Bot Account Analysis on Social Media with Artificial Intelligence Support on Twitter Example

Received: 1 January 2023; Accepted: 28 February 2023 *Research Article*

Refik Söylemez
Department of Computer Engineering
Istanbul Ticaret University
Istanbul, Türkiye
refiksoylemez@gmail.com
ORCID: 0000-0003-4580-067X

Ali Boyacı
Deparment of Computer Engineering
Istanbul Ticaret University
Istanbul, Türkiye
aboyaci@ticaret.edu.tr
ORCID: 0000-0002-2553-1911

***Abstract*—Launched in 2006 and available in 33 languages, Twitter is a social media platform that initially allowed users to share messages of up to 140 characters. It has since evolved into a platform used for various purposes, including communication, organization, sales and marketing, and microblogging. There have been numerous studies on data analysis (emotion, influence, education and training opportunities, political polarization, etc.) and data input (bot analysis, etc.) related to "tweets" - messages entered by users on Twitter since its inception. These studies have focused on analyzing bot tweets and accounts in order to prevent Twitter messages from informing people of false news. The accuracy performance of these analyses carried out with machine learning methods varies depending on the selection of training data used to create the model. In this study, the impact of randomly selected different training data on model performance was focused on and examined.**

Keywords—Twitter, machine learning, bot tweet.

I. INTRODUCTION

As social media technology and popularity have grown, people today are organizing on Twitter and attempting to influence social peace and governments by starting various conflicts and wars. Many uprisings use bot accounts and try to change the agenda with bot tweets. To eliminate this problem that threatens society, it is necessary to discover bot accounts and take appropriate action. The detection of bot accounts and tweets is of great importance. If undetected, bot tweets can inform people of false news and guide them toward unrealistic trends. Today's widespread use of Twitter bots has led to academic research focusing on this subject. For example, one study has pointed out that between 9% and 15% of active Twitter accounts are bot accounts. The study discusses the interaction between simple bots and those that mimic human behavior. It also attempts to classify different types of bot accounts - those that send spam, promote themselves, or publish content from linked applications - through clustering analysis. The study highlights the various purposes for which bot accounts are used [1]. The development of bots' ability to respond and process information like humans are expected to have a wide-ranging impact and potentially lead to various sociological, psychological, political, and even economic effects. Therefore, in recent years, research in this area has increased, taking into account the impact of detecting bot tweets and accounts.

II. CONCEPTUAL AND THEORETICAL FRAMEWORK / LITERATURE

A study from 2017 examined the use of Twitter in political communication, including the behavior of political users on the platform, the role of Twitter in political discussions, and the use of the platform in election campaigns, with a focus on the influence of bot accounts in the 2016 US elections [2].

Social media bots are prevalent today. As they are developed with the ability to respond like humans, they become increasingly influential. Therefore, detecting bot tweets and accounts using artificial intelligence algorithms is very important and has become a topic of much research in recent years. Bot accounts created by automated programs have targeted Twitter's increasing user numbers and overall structure. These bots have provided a platform for the spread of both good-faith content, such as news and blog updates, and spam or malicious content. Bots generally aim to follow many user accounts and be followed back randomly. Many efforts have been made to solve the problem of spam bots on social platforms. Different methods, such as extracting the text content of tweets, redirecting embedded URL addresses in other posts, and classifying the opening pages of URLs, have been tried to address this issue. A composite tool that can match tweets with commonly used basic templates has been proposed, going beyond the difficulty of labeling tweets without URLs as spam tweets [3]. A bio-inspired technique was introduced to model online social media user behaviors [4]. Instead of using more complex traditional feature engineering or natural language processing (NLP) tools, word embeddings were tried to encode tweets. This advantage allows the bot detection scheme to be faster and easier to implement and deploy. A Recurrent Neural Network (RNN) model using word embeddings, particularly a BiLSTM, was introduced to distinguish Twitter bots from human accounts. The study, which does not require prior knowledge or assumptions about user profiles, friendship networks, or past behaviors of the target account and is based only on tweets, and does not require heavy feature engineering, it is the first to develop an RNN model using word embeddings to detect bots. Their experiments on the publicly available Cresci-2017 dataset showed that models without hand-crafted feature engineering could achieve similar performance compared to existing studies [5]. The study evaluated the effectiveness of 30 classification algorithms for detecting bot tweets using supervised classification. Tree-based supervised classifiers performed the best, with the Random Forest classifier achieving the highest accuracy. The study also applied standard boosting and bagging techniques to further improve the accuracy of the Random Forest classifier [6]. Additionally, the study presents a system that utilizes supervised machine learning techniques to dynamically detect Twitter bot accounts. The classification results show a very high accuracy rate for this specific application [7]. An unsupervised method for detecting spam robots by comparing their behaviors to

identify similarities among automatic accounts has been proposed. This bio-inspired method for modeling online user behaviors is called "Digital DNA" sequences. Extracting digital DNA from an account means associating it with a series of codes that encode behavioral information for that account. Although it achieves good detection performances, many handcrafted behavioral features are still required [4]. There are also methods for identifying Twitter bots that rely on the assumption that bots differ from humans in fundamental ways. These differences can be divided into two categories: technical differences and differences based on purpose. Bots are computer programs that can act instantly, while humans need time to think and may be busy with other tasks. Therefore, it can be assumed that the timing and direction of the content published differs from human behavior to bot behavior. Additionally, bots have clear goals, such as disseminating political messages and making references. Bots carefully bring specific content to the attention of users, hashtags, and URLs [8].

III. Method

In this study, the Social Honeypot dataset was used. The dataset was collected on Twitter from December 30, 2009, to August 2, 2010. It includes 22,223 spamming users, their following counts over time, 2,353,473 tweets, 19,276 legitimate users, their following counts over time, and 3,259,693 tweets [6]. After downloading the dataset made available as open source, data preprocessing steps were applied in the Python environment in the first step. These steps include removing punctuation marks, converting words to lowercase, removing repeating words, and lemmatization. Verbal expressions must be made meaningful for machine learning or deep learning algorithms. Therefore, words must be expressed numerically. Algorithms such as One Hot Encoding, TF-IDF, Word2Vec, FastText, and Count Vectorizer, known as word embedding techniques used to solve such problems, allow words to be expressed mathematically.

In the second step, the data set, cleaned with preprocessing steps, is transformed into a vector form with Count Vectorizer and becomes input to the machine learning model in a format ready to be used.

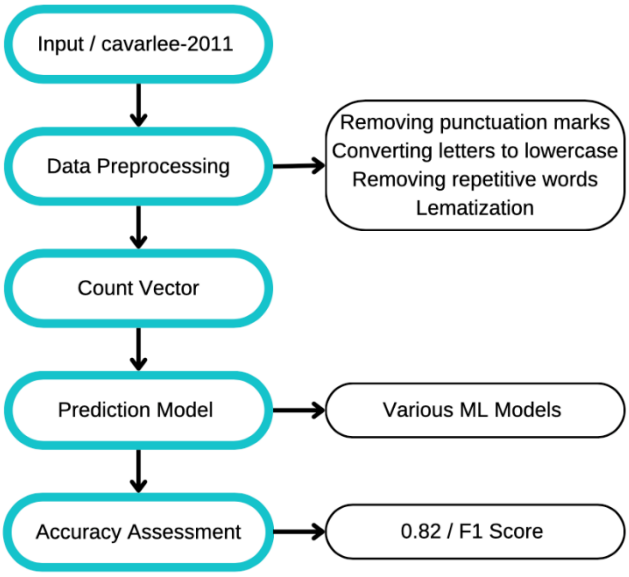

Fig. 1. Simple Workflow

IV. Results

A. Training with Default Hyperparameters

After splitting our dataset, which contained 1000 tweets, into a 25% test set and a 75% training set, the training phase was completed using various machine learning algorithms with their default parameters. The following accuracy results were obtained.

TABLE I. Accuracy Assessment 1

Algorithm	F1 Score Test	F1 Score Train
GaussianNB	0.83	0.99
MultinomialNB	0.81	0.97
ComplementNB	0.81	0.97
LinearSVC	0.80	0.99
SGDC	0.80	0.99
Logistic Regression	0.78	0.99
Random Forest	0.76	1.00
Decision Trees	0.74	1.00
Bagging	0.74	0.97
KNeighbors	0.69	0.68
AdaBoost	0.63	0.75

Based on the results obtained using the same test data, it can be seen that the best results were obtained using the Gaussian Naive Bayes algorithm when evaluating based on the F1 score. In the performance ranking, the Linear Support Vector Machines algorithm stands out immediately after the Naive Bayes algorithms, which are defined by probability principles. Machine learning models that can generalize well in problems with large feature spaces, such as SVM, produce better results than other models in text classification due to this power [9].

B. Training with Cross-Validation and Hyperparameter Optimization

The Logistic Regression, Decision Tree, Linear Support Vector Machines, Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, Random Forest, and K Nearest Neighbors algorithms were trained using hyperparameter optimization to improve the results obtained with default parameters. The Cross Validation method was also applied during training in addition to the previous study. Cross-validation is a method used to evaluate and compare machine learning models by splitting the data into two sets: one for training the model and the other for validation. In simple cross-validation, the training and validation sets are rotated in such a way that each data point has an opportunity

to be used for validation. This process involves using a diverse set of data for training and testing the model, which can provide a more accurate evaluation of the model's performance [10]. This ensures that each classifier is trained with a large number of parameter combinations and also prevents overfitting. The accuracy results obtained from the process are shown below.

TABLE II. ACCURACY ASSESSMENT

Algorithm	F1 Score Test	F1 Score Train
Random Forest	0.82	1.00
GaussianNB	0.82	0.99
Logistic Regression	0.82	1.00
MultinomailNB	0.82	0.99
ComplementNB	0.82	0.99
LinearSVC	0.71	0.99
SGDC	0.71	0.99
KNeighbors	0.69	1.00
Decision Trees	0.48	0.60

TABLE III. ACCURACY ASSESSMENT 3 - RANDOM STATE: 10

Algorithm	F1 Score Test	F1 Score Train
LinearSVC	0.81	0.99
MultinomailNB	0.81	0.97
ComplementNB	0.81	0.97
GussiamNB	0.81	0.99
Logistic Regression	0.80	0.99
SGDC	0.78	0.99
Random Forest	0.76	0.99
Decision Trees	0.74	1.00
KNeighbors	0.67	0.67

According to the performance graphs on the above, it can be seen that cross-validation and hyperparameter optimization had a positive effect on the performance of most algorithms. When examining the algorithms, it can be seen that Naive Bayes-based classifiers performed well in both experiments. This is because the Naive Bayes algorithm classifies based on probability principles. Studies have demonstrated that such classifiers give better results in language processing problems than linear classifiers.

C. Training with Different Random State Parameters

75% of training data is randomly split. However, this randomness can be controlled by a parameter. When this parameter changes, the data on which the machine learning models are also trained changes; therefore, the model results are also affected by this. The training data was created with four different randomness parameters to investigate the effect of training data split into different randomness on model results. A total of 36 results were obtained by training nine different algorithms with four randomly generated training data. The results obtained are shown in the tables below.

TABLE IV. ACCURACY ASSESSMENT 4: RANDOM STATE: 40

Algorithm	F1 Score Test	F1 Score Train
GussianNB	0.84	0.99
LinearSVC	0.83	0.99
MultinomailNB	0.82	0.98
ComplementNB	0.82	0.98
Logistic Regression	0.80	0.99
SGDC	0.79	0.99
Random Forest	0.75	1.00
Decision Trees	0.72	1.00
KNeighbors	0.65	0.69

V. DISCUSSION AND RESULTS

During the training part of a machine learning model, the data available is divided into two parts: training and testing. During training, the model does not see the test data. Therefore, the data that will be entered as training data for the model may vary depending on the given randomness parameter. A large number of training-test combinations are created depending on the size of the data. Therefore, it will be logical to take the accuracy averages of models trained with different randomness parameters to determine the most reliable accuracy. In addition, optimization can be done for the randomness parameter according to the desired performance in the study. According to the experiment results, the change in this parameter did not cause significant changes in performance. That is, there is no need to apply the optimization for hyperparameters for the randomness. According to the results, the change in randomness only caused performance difference between 0-2%

Across experiments, Gaussian Naive Bayes and Linear Support Vector Machines consistently outperformed tree-based algorithms in text classification. Their probabilistic nature and linear approach make them superior choices.

TABLE V. ACCURACY ASSESSMENT 5: RANDOM STATE: 100

Algorithm	F1 Score Test	F1 Score Train
SGDC	0.81	0.99
GaussianNB	0.80	0.99
MultinomailNB	0.80	0.98
ComplementNB	0.79	0.99
LinearSVC	0.88	0.99
Logistic Regression	0.75	0.99
Random Forest	0.75	1.00
Decision Trees	0.75	1.00
KNeighbors	0.65	0.69

TABLE VI. ACCURACY ASSESSMENT 6: RANDOM STATE: 300

Algorithm	F1 Score Test	F1 Score Train
GussianNB	0.84	0.99
LinearSVC	0.82	0.99
MultinomailNB	0.81	0.98
Logistic Regression	0.82	0.98
ComplementNB	0.81	0.99
SGDC	0.80	0.99
Random Forest	0.77	0.99
Decision Trees	0.76	1.00
KNeighbors	0.69	0.68

CONTRUBITION OF THE AUTHORS

Asst. Prof. Dr Ali Boyacı played a pivotal role in guiding and supervising the research project. His expertise in relevant fields provided invaluable insights that shaped the direction of the study. Hi were actively involved in conceptualizing the research design, formulating research questions, and advising on methodology.

As the lead author of this article, Refik, the focus was on conducting extensive research regarding the impact of training data selection on the performance of machine learning models in analyzing Twitter data. This encompassed designing and executing experiments to systematically evaluate the influence of different training data sets on the accuracy of the models. Additionally, a comprehensive analysis of the results was undertaken, drawing meaningful conclusions and insights from the findings.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor, Asst. Prof. Ali Boyacı for their invaluable guidance and support throughout the research process. Their expertise and mentorship were instrumental in the successful completion of this work.

CONFLICT OF INTEREST

There is no any conflicts of interest between the authors.

STATEMENT OF RESEARCH AND PUBLICATION ETHICS

Research and publication ethics were observed. Ethics committee approval was obtained for research conducted in all branches of science that requires ethics.

TABLE VII. ACCURACY COMPARISON

Algorithm	Minimum F1 Score	Maximum F1 Score	Mean F1 Score
GussianNB	0.81	0.84	0.82
LinearSVC	0.79	0.83	0.81
MultinomailNB	0.80	0.82	0.81
Logistic Regression	0.78	0.82	0.80
ComplementNB	0.80	0.82	0.81
SGDC	0.79	0.80	0.81
Random Forest	0.75	0.77	0.76
Decision Trees	0.73	0.76	0.75
KNeighbors	0.65	0.69	0.67

REFERENCES

[1] Varol, O., Ferrara, E., Davis, C. A., Menczer, F., & Flammini, A. “Online Human-Bot Interactions: Detection, Estimation, and Characterization”, 2017.

[2] Campos Domínguez, E. M. (2017). Twitter y la comunicación política. El profesional de la información, 26(5),785-794.

[3] HongyuGao,YanChen,KathyLee,DianaPalsetia,andAlokNChoudhary. 2012. Towards online spam filtering in social networks.. In NDSS, Vol. 12. 1–16.

[4] S. Cresci, M. Petrocchi, A. Spognardi, and S. Tognazzi, “On the capability of evolved spambots to evade detection via genetic engineering,” *Online Social Networks and Media*, vol. 9, pp. 1–16, 2019.

[5] Feng Wei and Uyen Trang Nguyen, “Twitter Bot Detection Using Bidirectional Long Short-term Memory Neural Networks and Word Embeddings” in IEEE TPS 2019.

[6] K. Lee, B. D. Eoff, and J. Caverlee, “Seven months with the devils: A long-term study of content polluters on twitter,” in Proc. Fifth Int. AAAI Conf. Weblogs Social Media, 2011.

[7] M. Alsaleh, A. Alarifi, A. M. Al-Salman, M. Alfayez, and A. Almuhaysin, "Tsd: Detecting sybil accounts in twitter," in *Proc. 13th Int. Conf. Mach. Learning and Appl.*, 2014.

[8] Jürgen Knauth. 2019. Language-Agnostic Twitter-Bot Detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 550–558, Varna, Bulgaria. INCOMA Ltd.

[9] Bellman, R.E. Adaptive Control Processes; Princeton University Press: Princeton, NJ, USA, 1961. [Google Scholar]

[10] Refaeilzadeh, P., Tang, L., Liu, H. (2009). Cross-Validation. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA