Generative LLM oversampling for the class imbalance problem in hate speech detection

Nicolas Antonio Cloutier[1] and Nathalie Japkowicz[2]

[1] Jackson-Reed High School, Washington, D.C., USA
nicocloutier1@gmail.com
[2] American University, Washington, D.C., USA
japkowicz@american.edu

Abstract. Online hate speech has become increasingly prevalent with the rise of social media. As such, methods for automatically detecting and classifying hate speech have become the subject of much research. A common challenge in this and other domains is the class imbalance problem, where one class in a dataset is far more common than another. We propose the use of a generative large language model (LLM), specifically OpenAI's GPT-2, as a method of oversampling text data in order to account for this imbalance, comparing it to other resampling methods on three tasks: binary classification of tweets as antisemitic or not, multi-class classification of antisemitic tweets into subtypes, and a combination of the two. We find that generative LLM resampling does not produce better results for binary classification than other resampling methods, but does improve performance on the other two tasks.

1 Introduction

In recent years, hate speech has become increasingly mainstream and common within social media sites [25]. This rise has not only caused online spaces to become less hospitable, but has also had several offline effects. On top of having severe psychological effects on the recipient [25], online hate speech also played a role in disseminating extremist anti-Rohingya voices leading to violence in Myanmar [12], and has provided motivation for several perpetrators of offline violent hate crimes [25]. These events have motivated responses from numerous parties, including the social media sites themselves, that are increasingly looking to stop the spread of these messages [27], and governmental bodies, that are seeking to regulate or prevent the spread of hate speech [2]. At the same time as hate speech has been becoming more common, social media sites have been generating more and more content, with popular social media site Twitter generating an average of 500 million tweets per day in 2019 [20].

With these developments and the large amount of content on social media sites, these sites have been increasingly looking to automatic detection methods for hate speech [27]. These methods use Machine Learning (ML) to automatically detect and classify hateful speech, removing some of the work done by moderators, whose primary job is to respond reactively to user reports of hate speech [27]. Chandra et al. [4] used a combination of image and text processing and classification algorithms to classify images and text on social media sites Twitter and Gab as antisemitic or not [4]. We use the text component of their dataset to further analyze the presence of hate speech on Twitter and investigate new algorithms for classification.

One difficulty with this dataset is that it is imbalanced, meaning one classification is far more common in the dataset than another. Imbalanced data can negatively impact the performance of ML classification algorithms [26], affecting numerous domains that use ML classification. Many ML algorithms are inadequately prepared to handle the class imbalance problem [26], leading many to look to other solutions, including resampling methods, that in some way change the training data by adding or removing samples in order to allay the effects of the class imbalance problem [14].

With this in mind, there are two crucial research questions this paper seeks to answer. First: how can automatic ML methods be improved in the domain of text classification for hate speech detection? Second: how can the class imbalance problem be dealt with for text data? These are the questions we seek to provide answers to, with the hope that they may inform future research and hate speech detection systems.

2 Previous work

Antisemitism detection, as well as hate speech detection generally, has been the subject of much research. Martins et al. [19] used ML models to analyze and classify antisemitism in posts on social media sites Gab and 4chan, and González-Pizarro and Zanettou [11] used large language models (LLMs) to do the same. Chandra et al. [4] introduced a new dataset of labeled antisemitic posts, including text and images. These posts were labeled not only for whether or not they are antisemitic, but also their type of antisemitism, with the researchers grouping antisemitic posts into political, economic, religious, and racial antisemitism, and trained models to classify both antisemitic status and type of antisemitism. This Twitter dataset is imbalanced, with the non-antisemitic class far outnumbering the antisemitic class, and with political and racial antisemitism being more common than religious and economic antisemitism.

Imbalance is a common issue in ML classification problems. The class imbalance problem can severely negatively affect model predictive power, with the less common class (termed the "minority class") often being misclassified due to its low prevalence in the dataset relative to the larger class (the "majority class") [1]. This problem has affected fields as distinct as medicine, fraudulent call detection, and risk management [26]. Due to the severity with which this problem can limit model performance and its widespread nature, being seen in numerous distinct fields, it has become a focus of researchers as an area of improvement [1], with numerous techniques being created to allay the effects of the problem.

One such group of techniques is resampling, altering the training data by adding new samples or taking existing samples away in order to improve model performance. Generally, there are two types of resampling: oversampling, which describes the process of adding synthetic samples to a dataset, and undersampling, which describes the process of removing existing samples from a dataset [24]. Resampling techniques can improve the performance of ML models when trained on imbalanced data [18] [16].

One method for oversampling is the use of generative models, including generative adversarial networks (GANs) and autoencoders to generate synthetic data. These methods seek to match the distribution of the original dataset and create synthetic samples in accordance with that distribution [13], with the goal of creating authentic synthetic examples for model training. They have achieved success in their applications, but primarily in computer vision and tabular data [13] [8] [3] [5]. Applications of these generative models to natural language processing (NLP) do occur [21], but they are generally less common than applications to other areas. NLP tends to use more traditional statistical methods for oversampling such as SMOTE and random oversampling [29] [10].

While these advances in resampling have been occurring, similar advances have been made in generative LLMs, such as the GPT series of models from OpenAI, which have the ability to generate human-like text [9] and have been applied to fields such as patent claim generation [17] and healthcare education [22]. These LLMs have been investigated as a method for oversampling text data to account for the class imbalance problem, with relatively consistent success over not using resampling [7] [28] [23]. That being said, these studies do not compare the method with other, simpler or more traditional resampling methods. Drummond & Holt [6] argue that complex methods of resampling are difficult to justify without marked improvements in performance over simpler methods. This provides motivation, now that the method has shown success in accounting for the class imbalance, for investigating how it compares to existing methods of resampling and under which circumstances it improves relative to those methods.

3 Methodology

The purpose of our paper is to study the effects of various types of resampling on hate speech detection and classification tasks. Our models were trained on three different tasks. The first was a simple binary classification task, where the models would attempt to classify the text of a tweet as either antisemitic or not antisemitic. The second was the 4-class type classification, where the dataset would be limited to only antisemitic samples, that would then be grouped into four classes: political, economic, religious, and racial antisemitism. Finally, the models were trained on a 5-class type classification task, where the dataset included both antisemitic and non-antisemitic samples, and the model would attempt to classify the samples into one of the four groups of antisemitism or classify them as not antisemitic, creating five classes total. Every model was trained and tested separately for all three tasks.

In order to best represent a variety of ML architectures, different algorithms and methods of text representation were used. For algorithms, we trained classifiers utilizing the Naïve Bayes, Extreme Gradient Boosting, Decision Tree, and Support Vector Machine algorithms. We also used three methods of representing text: term frequency-inverse document frequency (TF-IDF), raw frequency, and Bag of Words (BoW).[3] Each model was trained with each method of text representation, creating a total of twelve models that were trained for each task. In order to reduce the dimensionality of the dataset, representations for training and testing data were limited to words that had a frequency score of at least 0.5%, meaning that the word had to have a frequency of at least 0.5% of the original, full length of the text in order to be considered by the models.

The dataset is imbalanced. Figure 1 shows the distribution of different classes in the dataset. The first and third charts are for the entire dataset, and the second, containing classifications on types of antisemitism, only contains samples that are antisemitic.

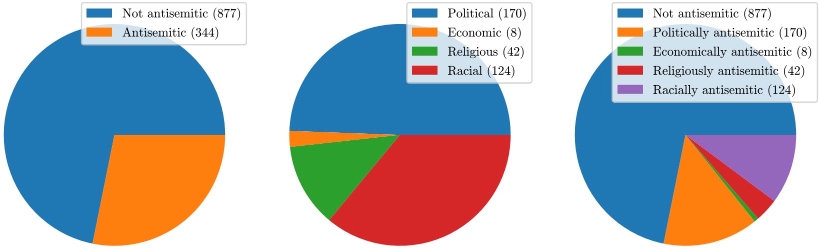

Figure 1. Classification distributions

Several methods of resampling were used to reduce the impact of the class imbalance. One set of models was trained with no resampling, with additional sets being trained using random undersampling, random oversampling, SMOTE with Tomek Links, AdaSyn, and a final set being trained on the augmented dataset generated using the LLM. GPT-2 was used to generate the samples because it is easily available and callable programmatically with the HuggingFace API, unlike more recently released GPT models. The augmented dataset was created by generating 20 samples from each sample in the original dataset, creating a dataset that was larger, but had the same percentages of class imbalance. These models then used random undersampling on this larger dataset to account for this.

In regards to testing, each model was tested using 10-fold cross-validation on the original dataset. It was ensured that no samples generated from oversampling methods were used during testing, and additionally that, when testing the augmented model, if a sample that was used to generate more samples appeared in the testing split, the samples generated with it would not appear in the training split. This ensures that the models were not unfairly advantaged, and that each was tested on a large amount of genuine, unseen data.

For model evaluation, the two main metrics used were the means of the recall and precision scores across each of the classes the model had to classify. This was used in lieu of accuracy in order to account for the class imbalance in the

[3] We initially included a BERT-based representation method, but the models trained with this method produced results no better than chance.

data. Once the models were evaluated, their answers to the testing samples were converted to a binary matrix with each column representing a model and each row representing a sample. We then used the Cochran's Q-test, an extension of McNemar's test appropriate for the case of comparing multiple algorithms on a single domain, to test for significant difference in the models, then the Dunn test for post-hoc analysis.[4] When testing the resampling methods against each other, the data were turned into another matrix with each column representing a resampling method and each row representing a model trained with that method, with the value in the cell being the mean recall of that model. A Friedman's χ^2 test was then performed with a post-hoc Nemenyi test to analyze the results.

4 Results

Overall, the models trained on the LLM-augmented data showed no improvement on the binary task, but they did show improvement on the 5-class classification and especially the 4-class classification tasks. Tables 1 & 2 contain information for the binary classification task with each resampling method, as well as the best-performing model performance and mean model performance for each resampling method, with "performance" meaning the mean recall or precision score the model achieved. The best model was the best performing model of all of the models trained with that particular resampling method, not taking into account the performance on other resampling methods. The data used to create Table 1 received a Friedman χ^2 test statistic of 13.09 and p-value of 2.2×10^{-2}, while Table 2 received a statistic of 19.43 and a p-value of 1.6×10^{-3}. After each table, an array is provided with the results of the post-hoc Nemenyi test on the resampling methods, with p-values displayed in the array. A higher p-value and lighter color means a greater degree of similarity between two models' answers and performance, while a lower p-value and darker color means a lower degree of similarity. In these arrays, "RU" stands for random undersampling, "ST" stands for SMOTE with Tomek links, "RO" stands for random oversampling, "AD" stands for AdaSyn, "NO" stands for no resampling method, and "LA" stands for LLM-augmented.

Table 1. Binary task resampling method recall

Resampling Method	Best score (% over none)	Mean score (% over none)
None	0.667 (0%)	0.613 (0%)
Random Undersampling	0.689 (3.3%)	**0.638 (4.1%)**
SMOTE-Tomek	0.668 (0.1%)	0.614 (0.2%)
Random Oversampling	**0.690 (3.4%)**	0.623 (1.6%)
AdaSyn	0.661 (-0.9%)	0.608 (0.8%)
LLM-Augmented	0.679 (1.8%)	0.617 (0.7%)

[4] See [15]

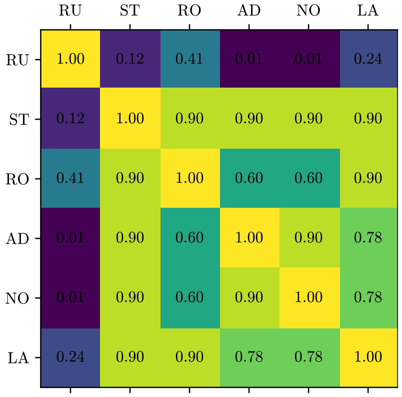

Figure 2. Binary recall Nemenyi test results

Table 2. Binary task resampling method precision

Resampling Method	Best score (% over none)	Mean score (% over none)
None	**0.734 (0%)**	**0.644 (0%)**
Random Undersampling	0.666 (-9.3%)	0.621 (-3.6%)
SMOTE-Tomek	0.670 (-8.7%)	0.612 (-5.0%)
Random Oversampling	0.674 (-8.2%)	0.617 (-4.2%)
AdaSyn	0.646 (-12.0%)	0.600 (-6.8%)
LLM-Augmented	0.668 (-9.0%)	0.613 (-4.8%)

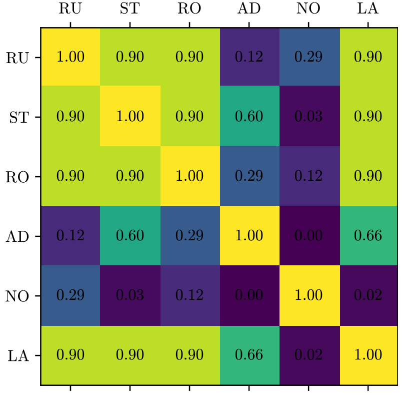

Figure 3. Binary precision Nemenyi test results

In the binary task, the LLM-augmented models show no performance improvement over the other models. Generally, the resampling methods tend to do better than the models with no resampling on recall, but have worse precision performance, with this pattern being sustained by the models trained on LLM-augmented data.

The results of the 5-class classification task are shown below. The data used to create Table 3 received a Friedman χ^2 test statistic of 33.24 and p-value of 3.4×10^{-6}, while Table 4 received a statistic of 11.57 and a p-value of 4.1×10^{-2}.

Table 3. 5-class task resampling method recall

Resampling Method	Best score (% above none)	Mean score (% above none)
None	0.312 (0%)	0.265 (0%)
Random Undersampling	0.321 (2.9%)	0.265 (0%)
SMOTE-Tomek	0.289 (-7.4%)	0.271 (2.3%)
Random Oversampling	0.369 (18.3%)	0.298 (12.5%)
AdaSyn	0.316 (1.3%)	0.251 (-5.3%)
LLM-Augmented	**0.402 (28.8%)**	**0.347 (30.9%)**

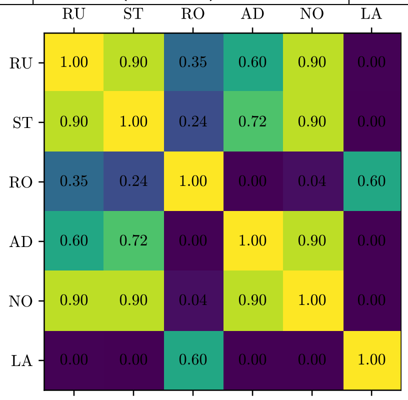

Figure 4. 5-class recall Nemenyi test results

Table 4. 5-class task resampling method precision

Resampling Method	Best score (% above none)	Mean score (% above none)
None	**0.524 (0%)**	**0.325 (0%)**
Random Undersampling	0.274 (-47.7%)	0.249 (-23.4%)
SMOTE-Tomek	0.295 (-43.7%)	0.264 (-18.8%)
Random Oversampling	0.319 (-39.1%)	0.271 (-16.6%)
AdaSyn	0.436 (-16.8%)	0.286 (-12.0%)
LLM-Augmented	0.302 (-42.4%)	0.270 (-16.9%)

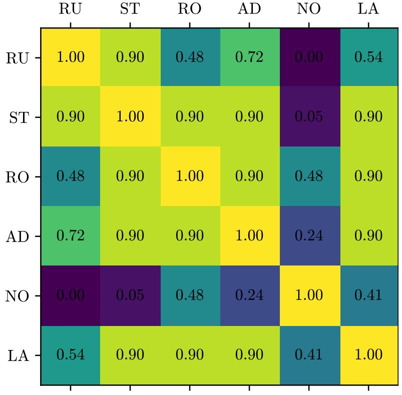

Figure 5. 5-class precision Nemenyi test results

The results on the 5-type task begin to show the benefits of LLM augmentation. The pattern of no resampling outperforming resampling on precision while the inverse happens on recall remains, but now LLM augmentation improves over no resampling by a far wider margin than any other resampling method, while retaining similar levels of drops in precision to other methods. Now, its gain in mean recall is larger than its loss in mean precision, which is not the case for any other resampling method in the 5-class task, and is only true for random undersampling in the binary task.

The results of the 4-class classification task are shown below. The data used to create Table 5 received a Friedman χ^2 test statistic of 21.95 and p-value of 5.3×10^{-4}, while Table 6 received a statistic of 27.7 and a p-value of 4.2×10^{-5}.

Table 5. 4-class task resampling method recall

Resampling Method	Best score (% above none)	Mean score (% above none)
None	0.337 (0%)	0.328 (0%)
Random Undersampling	0.414 (22.8%)	0.339 (3.4%)
SMOTE-Tomek	0.420 (24.6%)	0.366 (11.6%)
Random Oversampling	0.400 (18.7%)	0.358 (9.1%)
AdaSyn	0.471 (39.8%)	0.322 (-1.8%)
LLM-Augmented	**0.509 (51.0%)**	**0.433 (32.0%)**

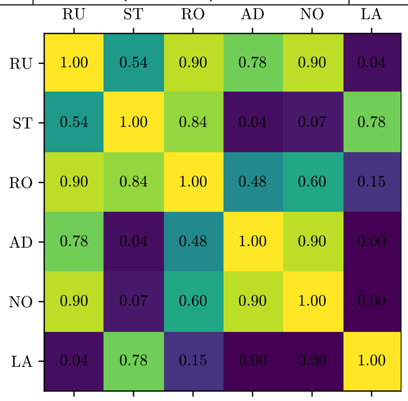

Figure 6. 4-class recall Nemenyi test results

Table 6. 4-class task resampling method precision

Resampling Method	Best score (% above none)	Mean score (% above none)
None	0.452 (0%)	0.372 (0%)
Random Undersampling	0.361 (-17.9%)	0.323 (-13.2%)
SMOTE-Tomek	0.381 (-15.7%)	0.347 (-6.7%)
Random Oversampling	0.374 (-17.3%)	0.351 (-5.6%)
AdaSyn	0.378 (-16.4%)	0.344 (-7.5%)
LLM-Augmented	**0.504 (11.5%)**	**0.409 (9.9%)**

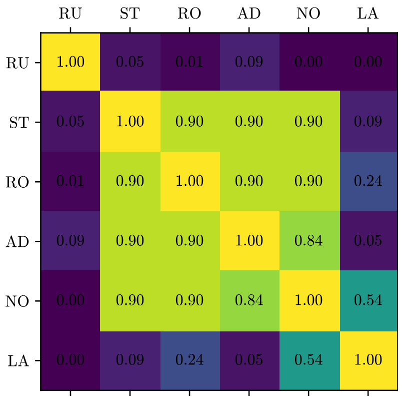

Figure 5. 4-class precision Nemenyi test results

Now, the pattern of gains on recall and losses on precision continue for all methods except LLM augmentation. LLM-augmented models not only increase the recall by a greater margin than any other method, but also increase the precision of the model on the 4-class task. This makes it the only resampling method in any of the three tasks to receive a higher precision score either on the best model or on mean score than no resampling, on top of widening the margin by which it beats no resampling compared to the 5-class task.

One initial possible explanation for the high performance of the models with LLM-augmented data relative to other models on the 5-class and especially 4-class tasks is that the LLM augmentation provides performance improvements with only one particular class that other resampling methods did poorly on, artificially raising the mean recall even though there was only one class affected significantly. Looking further at the data, however, while some classes were more affected than others, overall, the models trained on LLM-augmented data show improvements in multiple classes. On the 4-type classification task, Tables 7 & 8 show the mean recall and precision of each method on each particular class, respectively.

Table 7. 4-class task class-wise resampling method recall

Resampling Method	Political	Economic	Religious	Racial
None	**0.618**	0.106	0.203	0.283
Random Undersampling	0.379	**0.240**	0.434	0.200
SMOTE-Tomek	0.506	0.135	0.465	0.245
Random Oversampling	0.487	0.163	0.442	0.230
AdaSyn	0.574	0.135	0.223	0.258
LLM-Augmented	0.492	0.231	**0.471**	**0.405**

Table 8. 4-class task class-wise resampling method precision

Resampling Method	Political	Economic	Religious	Racial
None	0.598	0.029	0.254	0.390
Random Undersampling	0.601	0.024	0.205	0.362
SMOTE-Tomek	0.653	0.018	0.234	0.378
Random Oversampling	**0.661**	0.015	0.227	0.392
AdaSyn	0.602	0.017	0.243	0.390
LLM-Augmented	0.577	**0.111**	**0.367**	**0.431**

In precision and recall, the model is outperformed in political antisemitism (the largest class), and in recall it is slightly outperformed by random undersampling in economic antisemitism, but it outperforms the other models in religious and racial antisemitism on recall and in all classes other than political in precision.

5 Discussion

The most striking result of these trials is the difference in relative performance among the different tasks. Given that a class imbalance exists in all tasks, it may be expected that the generative LLM oversampling would produce similar results for each of these imbalances, but this is not the case. Furthermore, the difference in performance of the generative LLM-trained models to the models trained on data from other resampling methods grows from the 5-class to the 4-class task, suggesting that there is something particular to the binary classification problem that the generative LLM is not as good at accounting for as other resampling methods.

One possible reason for this is the relative performance of undersampling and oversampling methods in general. Generally, oversampling methods tend to, at best, perform as well as undersampling methods [6], explaining why the only pure undersampling method scored best on the binary classification task. That being said, in tasks where undersampling would leave a dataset very small, this pattern may reverse. In the case of the 4-class classification task, one of the classes has only 8 observations, meaning, during undersampling, all other classes would also be reduced to 8 samples, making a very small dataset, that is perhaps inadequate for classification, providing one possible explanation for why oversampling outperformed undersampling on the 4-class and 5-class problems. As for why the generative LLM specifically outperformed other oversampling methods, it may simply be that generating text and then converting it into a vectorized form produces more authentic results than generating new samples from those vectorized forms with methods such as AdaSyn and SMOTE. This would explain the augmented LLM's outperformance of other oversampling methods on the 4-class and 5-class tasks, but would leave unexplained why it was outperformed by random oversampling in the binary classification task.

6 Conclusions and Future Work

In order to reach a better understanding of the applications of generative LLM oversampling, more research should be done on the abilities of the method to account for class imbalances in text data. Hopefully, future research that applies the method to datasets with differing sizes, levels of severity of imbalance, and number of classes to classify will help pinpoint the exact situations where generative LLM oversampling can be most helpful, informing future systems that may use the method.

This paper proposes the use of generative LLMs to oversample imbalanced text datasets in order to account for the imbalance. Training models with different resampling methods, we find that results are unequal among tasks. The models trained on generative LLM data showed no improvement over other models in the binary classification of social media posts into an antisemitic/not antisemitic dichotomy, but they showed great improvement on the classification of tweets into types of antisemitism. Further research on imbalanced datasets with different characteristics is required to understand why the difference in relative performance was so stark and what factors most contribute to the utility of this method to accounting for the imbalance problem.

References

1. Abd Elrahman, S. M., Abraham, A: A review of class imbalance problem. Journal of Network and Innovative Computing **1**, 332–340 (2013)
2. Banks, J.: Regulating hate speech online. International Review of Law, Computers and Technology **24**(3), 233–239 (2010)
3. Bellinger, C., Japkowicz, N., Drummond, C.: Synthetic oversampling for advanced radioactive threat detection. IEEE Conference on Machine Learning and Applications 2015
4. Chandra, M., Pailla, D., Bhatia, H., Sanchawala, A., Gupta, M., Shrivastava, M., Kumaraguru, P.: "Subverting the Jewtocracy": Online Antisemitism Detection Using Multimodal Deep Learning. ACM Web Science Conference 2021, 148–157
5. Dai, W., Ng, K., Severson, K., Huan, W., Anderson, F., Stultz, C.: Generative oversampling with a contrastive variational autoencoder. IEEE International Conference on Data Mining 2019, 101–109
6. Drummond, C., Holte, R. C.: C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. Workshop on Learning from Imbalanced Datasets 2003
7. Edwards, A., Ushio, A., Camacho-Collados, J., De Ribaupierre, H., Preece, A.: Guiding Generative Language Models for Data Augmentation in Few-Shot Text Classification. *arXiv preprint arXiv:2111.09064*, (2021)
8. Engelmann, J., Lessmann, S.: Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications **174**, (2021)
9. Floridi, L., Chiriatti, M.: GPT-3: Its nature, scope, limits, and consequences. Minds and Machines **30**, 681–694 (2020)
10. Glazkova, A.: A comparison of synthetic oversampling methods for multi-class text classification. *arXiv preprint arXiv:2008.04636t*, (2020)
11. González-Pizarro, F., Zannettou, S.: Understanding and detecting hateful content using contrastive learning. *arXiv preprint arXiv:2201.08387*, (2022)
12. Green, P., MacManus, T., De la Cour Venning, A.: Countdown to Annihilation: Genocide in Myanmar. International State of Crime Initiative, London (2015)
13. Hao, J., Wang, C., Zhang, H., Yang, G.: Annealing genetic GAN for minority oversampling. *arXiv preprint*, (2020)

14. Japkowicz, N., Shaju, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis **6**(5), 429–449 (2002)
15. Japkowicz, N., Boukouvalas, Z., Shah, M.: Machine Learning Evaluation (in preparation)
16. Khushi, M., Shaukat, K., Alam, T. M., Hammed, I. A., Uddin, S., Luo, S., Yang, X., Consuelo Reyes, M.: A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data. IEEE Access **9**, 109960–109975 (2021)
17. Lee, J. S., Hsiang, J.: Patent Claim Generation by Fine-Tuning OpenAI GPT-2. World Patent Information **62**, (2020)
18. Lee, P. H.: Resampling methods improve the predictive power of modeling in class-imbalanced datasets. International Journal of Environmental Research and Public Health **11**(9), 9776–9789.
19. Martins, R., Gomes, M., Almeida, J. J., Novais, P., Henriques, P.: Hate speech classification in social media using emotional analysis. Brazilian Conference on Intelligent Systems 2018, 61–66
20. Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., Camacho-Collados, M.: Detecting and Monitoring Hate Speech in Twitter. Sensors **19**(21), 4654–4691 (2019)
21. Phung, N. M., Mimura, M.: Evaluation of a cGAN Model and Random Seed Oversampling on Imbalanced JavaScript Datasets. Journal of Information Processing **30**, 591–600 (2022)
22. Sallam, M.: The utility of ChatGPT as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. *medRxiv preprint*, (2023)
23. Shaikh, S., Daudpota, S. M., Imran, A. I., Kastrati, Z.: Towards Improved Classification Accuracy on Highly Imbalanced Text Dataset Using Deep Neural Language Models. Applied Sciences **11**(2), 869 (2021)
24. Shelke, M. S., Deshmukh, P. R., Shandilya, V. K.: A review on imbalanced data handling using undersampling and oversampling technique. International Journal of Recent Trends in Engineering Research **3**(4), 444–449 (2017)
25. Siegel, A.: Online Hate Speech. Social media and democracy: The state of the field, prospects for reform, 56–88 (2020)
26. Sun, Y., Wong, A., Kamel, M.: Classification of Imbalanced Data: A Review. International Journal of Pattern Recognition and Artificial Intelligence **23**(4), 687–719 (2009)
27. Ullmann, S., Tomalin, M.: Quarantining online hate speech: technical and ethical perspectives. Ethics and Information Technology **22**, 69–80 (2020)
28. Usuga-Cadavid, J. P., Grabot, B., Lamouri, S., Fortin, A.: Artificial Data Generation with Language Models for Imbalanced Classification in Maintenance. Artificial data generation with language models for imbalanced classification in maintenance. International workshop on service orientation in holonic and multi-agent manufacturing 2021, 57–68
29. Wijaya, I. D., Putrada, A. G., Oktaria, D.: Overcoming Data Imbalance Problems in Sexual Harassment Classification with SMOTE. International Journal on Information and Communication Technology **8**(1), 20–29 (2022)