vec method is a two-layer predictive model that predicts words based on their weights rather than their quantity [45]. Skip-gram and CBoW are the two sub-models of the Word2Vec method. The CBoW model outputs the neighbors of the desired word and attempts to predict a word based on those neighbors. The Skip-gram method tries to infer the input word's neighbors. The CNN classifier was used in conjunction with the CBoW and Skip-gram feature extractions in this investigation to produce classification results.

3.3.4 BML-based and CNN-based classifiers

Classification is a methodology that involves categorizing a given text into one or more classes [46]. In the study, information on the text classification algorithms employed is concisely presented below.

The Bayes theorem, which presupposes independent qualities, is used by the NB classifier, which functions according to conditional probability logic. Due to its ease of use and quick classification, the NB method is commonly employed in natural language research, such as hate analysis [47]. The SVM classifier, which produces excellent classification outcomes for highly dimensional text input, locates the plane that will maximize class separation. Depending on the number of categories with a hyper-plane, datasets are split into two or more classes. The margin, also known as the distance separating the two classes, should be as wide as feasible to reduce classification error [48]. The overfitting issue with DT algorithms is resolved by the RF algorithm, a classifier created by integrating the classification results provided by many DTs. The dataset and feature set are divided into various subsets that are randomly chosen and trained by the RF model [49].

Convolution, pooling, and fully connected layers originate CNN's feed-forward network approach [50]. The attributes of the sample texts are extracted based on the kernel chosen in the convolution layer. The hyperparameter values of CNN-based and BML-based classifiers used in this study are given in Table 1. The grid-search structure for BML-based classifiers was utilized to select the hyper-parameters. The CNN-based classifier's hyperparameters were chosen using the trial-and-error method. Given that there are two categories in this study, the output in the fully connected layer of the CNN-based classifier was set to 2.

Table 1. The hyperparameter values of CNN-based and BML-based classifiers.

Tablo 1. CNN tabanlı ve BML tabanlı sınıflandırıcıların hiperparametre değerleri.

Classifier	Hyperparameters
SVM	C = 10.01, LinearSVC() function was used
RF	Number of decision trees = 50
NB	alpha = 0.1
CNN	Convolutional layer: Three 1D convolution layers Kernel size: 2,3,4 Padding: same Activation: ReLu and Sigmoid Filters: 100,50,50 Pooling: GlobalMaxPooling1D Dense Layer: The first dense layer has 512 neurons, and the second one has 1 neuron Optimizer: ADAM Loss: binary_crossentropy Epochs: 15 Batch Size:128

Macro-Averaged F1 was chosen as the evaluation metric [51] of classifiers. In Macro-Averaged F1, each label is given a measured value, and the average is determined by the total number of labels in the dataset [7]. The Macro-Averaged F1 measure supports the performance of the few categories and generates classification results by assigning the same weight value to each category in the sample [52]. Below is the Macro-Averaged F1 Equation (1).

$$\text{Macro-Averaged F1} = \frac{1}{|\text{Classes}|} \cdot \sum_{i \in \text{Classes}} \text{F1}(i) \quad (1)$$

4 Experimental results

In this section, the classification results obtained from the combination of various data level balancing techniques, feature extraction algorithms and different classifiers are evaluated in order to analyze Turkish hate speech. After applying data-level balancing techniques to the dataset, BoW+TF-IDF, bigram+TF-IDF, trigram+TF-IDF, and Word2Vec (i.e. CBoW and Skip-gram) algorithms were used for data representation. After that, experimental evaluations were performed by using DL-based and BML-based classifiers. Table 2 through Table 5 present the Macro-Averaged F1 results of DL-based and BML-based classifiers which were combined with eight different balancing techniques. The Google Colab [53] was utilized to create the models' code environment using the Phyton programming language [54].

As is shown in Table 2, the TextGAN oversampling balancing strategy along with the combination of BoW+TF-IDF+SVM led to the best classification success. The Macro-Averaged F1 value achieved by LoRAS sampling approach was the lowest for the SVM classifier.

The TexGAN+bigram+TF-IDF+RF model, in which the RF classifier was utilized, performed the best value for the sampling balancing approach, as is shown in Table 3. On the other hand, the LoRAS sampling approach yielded the lowest Macro-Averaged F1 value for classification success among the RF classification models.

According to Table 4, when combined with feature extractions from the TextGAN method, the NB classifier model produced the best values. On the other hand, just like in the other RF and SVM classification models, the LoRAS method became a balancing technique that reached the lowest values for the NB classifier.

As observed in Table 5, utilizing CBoW and Skip-gram embedding methods in CNN models, the optimal classification results among both BML-based and CNN-based models were achieved through the TextGAN oversampling approach. In contrast to the other models, the clustering-based undersampling technique yielded the lowest outcome for the CNN classifier. The overall findings demonstrate that the TextGAN algorithm attains a higher classification success rate than all the data-level balancing techniques tested in this study.

Table 6 demonstrates the highest achieved classification performance utilizing various machine learning algorithms. In essence, experimental outcomes indicate that the classification of TextGAN through the separate amalgamation of CBoW and Skip-gram, along with CNN deep learning approach, attains the highest accuracy for the given text classification task. Furthermore, the fusion of TextGAN with SVM, RF, and NB classifiers also exhibits a robust performance. However, it is worth noting that the choice of model and feature extraction techniques may vary depending on the specific characteristics of the text data and the classification task at hand.

Figures 3, 5, and 7 depict the average Precision-Recall curves of the BML classifiers with the lowest LoRAS and highest TextGAN performance achieved through 10-fold stratified cross-validation on the ATC dataset. The average Precision and Recall values of the SVM, RF, and NB algorithms combined with the TextGAN method have reached the highest values compared to other BML models.

High Precision and Recall values indicate the ability of the TextGAN model to produce accurate and comprehensive results. A high Precision value demonstrates that the model has a low probability of making false hate predictions. The high Recall value in the model signifies how accurately genuine instances of hate speech are being detected.

The average Precision and Recall values of the SVM, NB, and RF algorithms combined with the LoRAS method are the lowest among all BML classification models. A low Precision value implies a high ratio of false hate instances (incorrectly predicted) among the classified hate instances. This suggests that a significant portion of instances labeled as hate by the LoRAS model are not actually hate speech. Similarly, the low Recall value of the same sampling model indicates a low ratio of classified hate instances among the actual hate instances. In other words, it suggests that the model's ability to capture genuine instances of hate speech is weak. Figure 9 displays the average Precision-Recall curves of the CNN classifier with the lowest Clustering-based undersampling and the highest TextGAN performance among all CNN models.

Figures 4, 6, 8, and 10 display the ROC curves of the BML and CNN classifiers with the lowest LoRAS and Clustering-Based Undersampling performance, as well as the highest TextGAN performance. The ROC value achieved by the TextGAN model indicates a high Recall and a low false hate detection rate, signifying its strong classification capability. Conversely, models with lower ROC values exhibit weaker hate classification abilities, potentially performing similarly to random hate predictions.

Table 2. Macro-Averaged F1 results of the SVM classifier.

Tablo 2. SVM sınıflandırıcısının makro ortalamalı F1 sonuçları.

Data-Balancing Methods	Classifier	BoW+ TF-IDF	bigram+ TF-IDF	trigram+ TF-IDF
No resampling	SVM	0,902	0,895	0,891
Random oversampling		0,899	0,890	0,880
SMOTE		0,800	0,767	0,752
K-means SMOTE		0,769	0,758	0,751
TextGAN		**0,958**	0,951	0,947
LoRAS		0,525	0,509	0,504
Nearmiss3		0,844	0,840	0,836
Tomek Links		0,854	0,899	0,891
Clustering-based undersampling		0,860	0,829	0,829

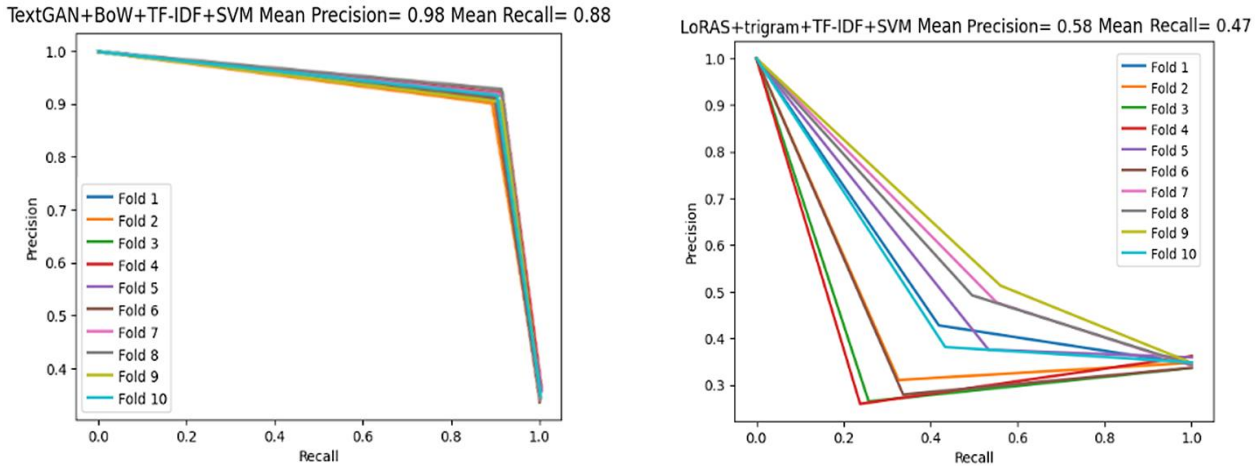

Figure 3. Average Precision-Recall curves for SVM classifier with lowest LORAS and highest TextGAN performance via 10-fold stratified cross-validation on the ATC Dataset.

Şekil 3. ATC veri seti üzerinde 10 katmanlı stratifiye çapraz doğrulama ile elde edilen en düşük LoRAS ve en yüksek TextGAN performansına sahip SVM sınıflandırıcının ortalama Kesinlik-Duyarlılık eğrileri.

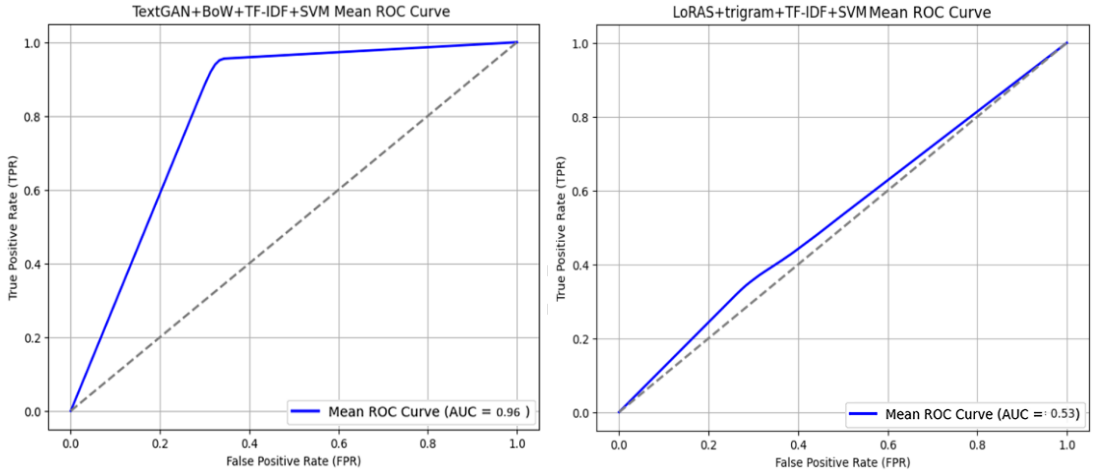

Figure 4. Average ROC curves of SVM classifier with minimum LORAS and maximum TextGAN performance on the ATC dataset.

Şekil 4. ATC veri setinde minimum LORAS ve maksimum TextGAN performansıyla SVM sınıflandırıcısının ortalama ROC eğrileri.

Table 3. Macro-Averaged F1 results of the RF classifier.

Tablo 3. RF sınıflandırıcısının Makro ortalamalı F1 sonuçları.

Data-Balancing Methods	Classifier	BoW+ TF-IDF	bigram+ TF-IDF	trigram+ TF-IDF
No resampling	RF	0,898	0,892	0,888
Random oversampling		0,889	0,888	0,885
SMOTE		0,874	0,865	0,860
K-means SMOTE		0,875	0,869	0,861
TextGAN		0,949	**0,954**	0,939
LoRAS		0,496	0,486	0,480
Nearmiss3		0,809	0,813	0,808
Tomek Links		0,896	0,893	0,888
Clustering-based undersampling		0,775	0,777	0,777

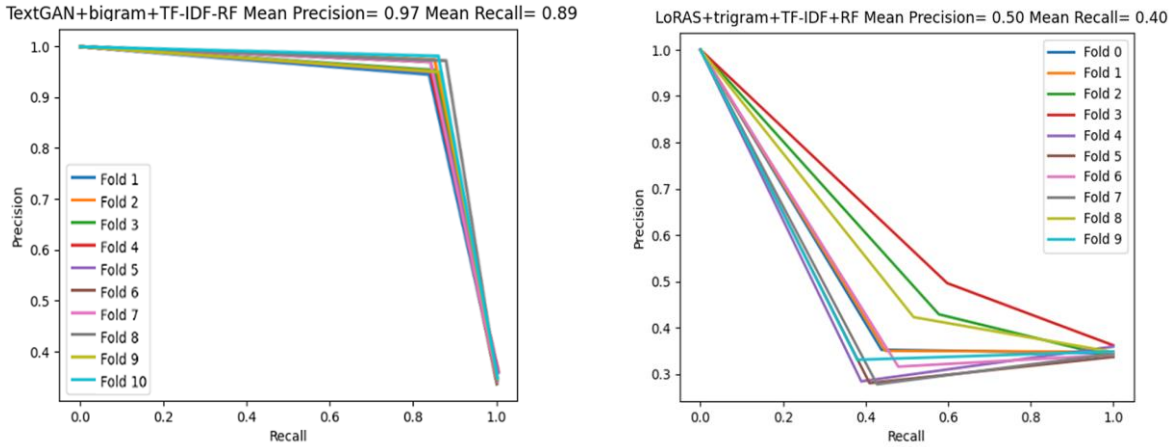

Figure 5. Average Precision-Recall curves for RF classifier with lowest LORAS and highest TextGAN performance via 10-fold stratified cross-validation on the ATC Dataset.

Şekil 5. ATC veri seti üzerinde 10 katmanlı stratifiye çapraz doğrulama ile elde edilen en düşük LoRAS ve en yüksek TextGAN performansına sahip RF sınıflandırıcının ortalama Kesinlik-Duyarlılık eğrileri.

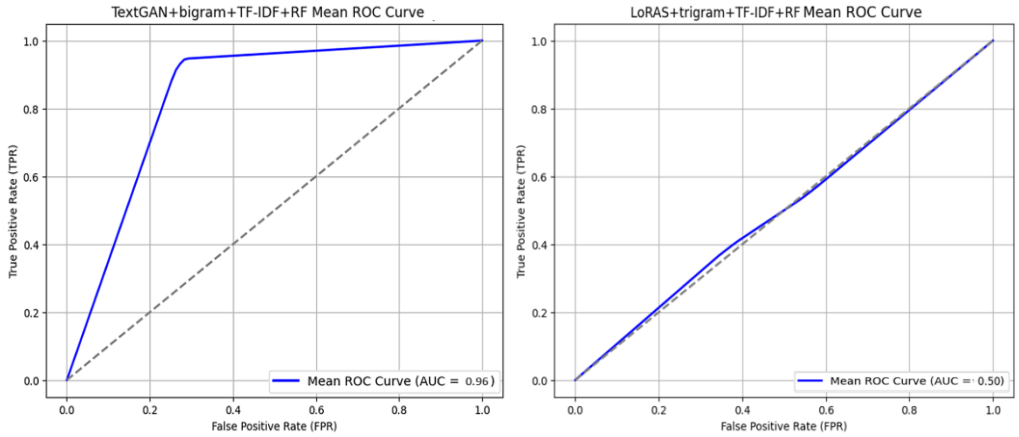

Figure 6. Average ROC curves of RF classifier with minimum LORAS and maximum TextGAN performance on the ATC dataset.

Şekil 6. ATC veri setinde minimum LORAS ve maksimum TextGAN performansıyla RF sınıflandırıcısının ortalama ROC eğrileri.

Table 4. Macro-Averaged F1 results of the NB classifier.

Tablo 4. NB sınıflandırıcısının Makro ortalamalı F1 sonuçları.

Data-Balancing Methods	Classifier	BoW+ TF-IDF	bigram+ TF-IDF	trigram+ TF-IDF
No resampling	NB	0,893	0,896	0,894
Random oversampling		0,867	0,870	0,868
SMOTE		0,890	0,894	0,892
K-means SMOTE		0,891	0,894	0,892
TextGAN		0,952	**0,955**	0,953
LoRAS		0,518	0,507	0,500
Nearmiss3		0,770	0,791	0,800
Tomek Links		0,893	0,883	0,898
Clustering-based undersampling		0,848	0,860	0,854

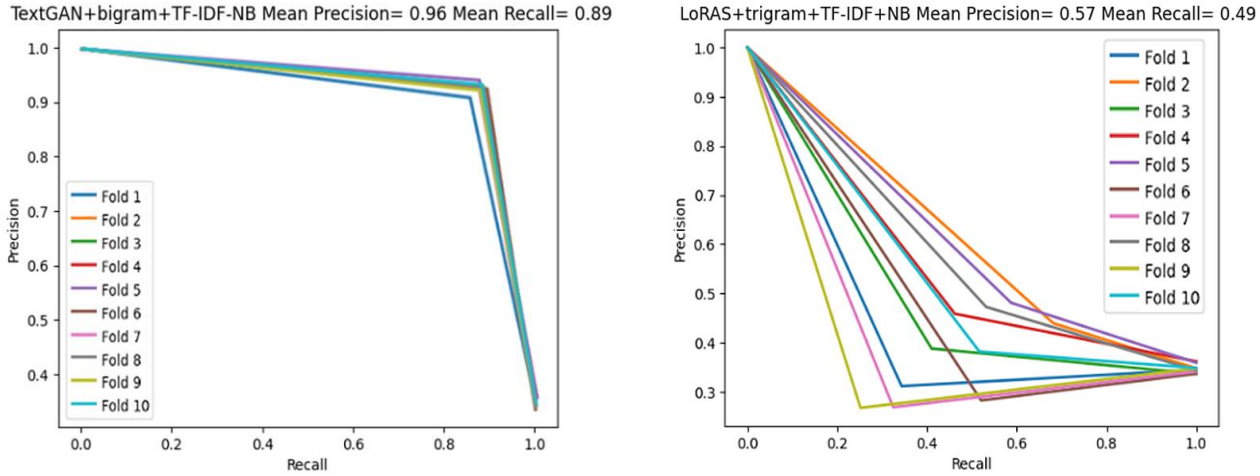

Figure 7. Average Precision-Recall curves for NB classifier with lowest LORAS and highest TextGAN performance via 10-fold stratified cross-validation on the ATC Dataset.

Şekil 7. ATC veri seti üzerinde 10 katmanlı stratifiye çapraz doğrulama ile elde edilen en düşük LoRAS ve en yüksek TextGAN performansına sahip NB sınıflandırıcının ortalama Kesinlik-Duyarlılık eğrileri.

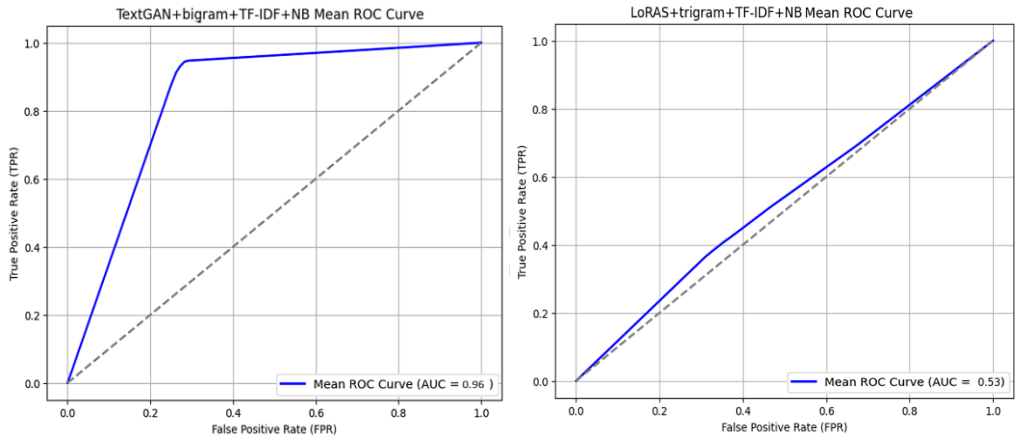

Figure 8. Average ROC curves of NB classifier with minimum LORAS and maximum TextGAN performance on the ATC dataset.

Şekil 8. ATC veri setinde minimum LORAS ve maksimum TextGAN performansıyla NB sınıflandırıcısının ortalama ROC eğrileri.

Table 5. Macro-Averaged F1 results of the CNN classifier.

Tablo 5. CNN sınıflandırıcısının Makro ortalamalı F1 sonuçları.

Data-Balancing Methods	Classifier	CBoW	Skip-gram
No resampling		0,912	0,912
Random oversampling		0,906	0,910
SMOTE		0,710	0,646
K-means SMOTE	CNN	0,875	0,881
TextGAN		**0,972**	**0,972**
LoRAS		0,747	0,683
Nearmiss3		0,806	0,762
Tomek Links		0,871	0,866
Clustering-based undersampling		0,671	0,666

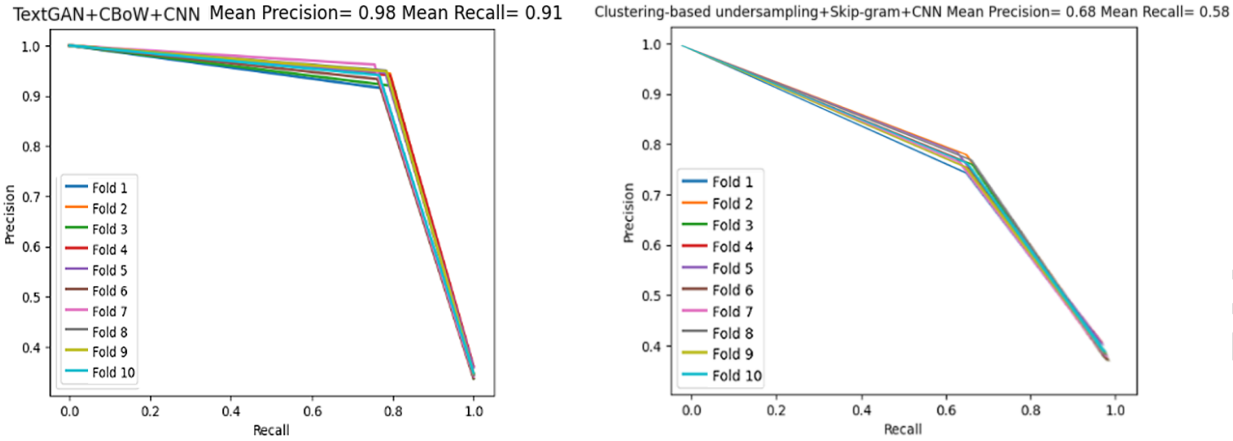

Figure 9. Average Precision-Recall curves for CNN classifier with lowest Clustering-based undersampling and highest TextGAN performance via 10-fold stratified cross-validation on the ATC Dataset.

Şekil 9. ATC veri seti üzerinde 10 katmanlı stratifiye çapraz doğrulama ile elde edilen en düşük Küme tabanlı alt örnekleme ve en yüksek TextGAN performansına sahip NB sınıflandırıcının ortalama Kesinlik-Duyarlılık eğrileri.

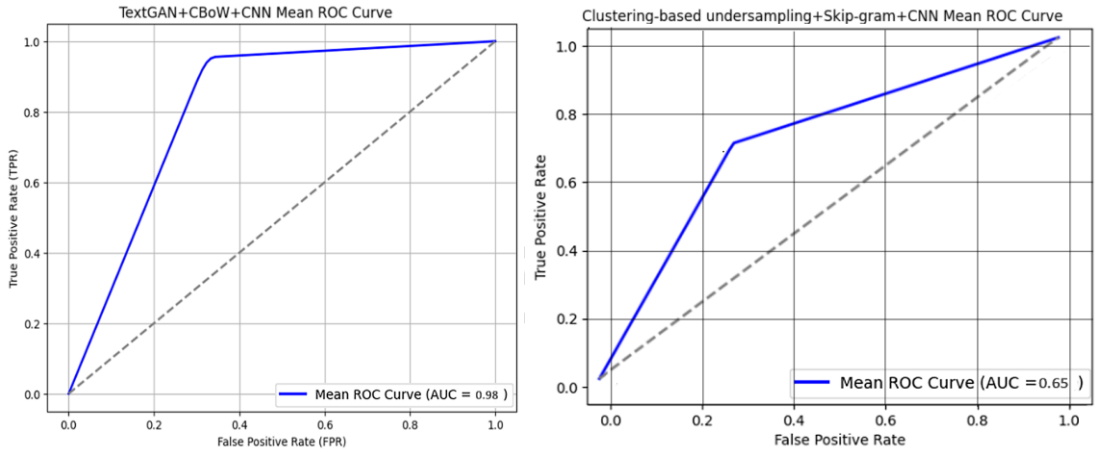

Figure 10. Average ROC curves of CNN classifier with minimum Clustering-based undersampling and maximum TextGAN performance on the ATC dataset.

Şekil 10. ATC veri setinde minimum Clustering-based undersampling ve maksimum TextGAN performansıyla NB sınıflandırıcısının ortalama ROC eğrileri.

Table 6. Best model combination classification results.

Tablo 6. En iyi model kombinasyon sınıflandırma sonuçları.

Model	Best Score
TextGAN+BoW+TF-IDF+SVM	0.958
TextGAN +bigram+TF-IDF+RF	0.954
TextGAN+bigram+TF-IDF+NB	0.955
TextGAN+CBoW+CNN	**0.972**
TextGAN+Skip-gram+CNN	**0.972**

5 Discussion and conclusions

This study conducted an empirical analysis on a dataset consisting of Turkish Instagram comments by employing data balancing approaches for hate speech detection. Different Machine Learning-Based (ML) and Deep Learning-Based (DL) models (such as SVM, RF, NB, and CNN) were combined with imbalanced data processing methods (including Random Oversampling, SMOTE, k-means SMOTE, TextGAN, LoRAS, Nearmiss3, Tomek Links, and Cluster-Based Undersampling). The macro-averaged F1 score was used for performance evaluation. The best performance (i.e., 0.972) was achieved through the TextGAN method for data-level resampling with the CBoW+CNN and Skigram-CNN models. The lowest performance was obtained from combinations utilizing the LoRAS sampling method.

In this study, the Macro-Averaged F1 score was employed, which computes the average F1 scores for each individual class, and thus is not influenced by class imbalance. In various text

sampling studies, F1 score has commonly been used for classification tasks [13,36]. Although this study is the first TextGAN application on a Turkish-language dataset, when compared with a similar study conducted in English [13], it can be observed that the TextGAN combined with the CNN deep learning approach yields better classification results. Furthermore, when compared with another study utilizing a CNN+GloVe model with a text generation GAN, better classification performance is achieved in our study [36]. The rationale behind the superior outcomes in both studies could be attributed to our dataset having high distinctiveness for detecting offensive words in Turkish due to the specificity of offensive language in Turkish, having binary labels for each class, the efficacy of the preprocessing procedure, the successful choice of hyperparameters, and the enhanced classification success through the TextGAN method.

This study demonstrates the success of text generation GAN models on Turkish datasets. Further research evaluating the usage of different text generation GAN models on Turkish datasets with more labels could prove beneficial. Additionally, sentiment analysis studies can be conducted using text generation GAN models in conjunction with Large Language Models (LLMs) for Turkish language research. More research can be carried out to investigate the effectiveness of the TextGAN data balancing method in combination with other deep learning algorithms, as well as with various transfer models such as BERT, distilBERT, AlBERT, and RoBERTa. Such research endeavors may contribute to the development of more accurate and efficient hate speech detection systems for social media platforms.

6 Author contribution statements

The authors confirm their contribution to the paper as follows: Author 1 designed and performed the experiments, derived the models, and analyzed the data. Author 2 supervised the study and contributed to the design and implementation of the research. Author 3 aided in interpreting the results and worked on the manuscript. All authors discussed the results and commented on the manuscript.

7 Ethics committee approval and conflict of interest statement

Ethics committee permission is not required for the article prepared. There is no conflict of interest with any person/institution in the article prepared.

8 References

1. Hudson, D. L. J. "Is Cyberbullying Free Speech". *ABA J.* 102, 2016.
2. Kottasová, I. "Europe says Twitter is failing to remove hate speech". https://money.cnn.com/2017/06/01/technology/twitter-facebook-hate-speech-europe/index.html. (19.12.2019).
3. Park JH, Fung P. "One-step and Two-step Classification for Abusive Language Detection on Twitter". *arXiv Prepr. arXiv1706.01206*, 41–45, 2017.
4. Chen H, McKeever S, Delany S J. "Harnessing the power of text mining for the detection of abusive content in social media". *Advances in Intelligent Systems and Computing,* 513, 187–205, 2017.
5. Wiegand M, Siegel M, Ruppenhofer J. "Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language". *GermEval 2018 Shared Task on the Identification of Offensive Language*, 2018.
6. Davidson T, Warmsley D, Macy M, Weber I. "Automated Hate Speech Detection and the Problem of Offensive Language". *Proceedings of the 11th Conference on Web and Social Media*, 2017.
7. Karayigit H, Acı Cİ, Akdaglı A. "Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods". *Expert Syst. Appl.*, 174, 114802, 2021.
8. Ozel SA, Akdemir S, Sarac E, Aksu H. "Detection of cyberbullying on social media messages in Turkish". *2017 International Conference on Computer Science and Engineering (UBMK),* 366–370, 2017. doi:10.1109/UBMK.2017.8093411
9. Wearesocial. Creative Agency – "We Are Social UK. (2021)". https://wearesocial.com/uk/. (8.11.2021)
10. Waseem Z. "Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter". *Proceedings of the First Workshop on NLP and CSS,* 138–142. 2016.
11. Zhang Z, Luo L. "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter". *Semant. Web,* 10, 925–945, 2018.
12. Badjatiya P, Gupta, S Gupta, M, Varma V. "Deep learning for hate speech detection in tweets". *Proceedings of the 26th international conference on World Wide Web companion*, 759-760, 2017. doi:10.1145/3041021.3054223
13. Tolba M, Ouadfel S, Meshoul S. "Hybrid ensemble approaches to online harassment detection in highly imbalanced data". *Expert Syst. Appl.*, 175, 114751, 2021.
14. Aydilek İB. "Yazılım hata tahmininde kullanılan metriklerin karar ağaçlarındaki bilgi kazançlarının incelenmesi ve iyileştirilmesi". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 24, 5, 906-914, 2018.
15. Elkan C. "The foundations of cost-sensitive learning". *International joint conference on artificial*, 2001.
16. Davidson, T., Warmsley, D., Macy, M. & Weber, I. "Automated Hate Speech Detection and the Problem of Offensive Language". T*he international AAAI conference on web and social media* 512–515, 2017.
17. Waseem Z, Hovy D. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter". *The NAACL Student Research Workshop,* 88–93, 2016. doi:10.18653/v1/N16-2013
18. ElSherief M, Nilizadeh S, Nguyen D, Vigna G, Belding E. "Peer to Peer Hate: Hate Speech Instigators and Their Targets". *12th Int. AAAI Conf. Web Soc. Media, ICWSM 2018,* 52–61, 2018.
19. Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M. "Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis". *arXiv Prepr. arXiv1701.08118*, 2017. doi:10.17185/duepublico/42132
20. Vigna F, Del Cimino A, Dell'orletta F, Petrocchi M, Tesconi M. "Hate me, hate me not: Hate speech detection on Facebook". *ITA-SEC 17*, 2017.
21. Kwok I, Wang Y. "Locate the Hate: Detecting Tweets against Blacks". in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
22. Warner W, Hirschberg J. "Detecting Hate Speech on the World Wide Web". *Proc. Second Work. Lang. Soc. media* 19–26, 2012.

23. Burnap P, Williams ML. “Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making”. *Policy and Internet*, 7, 223–242, 2015.
24. Djuric N, Zhou J, Morris R, Grbovic M, Radosavljevic V, Bhamidipati N. “Hate speech detection with comment embeddings”. *WWW 2015 Companion - Proc. 24th Int. Conf. World Wide Web,* 29–30, 2015. doi:10.1145/2740908.2742760
25. Founta AM, Chatzakou D, Kourtellis N, Blackburn J, Vakali A, Leontiadis I. “A Unified Deep Learning Architecture for Abuse Detection”. *Proc. 10th ACM Conf. Web Sci.* 105–114, 2018.
26. Song G, Huang D, Zhang Y. “A Hybrid Model for Monolingual and Multilingual Toxic Comment Detection”. *Teh. Vjesn.,* 28, 1667–1673, 2021.
27. He H, Bai Y, Garcia EA, Li S. “ADASYN: Adaptive synthetic sampling approach for imbalanced learning”. *Proc. Int. Jt. Conf. Neural Networks,* 1322–1328, 2008. doi:10.1109/IJCNN.2008.4633969
28. Al-Garadi MA, Varathan KD, Ravana SD, “Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network”. *Comput. Human Behav.,* 63, 433–443, 2016.
29. Last F, Douzas G, Bacao F. “Oversampling for Imbalanced Learning Based on K-Means and SMOTE”. *Inf. Sci.,* 465, 1–20, 2017.
30. Zhang Y, Gan Z, Fan K, Chen Z, Henao R, Shen D, Carin L. “Adversarial Feature Matching for Text Generation”. *International Conference on Machine Learning,* 4006–4015, 2017.
31. Ćosović M, Obradović S. “BGP Anomaly Detection with Balanced Datasets”. *Teh. Vjesn.,* 25, 766–775, 2018.
32. Anand A, Gorde K, Moniz JRA, Park N, Chakraborty T, Chu BT. “Phishing URL detection with oversampling based on text generative adversarial networks”. *IEEE International Conference on Big Data (Big Data),* 1168–1177, 2018.
33. Bej S, Davtyan N, Wolfien M, Nassar M, Wolkenhauer O. “LoRAS: an oversampling approach for imbalanced datasets”. *Mach. Learn., 2020 1102* 110, 279–301, 2020.
34. Srinivasan R, Subalalitha CN. “Sentimental analysis from imbalanced code-mixed data using machine learning approaches”. *Distrib. Parallel Databases,* 1–16, 2021. doi:10.1007/S10619-021-07331-4
35. Rupapara V, Rustam F, Shahzad HF, Mehmood A, Ashraf I, Choi GS. “Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model”. *IEEE Access,* 9, 78621–7863, 2021.
36. Imran A, Yang R, Kastrati Z, Daudpota S. “The impact of synthetic text generation for sentiment analysis using GAN based models”. *Egypt. Informatics,* 23, 547–557, 2022.
37. Rao S, Verma A, Bhatia T. “Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data”. *Expert Syst. Appl.,* 217, 119594, 2023.
38. Madani M, Motameni H, Mohamadi H. “KNNGAN: an oversampling technique for textual imbalanced datasets”. *J. Supercomput.,* 1–36, 2022. doi:10.1007/S11227-022-04851-3
39. Bozkurt F, Çoban Ö, Baturalp GF, Yücel AŞ. “High Performance Twitter Sentiment Analysis Using CUDA Based Distance Kernel on GPUs”. *Teh. Vjesn.*, 26, 1218–1227, 2019.
40. Aktunç H. *Big slang dictionary of Turkish: (with witnesses)*. YapıKredi Yayınları, 2000.
41. Developers, Scikit. “sklearn.model_selection.StratifiedKFold — scikit-learn 1.2.1 documentation”. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html (7.03.2023)
42. Tekin MC, TUNALI V. "Yazılım geliştirme taleplerinin metin madenciliği yöntemleriyle önceliklendirilmesi". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 25, 5, 615-620, 2019.
43. Tabinda Kokab S, Asghar S, Naz S. “Transformer-based deep learning models for the sentiment analysis of social media data”. *Array* 14, 100157, 2022.
44. Abdi A, Shamsuddin SM, Hasan S, Piran J. “Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion”. *Inf. Process. Manag.,* 56, 1245–1259, 2019.
45. Cevik F, Kilimci ZH. "Derin öğrenme yöntemleri ve kelime yerleştirme modelleri kullanılarak Parkinson hastalığının duygu analiziyle değerlendirilmesi". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 27, 2, 151-161, 2020.
46. Fatima M, Pasha M. “Survey of Machine Learning Algorithms for Disease Diagnostic”. *J. Intell. Learn. Syst. Appl.,* 09, 1–16, 2017.
47. Abooraig R, Al-Zu'bi S, Kanan T, Hawashin B, Al Ayoub M, Hmeidi I. “Automatic categorization of Arabic articles based on their political orientation”. *Digit. Investig.*, 25, 24–41, 2018.
48. Saric M, Dujmic H, Russo M. “Scene Text Extraction in IHLS Color Space Using Support Vector Machine”. *Inf. Technol. Control,* 44, 20–29, 2015.
49. Breiman L. “Random Forests”. *Mach. Learn.,* 45, 5–32, 2001.
50. Tao W, Chang D. “News Text Classification Based on an Improved Convolutional Neural Network”. *Teh. Vjesn.*, 26, 1400–1409, 2019.
51. Dogan T, Uysal AK. “Improved inverse gravity moment term weighting for text classification”. *Expert Syst. Appl.*, 130, 45–59, 2019.
52. Saraç E, Özel SA. “Effects of Feature Extraction and Classification Methods on Cyberbully Detection”. *Süleyman Demirel Üniversitesi Fen Bilim. Enstitüsü Derg.,* 21, 190, 2016.
53. Developers, C. “Colaboratory”. https://colab.research.google.com/. (5.12.2021)
54. Kedia A, Rasu M. Hands-on Python natural language processing: explore tools and techniques to analyze and process text with a view to building real-world NLP applications. Packt Publishing Ltd, 202