# Twitter bot detection using deep learning

Adam Kenyeres[1], György Kovács[1]

Luleå University of Technology
adamkenyeres@gmail.com, gyorgy.kovacs@ltu.se

**Abstract.** Social media platforms have revolutionized how people interact with each other and how people gain information. However, social media platforms such as Twitter and Facebook quickly became the platform for public manipulation and spreading or amplifying political or ideological misinformation. Although malicious content can be shared by individuals, today millions of individual and coordinated automated accounts exist, also called bots which share hate, spread misinformation and manipulate public opinion without any human intervention. The work presented in this paper aims at designing and implementing deep learning approaches that successfully identify social media bots. Moreover we show that deep learning models can yield an accuracy of 0.9 on the PAN 2019 Bots and Gender Profiling dataset. In addition, the findings of this work also show that pre-trained models will be able to improve the accuracy of deep learning models and compete with Classical Machine Learning methods even on limited dataset.

## 1 Introduction

Social media bot detection has more than a decade of history. In the early days, most detection methods were based on supervised machine learning algorithms, based solely on the information of individual accounts. However, as time went by, bots became more sophisticated and coordinated and distinguishing bots from real accounts became almost impossible even for humans (Cresci et al., 2017). At this point, researchers turned to unsupervised machine learning algorithms which focused on groups of accounts rather than individual ones. The problem with the previous approaches is that they cannot be generalized as they are platform dependent. Recently the PAN 2019 Bots and Gender Profiling task has taken place, with the intention of using only the textual characteristics (Tweet) of social media accounts (Rangel and Rosso, 2019). This has the advantage of creating platform-independent solutions as these methods do not rely on specific social media characteristics. The work presented here will be based on the PAN 2019 Bots and Gender Profiling task. In the following sections we show that deep learning models can compete with Classical Machine Learning (CML) approaches even on a limited dataset. Moreover, this work also shows that pre-trained models improve the accuracy of bot detectors.

### 1.1 Motivation

Today, platforms such as Twitter have become tools for manipulating public opinion and spreading or amplifying political misinformation. However, large scale manipulation and sharing of politically polarized content is not only done by humans but by social media bots (Bessi and Ferrara, 2016).

Bots can even play a key role in political elections. Although at the time it was not clear, today we know that social media bots played a key role in the 2016 U.S. presidential elections (Bessi and Ferrara, 2016) by spreading divisive messages which may have contributed to Trump's victory[1]. During the Brexit referendum in the United Kingdom, bots manipulated public opinion in favor of the country to leave the European Union (Howard et al., 2018). Bots also interfered with the French presidential elections in 2017 by spreading the Macron-Leaks campaign against Emmanuel Macron (Ferrara, 2017). As we can see bots actively shape public opinion, usually by negative campaigning (Howard et al., 2018) and influence individuals who may not be able to distinguish between human or bot-generated content. Therefore, countermeasures have to be taken in order to neutralize bots that exploit and spread misinformation by creating state-of-the-art bot detection methods.

### 1.2 Problem Definition

The project will be based on the PAN 2019 Bots and Gender Profiling task (Rangel and Rosso, 2019) where author profiling had to be solved by classifying Twitter feeds as bots or humans, based solely on the account's textual form of the tweets without any additional information, such as tweet time, name, followers, accounts followed or profile picture. The dataset consisted of 6760 labeled English and 4800 Spanish users, each with 100 tweets.

This paper aims to research current state of the art approaches for bot detection and implement deep learning-based models. Moreover, we try to improve the existing solutions of the English dataset.

Furthermore, the paper aims at answering the following research questions:

1. Can deep learning-based approaches compete with classical machine learning methods on the limited dataset which was provided?
2. How would increasing the available data with data augmentation influence the performance of the models?
3. What is the effect of using pre-trained models and language representations on the results?

[1] https://www.bloomberg.com/news/articles/2018-05-21/twitter-bots-helped-trump-and-brexit-win-economic-study-says

## 2 Related Literature

There are different bot detection approaches. One could use the Socialbakers rule set or the Camisani-Calzolari rule set (CC) (Camisani-Calzolari, 2012), which assigns an account human and bot scores. Based on the two scores, the classification of an account can be given. Unsupervised methods learn patterns from untagged data, such as by clustering accounts. Therefore, they are effective at identifying coordinated and synchronized accounts (Cresci, 2020). Moreover, adversarial methods are also examined where researchers try to predict the evolution of bots before the evolution actually takes place. As this work focuses on supervised methods below we will discuss this type of approach in more detail.

**Supervised Methods** During our literature review, most papers we discovered were using supervised methods. Similarly, when Cresci (2020) reviewed 236 bot detectors published since 2010, they also found the majority of detectors to be based on supervised methods. Most papers cited in this paper use CML methods (which require features to be predefined) but deep learning methods (which do not require a predefined set of features (Yan et al., 2015)) are also researched (see Table 1). An example of CML model is BotoMeter which calculates a score of a given account by extracting more than 1000 features and using a random forest. Kudugunta and Ferrara (2018) developed deep learning models by creating LSTM models (Hochreiter and Schmidhuber, 1997). Moreover, the researchers also created Contextual LSTM models, which were trained on the tweet's embedding along with user metadata. The LSTM model trained on tweets only, reached 95% accuracy, whereas the best contextual LSTM network achieved about 96%. The top 3 submissions of the PAN 2019 bots and gender profile task were also based on supervised approaches, moreover based on CML methods using feature extraction such as tweet length, number of URLs/mentions etc. Johansson (2019) combined a Logistic Regression (LR) with a Random Forest (RF), Fernquist (2019) trained a CatBoost classifier while Bacciu et al. (2019) used the output of a Support Vector Machine (SVM) and AdaBoost to train a Soft-Voting classifier. These submissions reached an accuracy of more than 0.94 and these solutions serve as a banchmark of this work.

| **Researcher(s)** | **Type** | **Algorithm(s)** |
|---|---|---|
| *Yang et al.Yang et al. (2019)* | Supervised | RF |
| *Pozzana et al. Pozzana and Ferrara (2020)* | Supervised | RF, DT, ET, AB |
| *Varol et al. Varol et al. (2017)* | Supervised | RF, AB, LR, DT |
| *Beskow et al. Beskow and Carley (2018)* | Supervised | NB, LR, SVM, DT, RF |
| *Lee et al. Lee et al. (2011)* | Supervised | 30 classifiers, best RF |
| *Almaatouq et al. Almaatouq et al. (2016)* | Supervised | ZeroR, BN, NB, LR, DT, RF |
| *Chu et al. Chu et al. (2012)* | Supervised | RF |
| *Kudugunta et al Kudugunta and Ferrara (2018)* | Supervised | LR, AB, LSTM |

Table 1: Summary of the cited supervised based papers.

# 3 Data and Methods

In this section different data preprocessing steps and modelling approaches will be introduced for detecting Twitter bots.

## 3.1 Data Preparation

The dataset of the PAN 2019 Bots and Gender Profiling task is split into three sets, namely training (2800 accounts), validation (1240 accounts) and test (2640 accounts) sets. Each account has 100 tweets and an account can belong to a human or a bot. In addition, we worked with the assumption that tweets written by bot accounts shall be classified as bot tweets. Moreover, all data sets are balanced making them ideal for machine learning algorithms.

The bot detectors which will be introduced shortly use GloVE (Pennington et al., 2014) (trained on tweets[2]) as a pre-trained word embedding. During the data preparation phase an important aspect is to preprocess the data in a similar manner as the authors of the word embedding. This would allow the predictive model to learn the semantic and syntatic meaning of words, which could also boost the detectors' performance.

During the data cleaning phase HTML tags, accented characters were removed. Mentions, hashtags, URLs, numbers, repeated characters and emojis were replaced by a tag, contractions were fixed, and special characters were surrounded by spaces. Moreover, the maximum tweet length was limited to 40 tokens in order to reduce the dimensionality of the corpus. Additional preprocessing steps had also been experimented such as using named entity recognition to replace words based on their entity, removing punctuations, stop words and lemmatizing the corpus. Unfortunately, these steps did not improve the accuracy; hence, they were omitted from the data preprocessing steps. After cleaning the data, the cleaned text is fed into the word embedding which creates the embedded representation of the tweets.

## 3.2 Modeling Approaches

In this section 17 models will be introduced, which will be evaluated and discussed. As Tweets are posted in a chronological order we have decided to use Recurrent neural networks (RNNs) as they are well suited for sequence classification, thus the core of seven models are based on LSTM networks, four are based on BERT models (Devlin et al., 2018), one a combination of the two. In addition the work by Kovács et al. (2019) was also combined with the models introduced here. The detectors were written in Python and the deep learning models are built using Pytorch.

[2] https://nlp.stanford.edu/data/glove.twitter.27B.zip

### 3.3 Tweet Classification

**LSTM Tweet Classifier** The aim of this model is to classify whether a tweet was written by a human or by a bot and based on the individual predictions new models can be trained which can classify the accounts. From here on, this model will be referenced as Tweet Classifier. After preprocessing the data, the sequences of the token indexes are fed into the GloVE embedding, which creates the embedding representation of the tokens. Next, an LSTM model learns the patterns of the tweets and finally, two fully connected layers output the prediction. The model outputs the classification, the probability of a tweet being bot and the hidden states of the last LSTM layer which may hold a mix of content and non-content features. These outputs will be used by models which classify the individual accounts.

**Fine-tuning BERT** BERT has also been fine-tuned to classify tweets as bots or humans. We have decided to use the BertForSequenceClassification model from the hugging face library. The preprocessing steps described earlier are irrelevant as BERT requires special preprocessing which can be found in the paper by Devlin et al. (2018).

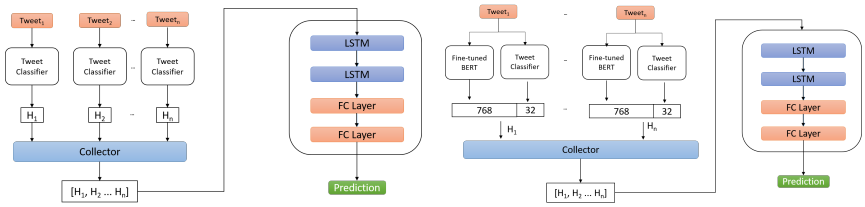
### 3.4 Account Classification

**Majority Vote** In order to classify an account one could simply feed the individual tweets to the Tweet Classifier, collect the predictions and, based on a majority vote, classify the account.

**Probability Based Prediction** A similar approach to the majority vote is to use tweet probabilities. In this scenario, the account's probability is calculated by averaging the sum of each of its tweets' probability. If the account probability is larger than 0.5, the account is considered a bot, otherwise a human.

**Combined Model** The output of the Tweet Classifier can also be the input of an account classifier model. For this approach, each tweet of an account is first fed into the Tweet Classifier model and the latent representation which holds information of the tweet is used as inputs for a second model to classify accounts. The individual hidden states are first collected, then used as inputs for the second model. In this model configuration, the Tweet classifier learns the features and most important characteristics of tweets corresponding to their origin (man-made or bot-generated). The hidden states represent this information; therefore the Tweet Classifier serves as an encoder where the model's output is disregarded.

The model consists of two LSTM layers and a fully connected layer which outputs the prediction for a given account (see Figure 1a). Moreover, during the implementation, 5-fold cross validation was used, where 5 independent models were trained on different partitions of the training set. The prediction of the models is based on the majority vote of the 5 predictions or based on the average of the 5 probabilities, creating an ensembled configuration.



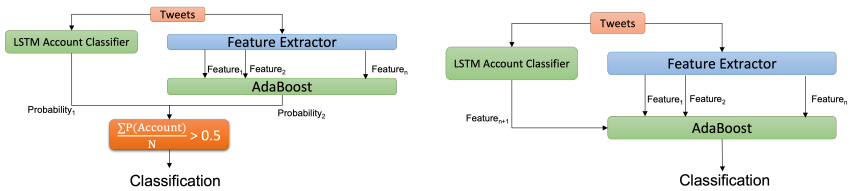

(a) An example of a prediction (b) Hidden State Concatenation

Fig. 1: Model Architectures

**Fine-tuning BERT** The same steps had been tried as described for the LSTM models that is, majority vote, probability-based account prediction and representing each tweet as the last hidden state of BERT and feeding them to the same combined LSTM model. Moreover, the LSTM based account classifier was also trained on the hidden states of a BERT model which was not fine-tuned.

The hidden states of BERT and the LSTM Tweet Classifier can also be combined. As can be seen on Figure 1b the information of the BERT model are concatenated and are fed into the LSTM Account Classifier. This approach allows the account classifier to learn from both the BERT and LSTM models which may supplement each other.

**Combining Deep Learning and CML Methods** Kovács et al. (2019) in their work extracted more than 160 features from the PAN 2019 Bots Dataset and trained an Adaboost model to classify individual accounts. They reached an accuracy of 0.89. In this work we extended this model by implementing a late fusion method which combines the classification and probabilities of the Adaboost model with the Account classifier model. In Addition the Adaboost model was also trained by adding the classification/probabilities of the LSTM Account classifier as an extra feature. The architecture can be seen on Figure 2.




(a) Combining outputs. (b) Account Classifier as extra feature.

Fig. 2: Model Architectures

| Model | LR | Hidden Size | Layers | Batch Size | FC Size | LSTM Dropout | FC Dropout | L2 |
|---|---|---|---|---|---|---|---|---|
| *Tweet Classifier* | 0.00100 | 32 | 1 | 64 | 128 | 0.400 | 0.100 | 0.1 |
| *BERT* | 0.00100 | 768 | - | 64 | - | - | - | - |

Table 2: Best hyper-parameters for tweet classification.

### 3.5 Data Augmentation

As the training set has fewer than 3k accounts, training the LSTM based account classifier network is quite challenging. In order to overcome this issue data augmentation was used on the hidden states of the accounts. Let A be the set of $\{T1, T2...T_n\}$ representing an account with n tweets and the set of the hidden states $H = \{H1, H2...H_n\}$. We define an augmentation function $f(H, l)$, where $l \leq n$, which returns all n-l consecutive subsets of $H$ of length $l$. As Tweets form a chronological sequence, subsets of tweets of an account should also form a valid account. This augmentation method significantly increased the availability of the training data, which should also increase the accuracy of the model.

### 3.6 Hyper-parameter Selection

For hyper-parameter optimization Ray-Tune (Liaw et al., 2018) was used using Tree-structured Parzen Estimator Approach (TPE) as the search algorithm and Asynchronous SHA (ASHA) to early terminate bad configurations. In order to find the best configuration 30 trials were executed. Table 2 and 3 shows the hyper-parameters of each model. It is also important to highlight that an additional preprocessing step lies before the execution of the LSTM account classifier which serves to normalize the hidden states of the tweets.

| Model | LR | Hidden Size | Layers | Batch Size | FC Size | LSTM Dropout | FC Dropout | L2 |
|---|---|---|---|---|---|---|---|---|
| *Account Classifier (LSTM, LSTM)* | 0.00800 | 32 | 2 | 64 | 128 | 0.499 | 0.900 | 9.1e-08 |
| *Account Classifier + Aug. (LSTM, LSTM)* | 0.00890 | 64 | 2 | 64 | 32 | 0.466 | 0.055 | 1.1e-06 |
| *Account Classifier (BERT, LSTM)* | 0.00040 | 128 | 2 | 64 | 512 | 0.088 | 0.100 | 6.7e-06 |
| *Account Classifier + Aug.* (BERT, LSTM) | 0.00329 | 512 | 2 | 64 | 32 | 0.198 | 0.500 | 2.8e-06 |
| *Account Classifier* (BERT w/o fine-tuning, LSTM) | 0.00017 | 256 | 1 | 32 | 32 | 0.402 | 0.176 | 3.9e-07 |
| *Account Classifier* (BERT + LSTM, LSTM) | 0.00171 | 128 | 2 | 64 | 64 | 0.4661 | 0.1055 | 9.9e-06 |

Table 3: Best hyper-parameters for account classification.

| Model | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|
| *LSTM Tweet Classifier* | 0.753 | 0.753 | 0.753 | 0.754 |
| ***BERT Tweet Classifier*** | **0.828** | **0.800** | **0.937** | **0.700** |

Table 4: Tweet classification results.

## 4 Results and Discussion

All 17 models designed in this work were evaluated on the test set and the results can be seen on Table 4, 5 and 6. Moreover, submissions to the competition were evaluated based on the accuracy of the models, therefore we will also report based on this measure.

The fine-tuned BERT model classified the origin of tweets with the highest accuracy, yielding an accuracy of 0.828. Following, the LSTM model reached an accuracy of 0.753 (see Table 4).

On the other hand, the single LSTM model performed much better than the fine-tuned BERT model during account classification using majority and probability based account predictions, beating the best BERT model by almost 3% (see Table 5).

As can be seen on Table 6, from the combined deep learning models the combined LSTM model yielded the highest accuracy of 0.892. On the other hand, combining the LSTM Account Classifier with the Adaboost model surpassed the best deep learning model with an accuracy of 0.9.

Augmenting the input data on average resulted in a 1% decrease in the accuracy (see Table 6). Although, data augmentation typically increases the accuracy, there can be several reasons why the models did not perform better. One reason is that during the hyper-parameter optimization, the search space was ill-defined and the optimization algorithm did not explore configurations that reach a high performance. On the other hand, maybe the augmented data was too complex and the models could not learn all the patterns. This can explain why the training accuracy was not improving while the validation accuracy was during training.

An interesting point is that although the fine-tuned BERT model predicts individual tweets better (+6%) than the LSTM based tweet classifier, the majority and probability-based account predictions are worse than the majority or probability predictions of the LSTM tweet classifier. This can be because of two

| Model | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|
| *Tweet Classifier - Majority Vote* | 0.873 | 0.864 | 0.930 | 0.806 |
| ***Tweet Classifier - Probability based pred.*** | **0.878** | **0.870** | **0.936** | **0.811** |
| *BERT - Majority Vote* | 0.849 | 0.826 | 0.975 | 0.717 |
| *BERT - Probability based pred.* | 0.852 | 0.830 | 0.976 | 0.721 |

Table 5: Account classification results based on single models.

| Model | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|
| *Account Classifier - Majority Vote (LSTM, LSTM)* | 0.891 | 0.890 | 0.895 | 0.886 |
| *Account Classifier - Probability based pred. (LSTM, LSTM)* | 0.892 | 0.893 | 0.881 | 0.906 |
| *Account Classifier + Augmentation (LSTM, LSTM)* | 0.880 | 0.880 | 0.878 | 0.884 |
| *Account Classifier - Majority Vote (BERT, LSTM)* | 0.836 | 0.828 | 0.872 | 0.787 |
| *Account Classifier - Probability based pred. (BERT, LSTM)* | 0.838 | 0.830 | 0.874 | 0.790 |
| *Account Classifier + Augmentation (BERT, LSTM)* | 0.835 | 0.821 | 0.903 | 0.753 |
| *Account Classifier (BERT w/o Fine-tuning, LSTM)* | 0.806 | 0.793 | 0.853 | 0.740 |
| *Account Classifier (BERT + LSTM, LSTM)* | 0.838 | 0.820 | 0.916 | 0.743 |
| *Account Classifier + Adaboost* | 0.878 | 0.880 | 0.890 | 0.870 |
| *Account Classifier prob. Feature + Adaboost* | 0.891 | 0.890 | 0.890 | 0.890 |
| ***Account Classifier pred. Feature + Adaboost*** | **0.900** | **0.900** | **0.900** | **0.900** |

Table 6: Account classification results based on multiple models.

factors. First, the BERT model has a low recall which means many human accounts are classified as bots; second, the distribution of the incorrect tweets are spanned across more accounts than the LSTM based classifier, which can have a significant impact during the account classification. Moreover, the LSTM account classifier model performed poorer with the BERT hidden states. Again, this could be because of the incorrect hyper parameter configurations of the LSTM account classifier, or the hidden representation of the tweets are too similar; that is, the LSTM model cannot differentiate humans and bots.

It is also important to highlight that the LSTM based tweet classifier classifies accounts just 1% lower than the highest performing model introduced in this work. However, it can be trained in a fraction of the training time of a complex model such as fine-tuning BERT or the combined LSTM models.

### 4.1 Comparison of the best results of the PAN 2019 Author Profiling task

The best results and the majority of the submissions solved the task by CML algorithms and used feature extraction. On the other hand the work presented here was based on deep learning models. Therefore, it is hard to compare the results. The top 3 submissions achieved more than 0.94. In summary it can be stated that the work presented is not superior to the benchmarked solutions and further improvements should be made to improve the performance of deep learning methods. Having said that, some submissions to the competition were

based on deep learning approaches. Onose et al. (2019) created Hierarchical Attention Networks and reached an accuracy of 0.89, while some submissions were based on LSTM models (Rangel and Rosso, 2019) but the accuracy was 0.87. As we can see the models introduced in this work outperformed the deep learning approaches that were reviewed and cited in this paper. Nevertheless, the work presented in this paper should be further improved and more research shall be made in this research area.

### 4.2 Reflection on the research questions

This paper aims at addressing three research questions, which are described in the problem definition. Referring to RQ1 it can be argued that deep learning based methods, especially LSTM based models, achieved good but not superior results compared to classical approaches.

For RQ2, data augmentation did not improve the accuracy because of the already mentioned reasons, such as by incorrectly specifying the hyper-parameter search space or incorrect implementation. Nevertheless, data augmentation in the NLP domain is challenging and an area that needs to be researched.

Regarding RQ3, it is clear that pre-trained models and language representations do improve the results. The fine-tuned BERT model beat the LSTM based tweet classification model by almost 7%.

## 5 Conclusion and Future work

The bot detectors introduced here successfully solve bot detection with an accuracy of 0.90. Although it can be argued, the presented deep learning models are not better than traditional classical machine learning methods. Having said that, the data set was quite limited in size and on large scale data the outcome may have been different. In addition, to the best of our knowledge, the presented architecture of the combining LSTM/BERT models with another LSTM model for classifying accounts as well as combining the above with an Adaboost model have not been researched before. Therefore, this work can also serve as a baseline for future improvements for deep learning and hybrid (deep learning and classical machine learning) based approaches.

### 5.1 Further development

The work presented here can be further improved in several areas. In the future, text augmentation should be applied, such as the approach by Wei and Zou (2019), which combines synonym replacement, random insertion, random swap and random deletion. Architectural changes could also be made to the models such as by using different loss functions, optimizers, replacing LSTMs by GRUs or by using different word embeddings.

Another approach could be to improve the hidden representation of the models by using Siamese neural networks (Chopra et al., 2005). These networks are trained to predict whether two input samples are the same or not by calculating the similarity of the inputs. These networks have shown great results in computer vision in the area of face verification, but they could also be used to improve the bot detection models of this work. If a Siamese network were to be used, which is trained on tweets of humans and bots, the model would be forced to learn the characteristics of bots and humans in order to differentiate them. Therefore, the hidden representations of the tweets would also be more similar if they came from the same account or same type of accounts such as a bot or a human. These hidden states could then be used by the account classification model, which was introduced earlier.

## References

Almaatouq, A., Shmueli, E., Nouh, M., Alabdulkareem, A., Singh, V.K., Alsaleh, M., Alarifi, A., Alfaris, A., et al.: If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. International Journal of Information Security 15(5), 475–491 (2016)

Bacciu, A., La Morgia, M., Mei, A., Nemmi, E.N., Neri, V., Stefa, J.: Bot and gender detection of twitter accounts using distortion and lsa. In: CLEF (Working Notes) (2019)

Beskow, D.M., Carley, K.M.: Bot-hunter: a tiered approach to detecting & characterizing automated activity on twitter. In: Conference paper. SBP-BRiMS: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. vol. 3, p. 3 (2018)

Bessi, A., Ferrara, E.: Social bots distort the 2016 us presidential election online discussion. First Monday 21(11-7) (2016)

Camisani-Calzolari, M.: Analysis of twitter followers of the us presidential election candidates: Barack obama and mitt romney. Online). http://digitalevaluations. com (2012)

Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 539–546. IEEE (2005)

Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Transactions on dependable and secure computing 9(6), 811–824 (2012)

Cresci, S.: A decade of social bot detection. Communications of the ACM 63(10), 72–83 (2020)

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: The paradigm-shift of social spambots: Evidence, theories, and tools for the arms

race. In: Proceedings of the 26th international conference on world wide web companion. pp. 963–972 (2017)
Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
Fernquist, J.: A four feature types approach for detecting bot and gender of twitter users. In: CLEF (Working Notes) (2019)
Ferrara, E.: Disinformation and social bot operations in the run up to the 2017 french presidential election. arXiv preprint arXiv:1707.00086 (2017)
Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
Howard, P.N., Woolley, S., Calo, R.: Algorithms, bots, and political communication in the us 2016 election: The challenge of automated political communication for election law and administration. Journal of information technology & politics 15(2), 81–93 (2018)
Johansson, F.: Supervised classification of twitter accounts based on textual content of tweets. In: CLEF (Working Notes) (2019)
Kovács, G., Balogh, V., Mehta, P., Shridhar, K., Alonso, P., Liwicki, M.: Author profiling using semantic and syntactic features. In: CLEF (Working Notes) (2019)
Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Information Sciences 467, 312–322 (2018)
Lee, K., Eoff, B., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on twitter. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 5 (2011)
Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 (2018)
Onose, C., Nedelcu, C.M., Cercel, D.C., Trausan-Matu, S.: A hierarchical attention network for bots and gender profiling. In: CLEF (Working Notes) (2019)
Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162
Pozzana, I., Ferrara, E.: Measuring bot and human behavioral dynamics. Frontiers in Physics 8, 125 (2020)
Rangel, F., Rosso, P.: Overview of the 7th author profiling task at pan 2019: bots and gender profiling in twitter. In: Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop (2019)
Varol, O., Ferrara, E., Davis, C., Menczer, F., Flammini, A.: Online human-bot interactions: Detection, estimation, and characterization. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 11 (2017)
Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
Yan, L.C., Yoshua, B., Geoffrey, H.: Deep learning. nature 521(7553), 436–444 (2015)

Yang, K.C., Varol, O., Davis, C.A., Ferrara, E., Flammini, A., Menczer, F.: Arming the public with artificial intelligence to counter social bots. Human Behavior and Emerging Technologies 1(1), 48–61 (2019)