Software Impacts 13 (2022) 100333

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Explainable machine learning pipeline for Twitter bot detection during the 2020 US Presidential Elections

Alexander Shevtsov [a,c,*], Christos Tzagkarakis [a], Despoina Antonakaki [a], Sotiris Ioannidis [b]

[a] Institute of Computer Science, Foundation for Research and Technology-Hellas, Greece
[b] School of Electrical and Computer Engineering, Technical University of Crete, Greece
[c] Department of Computer Science, University of Crete, Greece

ARTICLE INFO

Keywords:
Machine learning
Twitter bot detection
Model explainability

ABSTRACT

This study introduces a novel, reproducible and reusable Twitter bot identification system. The system uses a machine learning (ML) pipeline, fed with hundreds of features extracted from a Twitter corpus. The main objective of the proposed ML pipeline is to train and validate different state-of-the-art machine learning models, where the eXtreme Gradient Boosting (XGBoost) model is selected since it achieves the highest detection performance. The Twitter dataset was collected during the 2020 US Presidential Elections, and additional experimental evaluation on distinct Twitter datasets demonstrates the superiority of our approach, in terms of high bot detection accuracy.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-77
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/3418007/tree/v1
Legal Code License	MIT License
Code versioning system used	GitHub
Software code languages, tools, and services used	Python, sci-kit learn, Twitter API, Botometer API, BotSentinel
Compilation requirements, operating environments & dependencies	Python3 is required with the following libraries: sklearn, imblearn, xgboost, shap, numpy, pandas, matplotlib, ast
If available Link to developer documentation/manual	https://github.com/alexdrk14/USBotDetection
Support email for questions	shevtsov@ics.forth.gr

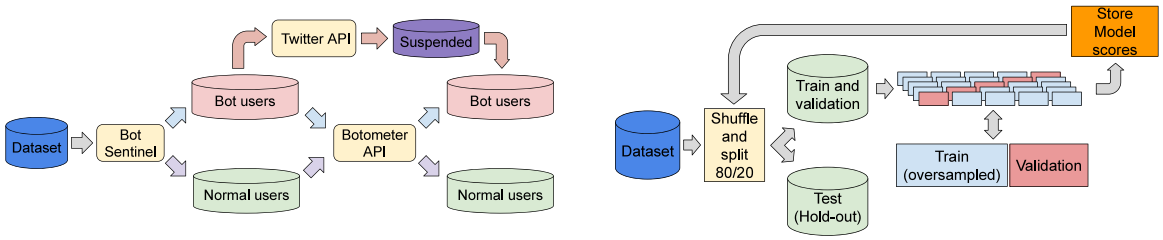
1. Introduction

The current article presents the design and implementation of the methodology introduced in "Identification of Twitter Bots Based on an Explainable Machine Learning Framework: The US 2020 Elections Case Study", where a novel Twitter bot identification system is described. We explore the impact of the implemented bot detection system, in terms of a reproducible and reusable scheme that could be exploited by the research community. The provided machine learning (ML) pipeline is ingested with hundreds of features extracted from the collected Twitter corpus, used to train and validate different state-of-the-art ML models. Based on the bot detection accuracy, measured through various metrics, such as the area under the precision–recall curve (PR-AUC), the receiver operating curve (ROC-AUC) and the F1-score, the eXtreme Gradient Boosting (XGBoost) model is selected, since it provides the highest detection performance. Our study is also accompanied with an explainability module, by adopting the Shapley Additive Explanations (SHAP) method for explaining the ML model predictions. The Twitter dataset was collected during the 2020 US Presidential Elections, and an

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Institute of Computer Science, Foundation for Research and Technology-Hellas, Greece.
E-mail addresses: shevtsov@ics.forth.gr (A. Shevtsov), tzagarak@ics.forth.gr (C. Tzagkarakis), despoina@ics.forth.gr (D. Antonakaki), sotiris@ece.tuc.gr (S. Ioannidis).

https://doi.org/10.1016/j.simpa.2022.100333
Received 2 June 2022; Accepted 10 June 2022



(a) Labeling pipeline (b) Bot detection pipeline

Fig. 1. Developed pipelines: (a) ground truth labeling process and (b) ML model tuning using K-Fold cross-validation.

additional experimental evaluation on distinct Twitter datasets demonstrates the superiority of our approach in terms of high bot detection accuracy.

In this work, we build a ML system over a large collected dataset, in order to detect bot Twitter accounts. The current study provides answers to the following research questions:

- Is it possible to implement and fine-tune a ML-based bot detection model to efficiently apply it to the US 2020 Elections dataset?
- Which types of features can be extracted from the Twitter application programming interface (API) to promote high performance?
- Does the proposed ML model act as a black box or could the ML model's mechanism be "unlocked" in order to investigate how it yields its predictions?

The dataset is publicly available, as well as the implemented ML pipeline which can be adjusted and adopted in other domains, too.

2. Methodology

In order to answer these questions, we initially tackle the following issues: data acquisition, ground truth bot labeling and accurate bot detector development. Towards this direction, we collect a Twitter dataset through Twitter API, where we narrow our search on the 2020 US Presidential Elections topic. For this purpose, we manually identify the most popular hashtags during the US Elections period and manage to collect 15.6 million tweets with 3.2 million users from September 1st, 2020 to November 3rd, 2020.

The Twitter API does not provide any labeling information whether an account is bot or not, and thus an account labeling methodology should be taken into consideration. In addition, Twitter API is able to identify an account as bot and suspend it. However, we cannot retrieve this information beforehands during our data acquisition phase. The Twitter suspension mechanism is one example of ground truth labeling, but in most cases it requires a longer period of time until a user account's suspension. Moreover, the user suspension does not always reflect a clear evidence whether an account is bot or not, since the account may be suspended due to rules violation. In order to tackle this issue, we develop a ground truth labeling procedure as depicted in Fig. 1.

During the labeling process, we utilize two off-the-shelf ML-based bot detection tools. We combine the detection results from both tools (i.e., Botometer[1] and BotSentinel[2]) via majority voting and keep only the similar labels. This procedure allows us to reduce the noise and miss-labeling phenomena, since none of the existing tools achieves 100% accuracy. The developed methodology provides ground truth labels and reduces dramatically the computational time from 650 days to only 18 days.

Next, a ML pipeline is developed that takes as input multiple numerical user features in the form of CSV file. A series of steps (i.e., feature selection, hyper-parameter tuning via K-fold cross-validation, selected ML model performance evaluation) are followed to detect whether a Twitter account is bot or not, as well as explaining the classification outcome through the SHAP[3] method.

The implemented ML pipeline can be utilized for additional classification tasks (not only limited to bot detection), even in a multi-class classification task, since it provides a separate fine-tuned model configuration that can be modified for any other use case. Since our methodology can provide an automated pipeline of the model fine-tuning and performance measurement, ML implementation knowledge is not required by the user.

3. Impact

The proposed explainable ML framework can be used by developers and researchers to accurately detect bot accounts on Twitter, based on state-of-the-art ML models. The Twitter bot detection tool evaluates state-of-the-art ML models, optimizing the feature selection and modeling steps and their hyper-parameters. By comparison with already existing methods, the proposed implementation allows the detection of Twitter bot accounts by providing a combined explanation of the classification results based on the SHAP method. As we mentioned earlier, the presented software is already utilized on the 2020 US Presidential Elections dataset, obtaining highly accurate results and it is already published in the proceedings of the International AAAI Conference on Web and Social Media (ICWSM) 2022.[4]

4. Conclusions and future work

The presented bot detection approach achieves high performance on the collected dataset (collected in different time periods), as compared to the training data portion. The performance is reduced by only 2% according to F1 score (i.e., from 0.916 to 0.896) and 0.03% according to ROC-AUC (i.e., from 0.98 to 0.977). These results suggests high generalization ability of our ML model. As future work, we intend to evaluate the bot detection performance on training and test data obtained in different time periods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the reviewers for their valuable comments. This document is the result of the research projects CONCORDIA (grant number 830927), CyberSANE (grant number 833683) and PUZZLE (grant number 883540) co-funded by the European Commission, with (EUROPEAN COMMISSION Directorate-General Communications Networks, Content and Technology).

[1] https://botometer.osome.iu.edu/.
[2] https://botsentinel.com/info/about.
[3] https://shap.readthedocs.io/en/latest/index.html.
[4] https://www.icwsm.org/2022/index.html/.