Synthetic data generation for machine learning models with cognitive agent simulations

Jim Blythe[1][0009−0000−4614−9976] and Alexey Tregubov[1][0000−0003−3374−0884]

USC Information Sciences Institute, Marina del Rey, CA (USA)
{blythe, tregubov}@isi.edu

Abstract. The use of synthetic data for training machine learning models (ML) in social media domains can address issues such as data availability and bias, but poses challenges, including properly reflecting causal relationships and matching the consistency of real data. In this paper, we explore the benefits and limitations of using synthetic data generated by cognitive agent simulations. By simulating human interactions and social media dynamics, these models can capture constraints and nuances of real-world scenarios. We report initial experiments that show that ML algorithms trained on real data augmented with synthetic data outperform those trained solely on original data, achieving up to 25% improvement in KS distance and RMSE metrics. This approach is applied to two domain problems: predicting code quality based on open-source code discussions and detecting and countering bot attacks on social media platforms. For code quality prediction, we used discussions and patches from the Linux Kernel Mailing List to predict patch reversions. In the bot attack detection problem, synthetic Reddit data helps create realistic social network environments to study interactions between influencers and bots under different conditions. The paper presents empirical evidence supporting the effectiveness of synthetic data in improving ML model performance and introduces an agent-based framework for generating realistic synthetic data for social media experiments. The findings suggest promising avenues for future research and highlight the potential of this approach.

Keywords: Cognitive agent based simulations · Machine learning · Synthetic data · Social media simulation

1 Introduction

There are many challenges to training ML models on social media data, including data availability, observability of user actions and external influences. Synthetic data generation approaches have improved the performance of ML systems in domains such as vision and image classification [13, 14] and question answering [16]. Synthetic data can not only address the potential scarcity of data required to train and validate ML models, but can also mitigate the effects of bias or privacy concerns.

Here we report on initial experiments to improve the performance of ML models applied to social media using synthetic data generated by cognitive agent

simulations. Our models capture constraints that stem both from the dynamics of human interaction and also from the social media platforms involved. When there is insufficient data to support a specific learning task, our models can be trained on larger amounts of data available in a broader context, and adapted to the specific needs.

Using synthetic data from simulations generated with the DASH cognitive agent platform[12], we found that ML algorithms augmented with synthetic data outperformed similar algorithms with only original data by up to 25% both in KS distance and RMSE. We also found that we could achieve similar results to those achieved with a real dataset by using 60% of that data augmented with synthetic data.

We illustrate our approach in two different domain problems: prediction of code quality based on an open-source code discussion, and bot attack prediction and countermeasure ("troll" attacks). Prediction of code quality based on an open-source code discussion has gained more attention recently as an important source of information that complements code analysis [18, 5]. In our experiments, we used LKML discussions and the Linux Kernel patches to predict if a new patch is going to be reverted within 6 months after acceptance. We trained our prediction models with different about of synthetic and real training data.

Bot detection has been a rapidly developing area in recent years [4, 6, 19]. Bot attack prediction and prevention is a closely related area that focuses on one impact of bots, though bot attacks may be difficult to define precisely [17, 7, 15]. Despite the fact that many techniques have been developed to identify the activities of bots, this cat-and-mouse game is likely to continue. Bots and networks of coordinated accounts are likely to operate and reach their operational goals (e.g. suppressing or promoting specific narratives) for some time before they are taken down. Their activities target specific users, narratives and discussions on specific topics. For example, if coordinated bot activity is intended to suppress certain narratives, the bots are likely to be triggered by certain keywords or hashtags. Influencers in such scenarios tend to use defensive measures such as avoiding known trigger keywords and adjusting posting schedules. In our bot attack prediction and countermeasure experiments, we used synthetic data to create a realistic environment of social network users where interactions between influencers and bots can be studied under different conditions. We studied the task of bot attack prediction (or "troll" attacks) based on actions taken by influencers.

The main contributions of this paper are (1) empirical evidence that synthetic data generated by a simulation model can improve the performance of ML models, (2) an agent-based framework that generates realistic synthetic data for various types of experiments in social media environments.

The following sections describe our Linux Kernel Mailing List and Reddit domains and our experiment results. We conclude with a discussion of characteristics that may predict the success of our approach and next steps.

2 Related work

With recent developments in ML and generative AI, synthetic data is now used in many domains for various reasons. For instance, in health care [13] and finance, [2] synthetic data can be a privacy-safe alternative to real data. A primary goal of the synthetic data in these domains is to have the same statistical characteristics as the original data, but with better resilience to privacy attacks.

Various machine learning methods that rely on training data also benefited from using synthetic data. Many deep learning models require a significant amount of annotated data, and simulated and synthetically generated training data can help [14, 11]. Nikolenkon et al. [14] distinguishes three main use cases for synthetic data in ML: (1) training ML models directly with synthetic datasets, (2) augmenting existing real datasets with synthetic data, (3) using synthetic data to preserve privacy. In this paper our experiments mostly focus on (1) and (2).

Synthetic data and augmentation of real data with synthetically generated data can also be used to reduce biases in real datasets [9, 3]. Breugel et al. proposed causally-aware generative neural networks to generate tabular data from real data with biases [3]. An explicit causal model was used to debias the generative network. Similarly, our Reddit simulation renders a synthetic dataset that is based on real data with infused behaviors of specific actors. We used agent-based simulation to recreate coordinated bot actions ("troll" attacks) and influencers' counter actions in the environment of real users (agents trained on real data).

Generating high-quality synthetic data that properly captures all necessary properties of a real training data set is a challenging task [9, 8]. Hansen et al. [8] provides a comprehensive evaluation of generative neural networks for tabular data. To evaluate the performance of different models they tested them on a classification task and measured statistical fidelity of synthetic tabular data with KL-Divergence, Maximum Mean Discrepancy metrics. Our experiments similarly test the performance of our simulated synthetic data sets with classification tasks. Additionally, we measure the statistical differences of predictions from models trained on different amounts of synthetic and ground truth data. Our synthetic data was generated with an agent-based simulation that was designed to capture relationships among the agents (social network users). This allows us to generate a synthetic data set with coherent relationships among all events, which is hard to guarantee with other approaches.

3 DASH Framework

Our simulation models were developed using the DASH framework, designed for cognitive agent-based simulations of social networks with varying scales and levels of accuracy [12]. This framework has been successfully employed in simulating social media platforms like Twitter/X, YouTube, Reddit, and Telegram. It is implemented in Python, and supports both single-host and cluster modes

of operation for large-scale simulations. DASH uses a discrete event simulation to schedule agents efficiently. Agent actions are scheduled in the queue and then executed. Agents can activate themselves or can be triggered by other agents, in both cases by being placed at the appropriate point in the queue. Their actions update the state of the environment (e.g., social media newsfeeds or content discussions) as well as their own internal state. The DASH framework is available as a GitHub repository [1].

In this project, we extended already existing agent-based simulations of LKML and Reddit users [**?**,12]. The agents in our simulation represent social platform users. Each agent possesses a distinct set of features reflecting their past activities, such as their frequency of participation in Reddit or LKML discussions, and their frequency of posting new content. Interactions among the agents take place through commenting and posting within the simulated environment. Comments and posts contain metadata such as authors, dates, and narratives.

In these domains, the simulation outputs represent a log of agent actions and events. Our simulation model is able to closely reproduce user activity using various metrics on both user and network levels [12].

4 LKML experiment

4.1 LKML experiment setup

The Linux Kernel Mailing List (LKML) is the main collaboration venue where Linux kernel developers discuss, propose, and review all updates to Linux kernel modules. Tens of thousands of developers contributed to LKML. Each LKML message is an email. The majority of the emails discuss Linux kernel patches with module updates. Different parts of the Linux kernel are associated with different maintainer groups that are responsible for a subset of files and directories in the Linux kernel repository. When a developer proposes a new patch, an informal process is followed in which a maintainer for the relevant area of the code briefly inspects the patch and assigns one or more reviewers to check it in more detail. The reviewers may ask questions of the developer before deciding whether to recommend accepting the patch or asking the developer to revise it. Any other individuals may join the discussion, but the maintainer will make the final decision.

In our experiments, DASH agents represent LKML users and whose actions are email messages that convey higher-level actions such as introducing a patch, reviewing or discussing the patch. Our simulation only generated email metadata such as the timestamp, authors, patch-related file names, involved maintainer groups, and the type of message (patch proposal, review, comment, patch rejection, patch acceptance, etc.).

Our DASH agents were trained on 6 months of LKML messages. Trained agents have an initial state that allows them to generate a continuous stream of actions (and interactions), which constitutes the synthetic data. This synthetic data can be generated for any period of time.

In our experiments, we simulated the activity of 5K LKML contributors over the course of 6 months using DASH agents trained from an earlier period of LKML data. A more detailed description of the pipeline is provided below in Figures 1 and 2. Different combinations of real and synthetic data were then used to train ML models that predict future patch reversal (whether a patch will be reversed in the next 6 months).

Synthetic data can be used to generate features of each data point (patch features) and classification labels (whether the patch is reversed in the next 6 months). The simulation by itself can also be used as a prediction model when it generates future statuse of existing patches from training data.

In this experiment we explored two experimental configurations:

- Configuration I: The first half of 2021 LKML data was used to run the simulation and predict reversal (reversed/not reversed labels) of patches in the second half of 2022 (Figure 1). In this setup, a synthetic data point consists of a *real* patch created (with all associated features) from the first half of 2021 and a simulated prediction of its ban in the future. Real data uses ground truth reversal labels. ML models were trained on various mixtures of real and synthetic data. This setup can be used to generate synthetic data points with simulated reversal labels when not all ground truth is known (or not known at all).
- Configuration II: The first half of 2021 LKML data was used to simulate the entire 2021 LKML activity (both with patch features and reversed/not reversed labels) (Figure 2). In this setup, a synthetic data point consists of a *simulated* patch (with all associated features) and a simulated prediction of its future ban. Real data uses ground truth reversal labels and real patches. Our ML models were trained on various mixtures of real and synthetic data. This setup can be used to generate synthetic data points with patch features and reversal labels.

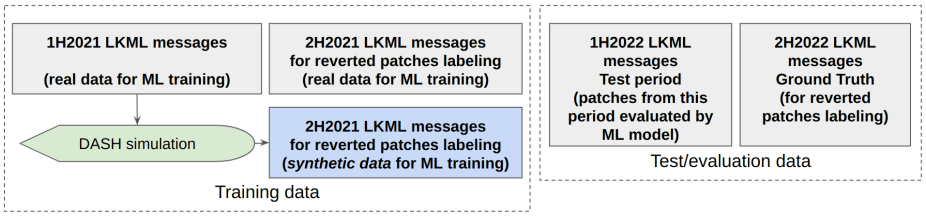

Fig. 1. Experiment training and evaluation setup I: simulation produced patch reversal labeling.

The next section discussed the results of the experiments.

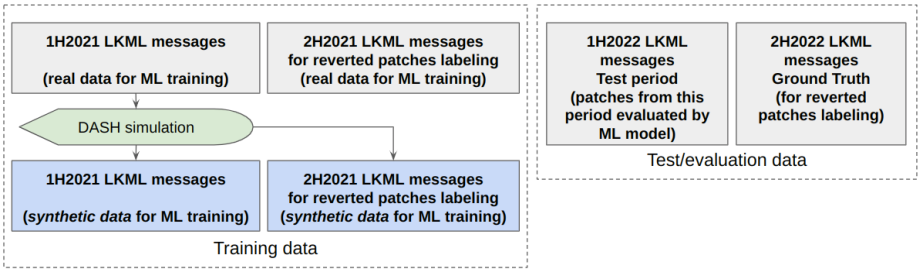

Fig. 2. Experiment training and evaluation setup II: simulation produced both patch reversal labeling and training synthetic data with features

4.2 LKML experiment results

Figure 3 shows results for the setup I (simulation was used to predict patch reversal labels). Figure 3 (a) and (b) show the impact of different mixtures of real training and synthetic data on the D-values of the KS test and RMSE. The axes show the amount of real and synthetic data used. For example, a mixture of 25% of real data and 75% of synthetic data in this setup means that 75% of reverted/not reverted labels came from simulation and 25% from real data.

Figure 3 (c) and (d) show the projections of the same results with error bars. From this figure we can see that with just half of the real data ($\geq 50\%$) adding 75% synthetic (simulated) labels can improve RMSE to the levels comparable with 100% real data. Overall we can see that adding 25% synthetic data outperforms similar models with only original data by up to 25% both in KS test and RMSE.

We can observe similar results in figure 4 for setup II (the entire simulation was used as training data). In this setup, 25% of real data and 75% synthetic data mean that 75% of data points (both features and reverted/not reverted labels) came from simulation and 25% from real data. RMSE charts show that adding synthetic data can compensate for some amount of real training data.

5 Reddit experiment

5.1 Reddit experiment setup

In this experiment, we simulate Reddit users with a focus on coordinated bot attacks ("troll" attacks) on influencers. The purpose of this simulation is to show how the simulated synthetic environment can be used to explore various actors in that environment. The simulation models the actions of regular users (background users), influencers, and bots. In this experiment setup actions of background users are trained on real observational data; therefore, background agents represent a general population of the social network (corresponding subreddit groups). Actions of bot agents and influencers are simulated algorithmically: bot

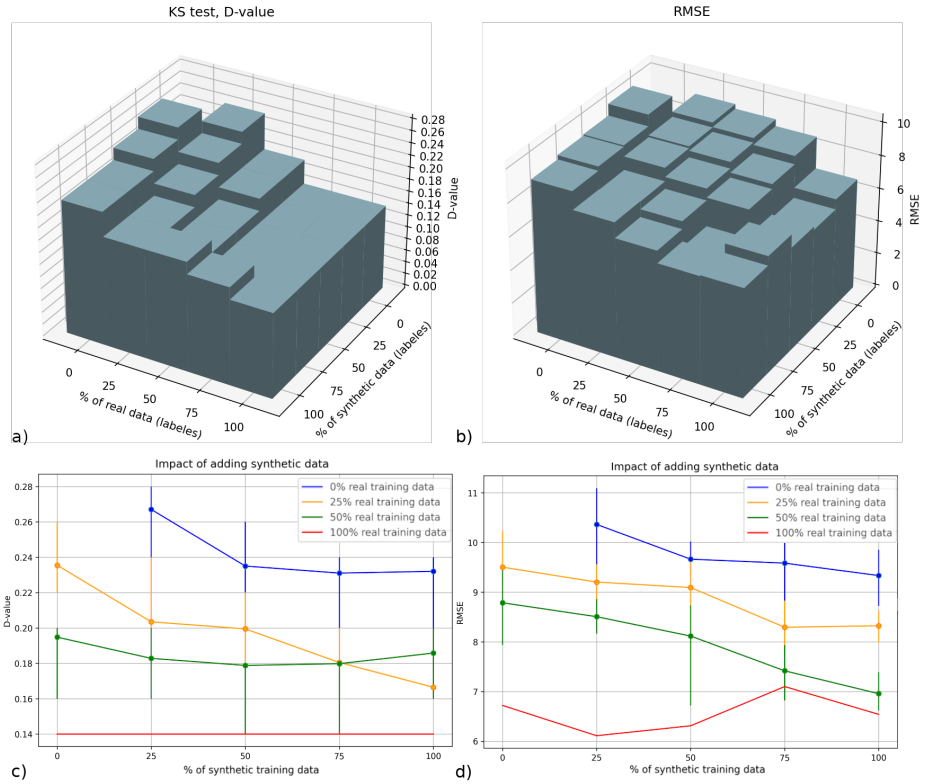

Fig. 3. LKML setup I: Impact of different mixtures of real training and synthetic data on (a,c) D-value of KS test and (b,d) on RMSE

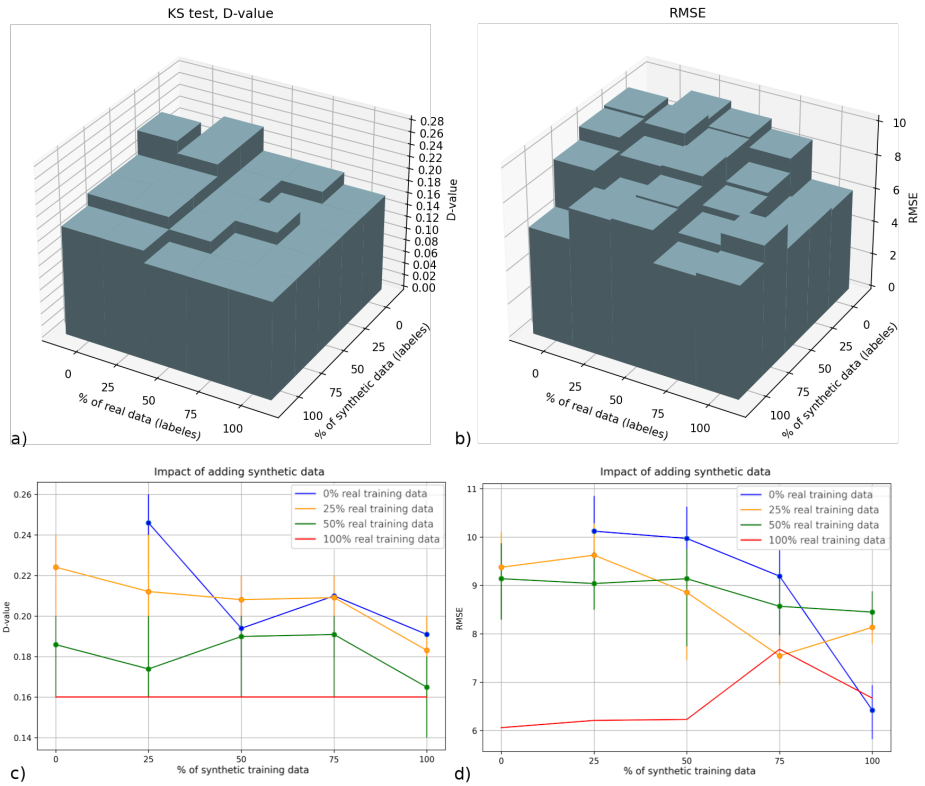

Fig. 4. LKML setup II: Impact of different mixtures of real training and synthetic data on (a,c) D-value of KS test and (b,d) on RMSE.

agents follow a predefined set of rules that trigger their attack (e.g. popularity of influencers, frequency of posts, mentioned narratives, etc.) and influencers have a predefined schedule of posts (e.g. daily or weekly posts). This simulation approach aims to reproduce observed user behavior (background users) and evaluate interactions among users that follow known rules and patterns (influencers and bots). Specifically, it aims to evaluate the effectiveness of possible counteractions by influencers against bot attacks.

We used DASH agents that were originally trained on 6 months of Reddit data. Our experiments simulated the activity of 22K Reddit users over the course of 6 months using agents trained from Reddit data.

There are therefore three types of agents in our simulation:

- Background users - they are trained on real data and pursue their goals ignoring the bot's actions.
- Influencers - they are trained on real data and pursue their goals while taking into account information learned about the bot's attack rules (e.g. they can change the schedule of their posts, adjust narratives, etc.). In our experiments, we used 100 influencer agents, which were attacked by bots.
- Bot agent - an agent that represents a group of bots that attack other users. The bots' attack on a user is modeled as a reaction to user activity. Once attacked, a user is marked as attacked for several days (in our experiments it was 3 days). This bot attack mark is visible to all agents. In our simulation, the bot agent had unlimited capacity to attack as many users as matched its attack rules.

Learning rules that trigger the attention of a coordinated network of bots on influencers' activity and then adjusting their actions can be a very effective defense measure. In this experiment, we propose a policy learning system that uses agent-based simulation to generate a mixture of real and synthetic training data. Synthetic training data is used for test scenarios and policies that attract a coordinated response from bots. The policy learning system operates in a distributed manner, where multiple influencers observe bots' reactions to their actions and the actions of their peers and using a federated learning approach [10] train a classification model that predicts bots' reaction to their future content.

Our resisting users used ML classifier predictions about the likelihood of their next action to attract bots' attack. If the likelihood of a bot attack is high, than an agent's action is postponed and rescheduled within 7 days. Only one rescheduling is allowed per original action.

ML classifier is trained on observed bot attacks on other background agents. For the integrity of the experiment, the classifier was using features agnostic to true bot activation rules. It was relying on features such as frequency of posts, narrative, number of comments, and other frequentist properties of each post.

We tested several different algorithms for this classifier and chose to present a boosted trees classifier because it showed a better performance compared to other classifiers.

5.2 Reddit experiment results

To evaluate the performance of the ban prediction classifier we used the number of bot attacks that influencers experienced. Since the classifier's predictions are used to counter a potential bot attack, the smaller the number of actual attacks the better (more useful) the classifier's predictions are. Influencers can only counteract by rescheduling their posts. It is a limited way of defeating one's post from bot attacks for two reasons: (1) rescheduling can only happen once, (2) frequent rescheduling may not help with future bans. In other words, we chose this evaluation metric for the classifier because it reflects how useful the classifier's predictions were in real settings (unlike just precision, accuracy and F1 scores).

Figure 5 shows the total number of bot attacks that influencers experienced in 3 months in experiments with different amounts of training. Error bars show 95% confidence intervals. More training improves the quality of the classifier (bot attack predictions) which in turn leads to a more effective reaction from influencers (a better rescheduling tactic). From the figure we can see that the bot attack prediction classifier trained on 6 months of data avoids almost the same number of bot attacks as rescheduling with perfect predictions.

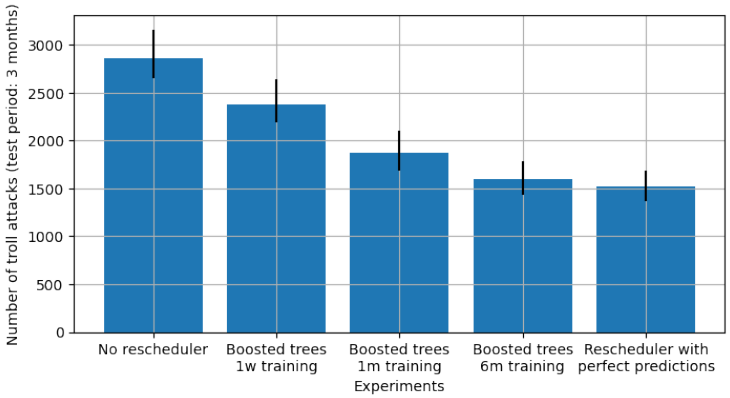

Fig. 5. Reddit experiment: the number of bots attacks in 3 months in different experiments.

6 Discussion

We showed that ML algorithms augmented with synthetic data outperform similar algorithms with only original data by up to 25% both in KS distance and RMSE. We also showed that we could achieve similar results to those achieved with a real dataset by using 60% of that data augmented with synthetic data.

Finally, in a simulation domain without access to ground-truth data, we use synthetic data to demonstrate the feasibility of learning a strategy.

We illustrated the approach in two different problem areas: prediction of code quality based on an open-source code discussion and Bot detection. Synthetic data has a different purpose in the two experiments. The first is a prediction task, to decide whether a software patch will be reverted in a 6-month period based on discussion that takes place in the previous 6-month period. Our ML model is trained on 2021 data and used to predict reversals based on 2022 data. Within this framework there are many ways that synthetic data can be used to augment real data, and we contrasted the use of completely real data with the addition of simulated patches, conversations and reversals in 2021, based on the Dash simulator trained with 2021 data. We observed better performance when synthetic data was used in conjunction with real data, both in cases where we used all real data available — up to 25% improvements in KS and RMSE scores — and when we held real data back — achieving similar results with only 60% of the original data.

Our second task concerns learning to act on social media so as to reduce the probability that a bot-triggered "troll" attack will occur while posting or retweeting content. In this task we do not have access to real data on attacks, but simulate the troll bots and content posters against a backdrop of simulated agents trained from Reddit data. Here the synthetic data allows a feasibility test of the learning approach, allowing us to explore the conditions under which the bots' triggers are learnable. We showed that under reasonable assumptions it is possible to learn a troll attack predictor that yields avoidance success close to that of an omniscient predictor.

The results in the two domains are very encouraging for the use of synthetic data in social media domains generated by cognitive agent simulations. We intend to apply this approach in future work in both discussion and engineering social media (such as source code discussions). We also plan further experiments to evaluate the contributions to these performance enhancements that stem from accurate modeling of activity and media constraints, such as the user's tweet horizon or observed patch activity levels, compared to synthetic data generated with less faithful methods.

References

1. Dash agent-based modeling framework, https://github.com/isi-usc-edu/dash/
2. Assefa, S.A., Dervovic, D., Mahfouz, M., Tillman, R.E., Reddy, P., Veloso, M.: Generating synthetic data in finance: opportunities, challenges and pitfalls. In: Proceedings of the First ACM International Conference on AI in Finance. ICAIF '20, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3383455.3422554, https://doi.org/10.1145/3383455.3422554
3. van Breugel, B., Kyono, T., Berrevoets, J., van der Schaar, M.: Decaf: Generating fair synthetic data using causally-aware generative networks. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural

Information Processing Systems. vol. 34, pp. 22221–22233. Curran Associates, Inc. (2021)
4. Chavoshi, N., Hamooni, H., Mueen, A.: Debot: Twitter bot detection via warped correlation. In: Icdm. vol. 18, pp. 28–65 (2016)
5. Eken, B., Palma, F., Ayşe, B., Ayşe, T.: An empirical study on the effect of community smells on bug prediction. Software Quality Journal **29**, 159–194 (2021)
6. Feng, S., Wan, H., Wang, N., Li, J., Luo, M.: Twibot-20: A comprehensive twitter bot detection benchmark. In: Proceedings of the 30th ACM International Conference on Information Knowledge Management. pp. 4485–4494 (2021)
7. Fornacciari, P., Mordonini, M., Poggi, A., Sani, L., Tomaiuolo, M.: A holistic system for troll detection on twitter. Computers in Human Behavior **89**, 258–268 (2018). https://doi.org/https://doi.org/10.1016/j.chb.2018.08.008, https://www.sciencedirect.com/science/article/pii/S0747563218303832
8. Hansen, L., Seedat, N., van der Schaar, M., Petrovic, A.: Reimagining synthetic tabular data generation through data-centric ai: A comprehensive benchmark. Advances in Neural Information Processing Systems **36**, 33781–33823 (2023)
9. Jaipuria, N., Zhang, X., Bhasin, R., Arafa, M., Chakravarty, P., Shrivastava, S., Manglani, S., Murali, V.N.: Deflating dataset bias using synthetic data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)
10. Li, L., Fan, Y., Tse, M., Lin, K.Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020)
11. de Melo, C.M., Torralba, A., Guibas, L., DiCarlo, J., Chellappa, R., Hodgins, J.: Next-generation deep learning based on simulators and synthetic data. Trends in cognitive sciences **26**(2), 174–187 (2022)
12. Murić, G., Tregubov, A., Blythe, J., Abeliuk, A., Choudhary, D., Lerman, K., Ferrara, E.: Large-scale agent-based simulations of online social networks. Autonomous Agents and Multi-Agent Systems **36**(2), 38 (2022)
13. Murtaza, H., Ahmed, M., Khan, N.F., Murtaza, G., Zafar, S., Bano, A.: Synthetic data generation: State of the art in health care domain. Computer Science Review **48**, 100546 (2023). https://doi.org/https://doi.org/10.1016/j.cosrev.2023.100546, https://www.sciencedirect.com/science/article/pii/S1574013723000138
14. Nikolenko, S.I.: Synthetic data for deep learning, vol. 174. Springer (2021)
15. Orozco Camacho, A.: A study of social media trolls via graph representation learning (2023)
16. Puri, R., Spring, R., Patwary, M., Shoeybi, M., Catanzaro, B.: Training question answering models from synthetic data. arXiv preprint arXiv:2002.09599 (2020)
17. Tsantarliotis, P., Pitoura, E., Tsaparas, P.: Defining and predicting troll vulnerability in online social media. Social Network Analysis and Mining **7**, 1–15 (2017)
18. Uchôa, A., Barbosa, C., Coutinho, D., Oizumi, W., Assunçao, W.K., Vergilio, S.R., Pereira, J.A., Oliveira, A., Garcia, A.: Predicting design impactful changes in modern code review: A large-scale empirical study. In: 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). pp. 471–482. IEEE (2021)
19. Wei, F., Nguyen, U.T.: Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In: 2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA). pp. 101–109. IEEE (2019)