BotArtist: Generic approach for bot detection in Twitter via semi-automatic machine learning pipeline

Alexander Shevtsov[1,2,3], Despoina Antonakaki[1,2], Ioannis Lamprou[1], Polyvios Pratikakis[2,3], and Sotiris Ioannidis[1,2]

[1] Technical University of Crete, Greece {dantonakaki,ilamprou1}@tuc.gr
[2] Foundation for Research and Technology – Hellas, Greece
[3] University of Crete, Greece shevtsov@csd.uoc.gr

Abstract. X (aka. Twitter), as one of the most popular social networks, provides a platform for communication and online discourse. Unfortunately, it has also become a target for bots and fake accounts, resulting in the spread of false information and manipulation. This paper introduces a semi-automatic machine learning pipeline (SAMLP) designed to address the challenges associated with machine learning model development. Through this pipeline, we develop a comprehensive bot detection model named BotArtist, based on user profile features.
SAMLP leverages nine distinct publicly available datasets to train the BotArtist model. To assess BotArtist's performance against current state-of-the-art solutions, we evaluate 35 existing X bot detection methods, each utilizing a diverse range of features. Our comparative evaluation of BotArtist and these existing methods, conducted across nine public datasets under standardized conditions, reveals that the proposed model outperforms existing solutions by almost 10% in terms of F1-score, achieving an average score of 83.19% and 68.5% over specific and general approaches, respectively.

Keywords: Bot detection, Machine LearningDataset

1 Introduction

Online social media has become integral to modern communication, enabling real-time information sharing and widespread content creation. Among these platforms, Twitter/X plays a central role in news dissemination, political discourse, and social interaction. However, it has also become a hotspot for bots and fake accounts that manipulate discussions and spread misinformation. Research shows that bots actively participate in sensitive topics, including political debates (e.g., the 2016 elections in the US, Germany, Sweden, France, and Spain), the vaccination debate, and COVID-19 discussions [13, 12, 14, 15]. This widespread bot activity has raised concerns about information integrity on the platform.

Table 1. Description of selected datasets and information contained in those datasets.

Dataset	C-15	G-17	C-17	M-18	C-S-18	C-R-19	B-F-19	TwiBot-20	TwiBot-22
# Total User	5,301	2,484	14,368	50,538	13,276	693	518	229,580	1,000,000
# Human	1,950	1,394	3,474	8,092	6,174	340	380	5,237	860,057
# Bot	3,351	1,090	10,894	42,446	7,102	353	138	6,589	139,943
# Total Tweet	2,827,757	0	6,637,615	0	0	0	0	33,488,192	88,217,457
# Human Tweet	2,631,730	0	2,839,361	0	0	0	0	927,292	81,250,102
# Bot Tweet	196,027	0	3,798,254	0	0	0	0	1,072,496	6,967,355
# Graph Edges	7,086,134	0	6,637,615	0	0	0	0	33,488,192	170,185,937

Various machine learning (ML) and neural network-based bot detection methods have been proposed, often tailored to specific scenarios. However, their performance degrades in general-use cases across varying timeframes, topics, and languages [24]. Many existing approaches also neglect optimization steps such as feature selection and hyperparameter tuning, limiting their effectiveness. To address these challenges, we present a semi-automatic machine learning pipeline (SAMLP[4]) for building a general-purpose bot detection model, BotArtist[5]. SAMLP performs recursive hyperparameter tuning during feature selection and accounts for class imbalance, reducing data noise and improving real-world applicability.

We evaluate BotArtist against 35 state-of-the-art methods across nine public datasets. Two evaluation scenarios are used: dataset-specific (train/test on individual datasets) and general (train/test on a merged dataset). BotArtist achieves the top F1-score on three datasets and an average F1-score of 83.19. In general evaluation, it improves F1-score by over 9% compared to the best existing models, while using only a limited set of profile features—making it API-independent and suitable for historical data analysis.

Alongside the source code, we release the model outputs: BotArtist predictions on 10,929,533 user profiles linked to 127,275,386 tweets [23], adding a new layer of user-based annotation to existing textual datasets[6]. For consistency, we refer to the platform as Twitter throughout, aligning with most prior work.

2 Related Work

Detecting bots on Twitter remains challenging due to their increasing sophistication. Prior research typically falls into three categories—feature-based, text-based, and graph-based—each exploiting different aspects of user behavior and platform metadata.

Feature-based approaches rely on engineered attributes derived from user profiles and activity patterns. These include metadata, tweets, usernames, descriptions, temporal activity, and follow relationships [8, 11, 5]. Some methods

[4] GitHub: https://github.com/alexdrk14/SAMLP
[5] GitHub: https://github.com/alexdrk14/BotArtist
[6] Zenodo: https://zenodo.org/records/8431047

enhance scalability [18], discover unknown bots through correlation [2], or improve precision-recall balance. However, bot developers continuously adapt to detection strategies, diminishing the long-term effectiveness of these methods [5].

Text-based methods apply natural language processing techniques to tweets and profile descriptions. Techniques span sequence fingerprints, word embeddings, RNNs, attention mechanisms, transformers, and pre-trained language models [5, 11]. Several studies combine text and profile features [4, 6, 17], use unsupervised learning [5], or address multilingual content [7]. While these methods show strong results, they remain vulnerable to mimicry by bots reusing human content [31] and often underperform when used alone [9].

Graph-based techniques utilize graph analytics and geometric deep learning, employing node centrality [21], node embeddings [25], graph neural networks (GNNs) [19], and heterogeneous GNNs [9]. Recent studies merge strategies across categories [20, 7] and propose novel architectures for modeling network heterogeneity [16]. Although promising, these approaches demand substantial data and computational resources.

Despite their progress, existing bot detection techniques have clear limitations. Feature-based models often lack generalizability; text and graph-based models, while more sophisticated, are prone to overfitting and require extensive resources. Additionally, the monetization of Twitter's API [1] increases the cost of operation for many methods. In response, our work introduces a lightweight yet robust solution—relying solely on a single Twitter user object (API v1.1 or v2)—capable of accurate detection with minimal API usage and reduced operational overhead.

3 Datasets

For this research paper, we collect nine well-known publicly available datasets [3]. All selected datasets already contain ground truth labels, primarily obtained through manual analysis or crowd-sourcing. For simplicity, we label the selected datasets as follows: C-15 [22], G-17 [28], C-17 [29], M-18 [18], C-S-18 [30], C-R-19 [11], B-F-19 [10], TwiBot-20 [3], and TwiBot-22 [24]. In Table 1, we present the information provided in each dataset, along with the volume of normal and bot accounts.

Furthermore, in collaboration with [23] collect 10.929.533 Twitter profiles correlated with 127.275.386 publicly available tweets related to the public discussion topic of the 2022 Russo-Ukrainian War. Our collection of user profiles is based on the monitoring of selected topics starting from February 23, 2022, till June 23, 2023. The shared datasets contain a set of extracted features in an anonymized form of a CSV file and contain only preprocessed numerical features to protect user information. The provided dataset also provides anonymized user IDs which are identically correlated with publicly available user tweet dataset [23].

4 Methodology

We propose a semi-automatic machine learning pipeline (SAMLP) to develop a general-purpose Twitter bot detection model. This pipeline simplifies the model-building process and prevents common mistakes in data processing, feature selection, hyperparameter tuning, and evaluation. To preserve class distributions, we apply a stratified 70:30 split for training/validation and testing. The testing set remains unseen until final evaluation to avoid information leakage.

Feature selection is conducted on the train/validation set using 5-fold cross-validation with Lasso regression. Since Lasso is sensitive to class imbalance, we apply under-sampling of the majority class and repeat the process 10 times to mitigate information loss. For each run, we store the best α (regularization) value based on mean squared error. The most frequent α is then used to train Lasso on the full train/validation set. If the selected α is at the boundary of the search space, we expand the search area to find a more optimal value. This approach enables automatic, robust feature selection, adaptable to both balanced and imbalanced datasets without manual intervention.

After dimensionality reduction, we evaluate three classifiers: SVM, Random Forest, and XGBoost. For each, we define wide hyperparameter ranges and sample $C = 50$ configurations randomly. Class imbalance is addressed using class weights. We evaluate each configuration using stratified K-Fold cross-validation and select the best-performing model based on average F1-score. The final model is tested on the hold-out set, and SHAP is applied for model explainability. For binary classification, we optimize the decision threshold using the precision-recall curve. The final model is retrained on the full dataset using the optimal threshold and can be deployed in real-world scenarios. Experiments were conducted on a machine with an AMD Ryzen 9 CPU (16 cores/32 threads), 64 GB RAM, and an NVIDIA RTX 3080 GPU with 12 GB memory.

Our aim is to build a lightweight model that relies solely on Twitter profile data—ensuring compatibility with both Twitter API v1.1 and v2 and minimizing reliance on unstable textual or graph features [32]. We extract 49 features, categorized as count, real-valued, and boolean. **Count features** include raw values such as followers, friends, statuses, and list subscriptions. We also compute character-type counts (uppercase, lowercase, digits, special) for user name, screen name, and description, and count mentions, hashtags, and URLs in the description.

Real-valued features include account age (in days) and activity rates normalized by age (e.g., statuses/day). We also compute Jaccard similarity between user name and screen name, entropy of both fields, and character-type percentages relative to field length.

Boolean features identify the presence or absence of specific attributes, such as whether the account is verified, protected, includes location or URL, or uses a default profile image.

This comprehensive feature set captures user behavior and profile characteristics while ensuring compatibility with Twitter API limitations and supporting accurate bot prediction.

Table 2. The performance of each selected bot detection model, as reported in the [3] paper, is compared with that of BotArtist. Performance is measured using the F1-score. In this benchmark, each model is trained and tested on each dataset separately.

Method	Type	C-15	G-17	C-17	M-18	C-S-18	C-R-19	B-F-19	TB-20	TB-22	Average
SGBot	F	77.9	72.1	94.6	99.5	82.3	82.7	49.6	84.9	36.6	75.57
Kudugunta et al.	F	75.3	49.8	91.7	94.5	50.9	49.2	49.6	47.3	51.7	62.22
Hayawi et al.	F	85.6	34.7	93.8	91.5	60.8	60.9	20.5	77.1	24.7	61.06
BotHunter	F	97.2	69.2	91.6	99.6	82.2	82.9	49.6	79.1	23.5	74.98
NameBot	F	83.4	44.8	85.7	91.6	61.1	67.5	38.5	65.1	0.5	59.80
Abreu et al.	F	76.4	66.7	95.0	97.9	76.9	83.5	53.8	77.1	53.4	75.63
BotArtist	F	98.3	76.1	97.0	**99.7**	80.6	**88.3**	**68.4**	82.2	58.2	**83.19**
Cresci	T	1.17	-	22.8	-	-	-	-	13.7	-	-
Wei	T	82.7	-	78.4	-	-	-	-	57.3	53.6	-
BGSRD	T	90.8	35.7	86.3	90.5	58.2	41.1	13.0	70.0	21.1	56.30
RoBERTa	T	95.8	-	94.3	-	-	-	-	73.1	20.5	-
T5	T	89.3	-	92.3	-	-	-	-	70.5	20.2	-
Efthimion	FT	94.1	5.2	91.8	95.9	68.2	71.7	0.0	67.2	27.5	57.95
Kantepe	FT	78.2	-	79.4	-	-	-	-	62.2	**58.7**	-
Miller	FT	83.8	59.9	86.8	91.1	56.8	43.6	0.0	74.8	45.3	60.23
Varol	FT	94.7	-	-	-	-	-	-	81.1	27.5	-
Kouvela	FT	98.2	66.6	99.1	98.2	80.4	81.1	28.1	86.5	30.0	74.24
Santos	FT	78.8	14.5	83.0	92.4	65.2	75.7	21.0	60.3	-	-
Lee	FT	98.6	67.8	**99.3**	97.9	82.5	82.7	50.3	80.0	30.4	76.61
LOBO	FT	**98.8**	-	97.7	-	-	-	-	80.8	38.6	-
Moghaddam	FG	73.9	-	-	-	-	-	-	79.9	32.1	-
Alhosseini	FG	92.2	-	-	-	-	-	-	72.0	38.1	-
Knauth	FTG	91.2	39.1	93.4	91.3	**94.0**	54.2	41.3	85.2	37.1	69.64
FriendBot	FTG	97.6	-	87.4	-	-	-	-	80.0	-	-
SATAR	FTG	95.0	-	-	-	-	-	-	86.1	-	-
Botometer	FTG	66.9	**77.4**	96.1	46.0	79.6	79.0	30.8	53.1	42.8	63.5
Rodrifuez-Ruiz	FTG	87.7	-	85.7	-	-	-	-	63.1	56.6	-
GraphHist	FTG	84.5	-	-	-	-	-	-	67.6	-	-
EvolveBot	FTG	90.1	-	-	-	-	-	-	69.7	14.1	-
Dehghan	FTG	88.3	-	-	-	-	-	-	76.2	-	-
GCN	FTG	97.2	-	-	-	-	-	-	80.8	54.9	-
GAT	FTG	97.6	-	-	-	-	-	-	85.2	55.8	-
HGT	FTG	96.9	-	-	-	-	-	-	**88.2**	39.6	-
SimpleHGN	FTG	97.5	-	-	-	-	-	-	**88.2**	45.4	-
BotRGCN	FTG	97.3	-	-	-	-	-	-	87.3	57.5	-
RGT	FTG	97.8	-	-	-	-	-	-	88.0	42.9	-

5 Experimental Results

Following the SAMLP methodology, we develop BotArtist—a semi-automated ML-based bot detection model—and evaluate its performance using nine public datasets and the comprehensive TwiBot-22 benchmark [24], which enables fair comparison with 35 existing approaches under identical data splits. Compared models span five categories: feature-based (F), text-based (T), graph-based (G), and combinations thereof.

Table 3. The measurement of performance in the case of general bot detection approaches, involves training and testing models on all datasets. Performance is assessed using the F1-score.

Method	C-15	G-17	C-17	M-18	C-S-18	C-R-19	B-F-19	TB-20	TB-22	Total	Average
BotArtist	82.7	<u>39.9</u>	87.3	<u>99.0</u>	**80.6**	<u>73.8</u>	16.6	**80.3**	**56.9**	**63.7**	**68.5**
Lee	82.3	0.0	83.6	97.7	78.2	67.7	20.0	8.5	42.4	52.9	53.3
Abreu	<u>84.4</u>	0.3	80.1	88.4	67.1	40.8	11.7	15.6	29.0	40.4	46.3
SGBot	75.0	3.6	79.8	**99.2**	76.7	68.9	0.0	15.2	<u>43.3</u>	<u>53.8</u>	51.5
BotHunter	73.4	7.1	76.0	**99.2**	76.0	44.8	11.1	14.7	28.0	43.1	47.8
Kouvela	**95.5**	20.4	<u>94.7</u>	98.1	78.4	71.4	<u>21.0</u>	28.5	36.0	52.0	60.5
Botometer	66.9	**77.4**	**96.1**	46.0	<u>79.6</u>	**79.0**	**30.8**	<u>53.1</u>	42.8	45.3	<u>63.5</u>

5.1 Model Comparison

To assess generalizability, we design two evaluation scenarios. First, we test all models, including BotArtist, on each dataset individually using dataset-specific training and testing. This scenario reflects performance in constrained, real-world applications. As shown in Table 2, BotArtist outperforms all other methods on three datasets (M-18, C-R-19, B-F-19) and achieves the highest overall average F1-score of 83.19—an improvement of 6.5% over the best existing method. Notably, models relying on text or graph features perform poorly on datasets lacking such data. Second, to evaluate generalization, we train models on a merged dataset combining training data from all nine datasets and test on both individual and merged test sets. This setup mimics a broader real-world scenario involving varied periods, topics, and communities.

Table 3 presents results from this general-case evaluation. Among the top performers are BotArtist, Botometer [20], and SGBot. BotArtist achieves the highest total and average F1-scores, outperforming other methods by nearly 10%. These results demonstrate that well-tuned models using a compact feature set can effectively generalize and distinguish between bots and real users across diverse contexts.

6 Conclusions and Future Work

This paper presents SAMLP, a semi-automatic machine learning pipeline that streamlines feature selection, hyperparameter tuning, model evaluation, binary threshold optimization, and SHAP-based explainability. Using this pipeline, we develop BotArtist—a profile-based Twitter bot detector. Evaluated across nine datasets, BotArtist outperforms state-of-the-art methods, achieving up to a 10% improvement in total F1 score and 6.5% on individual datasets. SHAP analysis provides transparency into the model's decisions. While effective, BotArtist relies on a limited feature set, which could be targeted by adaptive bot strategies. Future work will explore its robustness over time and expand the feature space to adapt to evolving bot behavior.

Acknowledgements This work is supported by project CYBERUNITY, funded by Digital Europe Programme (DIGITAL) with GA No. 101128024.

References

1. Twitter, "Twitter API price list," 2023. [Online]. Available: https://developer.twitter.com/en/products/twitter-api. [Accessed: Sep. 9, 2023].
2. N. Chavoshi, H. Hamooni, and A. Mueen, "Debot: Twitter bot detection via warped correlation," in *Icdm*, vol. 18, 2016, pp. 28–65.
3. S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Twibot-20: A comprehensive twitter bot detection benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4485–4494.
4. P. G. Efthimion, S. Payne, and N. Proferes, "Supervised machine learning bot detection techniques to identify social twitter bots," *SMU Data Science Review*, vol. 1, no. 2, p. 5, 2018.
5. S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Satar: A self-supervised approach to twitter account representation learning and its application in bot detection," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3808–3817.
6. M. Kantepe and M. C. Ganiz, "Preprocessing framework for Twitter bot detection," in *2017 International conference on computer science and engineering (ubmk)*, IEEE, 2017, pp. 630–634.
7. J. Knauth, "Language-agnostic twitter-bot detection," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 550–558.
8. S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, 2018.
9. S. Feng, H. Wan, N. Wang, and M. Luo, "BotRGCN: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2021, pp. 236–239.
10. K.-C. Yang et al., "Arming the public with artificial intelligence to counter social bots," *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 48–61, 2019.
11. M. Mazza, S. Cresci, M. Avvenuti, W. Quattrociocchi, and M. Tesconi, "Rtbust: Exploiting temporal patterns for botnet detection on twitter," in *Proceedings of the 10th ACM conference on web science*, 2019, pp. 183–192.
12. A. Shevtsov, C. Tzagkarakis, D. Antonakaki, and S. Ioannidis, "Identification of Twitter Bots Based on an Explainable Machine Learning Framework: The US 2020 Elections Case Study," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, 2022, pp. 956–967.
13. Y. Golovchenko, C. Buntain, G. Eady, M. A. Brown, and J. A. Tucker, "Cross-platform state propaganda: Russian trolls on Twitter and YouTube during the 2016 US presidential election," *The International Journal of Press/Politics*, vol. 25, no. 3, pp. 357–389, 2020.
14. A. Shevtsov et al., "What Tweets and YouTube comments have in common? Sentiment and graph analysis on data related to US elections 2020," *Plos one*, vol. 18, no. 1, p. e0270542, 2023.
15. G. K. Shahi, A. Dirkson, and T. A. Majchrzak, "An exploratory study of COVID-19 misinformation on Twitter," *Online social networks and media*, vol. 22, p. 100104, 2021.

16. S. Feng, Z. Tan, R. Li, and M. Luo, “Heterogeneity-aware twitter bot detection with relational graph transformers,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 3977–3985.
17. M. Kouvela, I. Dimitriadis, and A. Vakali, “Bot-Detective: An explainable Twitter bot detection service with crowdsourcing functionalities,” in *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, 2020, pp. 55–63.
18. K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, “Scalable and generalizable social bot detection through data selection,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 1096–1103.
19. S. H. Moghaddam and M. Abbaspour, “Friendship preference: Scalable and robust category of features for social bot detection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1516–1528, 2022.
20. K.-C. Yang, E. Ferrara, and F. Menczer, “Botometer 101: Social bot practicum for computational social scientists,” *Journal of Computational Social Science*, vol. 5, no. 2, pp. 1511–1528, 2022.
21. A. Dehghan et al., “Detecting bots in social-networks using node and structural embeddings,” *Journal of Big Data*, vol. 10, no. 1, pp. 1–37, 2023.
22. S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “Fame for sale: Efficient detection of fake Twitter followers,” *Decision Support Systems*, vol. 80, pp. 56–71, 2015.
23. A. Shevtsov, C. Tzagkarakis, D. Antonakaki, P. Pratikakis, and S. Ioannidis, "Twitter dataset on the Russo-Ukrainian war", arXiv preprint arXiv:2204.08530, 2022.
24. Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., Zheng, Q., Zhang, W., Lei, Z., Yang, S., et al.: TwiBot-22: Towards graph-based Twitter bot detection. *Advances in Neural Information Processing Systems* **35**, 35254–35269 (2022)
25. Pham, P., Nguyen, L.T.T., Vo, B., Yun, U.: Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. *Information Systems* **103**, 101771 (2022)
26. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152 (1992)
27. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al.: XGBoost: Extreme gradient boosting. *R Package Version* **0.4-2**(1), 1–4 (2015)
28. Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L., Crowcroft, J.: Of bots and humans (on Twitter). In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 349–354 (2017)
29. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: Social fingerprinting: Detection of spambot groups through DNA-inspired behavioral modeling. *IEEE Trans. Dependable Secure Comput.* **15**(4), 561–576 (2017)
30. Cresci, S., Lillo, F., Regoli, D., Tardelli, S., Tesconi, M.: FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In: *Proc. Int. AAAI Conf. on Web and Social Media (ICWSM)*, vol. 12 (2018)
31. S. Cresci, “A decade of social bot detection,” *Communications of the ACM*, vol. 63, no. 10, pp. 72–83, 2020.
32. A. Shevtsov, D. Antonakaki, I. Lamprou, I. Kontogiorgakis, P. Pratikakis, and S. Ioannidis, “Russo-Ukrainian War: Prediction and explanation of Twitter suspension,” in *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2024, pp. 348–355.