symmetry

MDPI

Article

Social Bots Detection via Fusing BERT and Graph Convolutional Networks

Qinglang Guo [1,2], Haiyong Xie [1,3,4,*], Yangyang Li [2], Wen Ma [5] and Chao Zhang [2]

[1] School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, China; gql1993@mail.ustc.edu.cn

[2] National Engineering Research Center for Public Safety Risk Perception and Control by Big Data (RPP), China Academy of Electronics and Information Technology, Beijing 100041, China; liyangyang@cetc.com.cn (Y.L.); zhangchao26@cetc.com.cn (C.Z.)

[3] Key Laboratory of Cyberculture Content Cognition and Detection, Ministry of Culture and Tourism, University of Science and Technology of China, 96 Jinzhai Road, Hefei 237009, China

[4] Advanced Innovation Center for Human Brain Protection, Capital Medical University, Beijing 100160, China

[5] School of Software, Xinjiang University, Urumqi 830049, China; mawen@stu.xju.edu.cn

[*] Correspondence: hxie@ustc.edu.cn

Abstract: The online social media ecosystem is becoming more and more confused because of more and more fake information and the social media of malicious users' fake content; at the same time, unspeakable pain has been brought to mankind. Social robot detection uses supervised classification based on artificial feature extraction. However, user privacy is also involved in using these methods, and the hidden feature information is also ignored, such as semi-supervised algorithms with low utilization rates and graph features. In this work, we symmetrically combine BERT and GCN (Graph Convolutional Network, GCN) and propose a novel model that combines large scale pretraining and transductive learning for social robot detection, BGSRD. BGSRD constructs a heterogeneous graph over the dataset and represents Twitter as nodes using BERT representations. Corpus learning via text graph convolution network is a single text graph, which is mainly built for corpus-based on word co-occurrence and document word relationship. BERT and GCN modules can be jointly trained in BGSRD to achieve the best of merit, training data and unlabeled test data can spread label influence through graph convolution and can be carried out in the large-scale pre-training of massive raw data and the transduction learning of joint learning representation. The experiment shows that a better performance can also be achieved by BGSRD on a wide range of social robot detection datasets.

Keywords: social bots detect; GNN; GCN; pre-training; BERT

check for updates

Citation: Guo, Q.; Xie, H.; Li, Y.; Ma, W.; Zhang, C. Social Bots Detection via Fusing BERT and Graph Convolutional Networks. *Symmetry* **2022**, *14*, 30. https://doi.org/10.3390/sym14010030

Academic Editor: Jeng-Shyang Pan

Received: 16 November 2021
Accepted: 20 December 2021
Published: 27 December 2021

1. Introduction

News content that is easier to consume is due to the introduction of social media [1]. The development of social media is a double-edged sword as it also has negative effects, such as bringing us unspeakable pain. Social media is different from traditional media (newspapers, television and radio), and the new news trend of "fake news" is also welcomed by social media, which quickly spreads some news with intentionally misleading information. The malicious activities of attackers, spammers and fraudsters are also due to the typical characteristics of the openness and sharing of online social networks. One of the highest security threats in online social networks is social robots, which are more vulnerable to attackers. The interaction between social robots and humans on social media is the imitation of computer software that automatically generates content, and this imitation will also change their behaviour. Creating an illusion is the main goal of these social robots, so the positive influence of social networks on public opinion can be explained in this way [2]; political penetration [3] is triggered and malicious content is also widely spread. These malicious social robots will also have a negative impact on popular social networks, mainly on human users.

At present, social robots on Twitter are facing three main challenges, mainly in the following respects: it is difficult to fully extract features, which is the first challenge of social robots on Twitter because they are characterized by complexity. In order for a social robot to avoid being discovered, it is necessary for it to pretend to be an ordinary user. To describe social robots more accurately, it is necessary to consider their characteristics and various contents. Only extracting the features of social robots from a single angle [4,5] cannot fully describe them, which is the result of many existing types of research. Building a detection model only uses a small number of features, considering the features of social robots [6,7] and studying them from several perspectives, which is the research content of other works. It is difficult to obtain large-scale tags for research datasets from Twitter, which is the second challenge. On Twitter, the lack of large-scale reliable datasets is caused by the relative rarity of social robot detection research; it needs rich, experienced support and takes a lot of time to mark manual proofing. Small-scale datasets [4,7,8] are the basis of most existing studies. Another great challenge of current research is to accurately and effectively scale datasets, which is needed for the detection of social robots on Twitter. When classical detection methods are used to detect social robots on Twitter, their performance is not very good. This is the third challenge. The performance of detection methods has been improved because machine learning detection methods have been used in previous work [4,9], but there is still much work to be done. Therefore, the detection method needs to be further developed for the detection of high-performance social robots based on deep neural networks.

BERT can learn the semantic information of the text in advance on large-scale text, and then fine-tune it on the Twitter dataset to learn the distribution characteristics of the Twitter data, so as to overcome the pain point of missing the large-scale dataset. However, GCN has a good ability to capture and learn the propagation and co-occurrence relationships of Twitter and can learn the complex features of Twitter robots in multiple dimensions.

As shown in Figure 1, a new model—BGSRD—is proposed in this work, and the detection of social robots is, through its symmetry, a combination of BERT and GCN. Large-scale pre-training and transduction learning of social robot detection is carried out by this model, combining the following advantages. BGSRD constructs a heterogeneous graph of the corpus, which uses pre-trained BERT embedded nodes as word or document nodes to classify, initialize and use the GCN of robot classification. The model that can take advantage of the two worlds is obtained by jointly training BERT and GCN modules: (1) massive raw data can be pre-trained on a large scale; (2) The label's influence through the edge of the graph can be carried out through the transduction learning of the representation of learning training data and unlabeled test data. The above three challenges can be overcome by combining the pre-training model and a graph neural network. The successful combination of large-scale pre-training and the power of the graph network is the BGSRD model. At the same time, a better performance is obtained, especially on a wide range of social robot detection datasets.

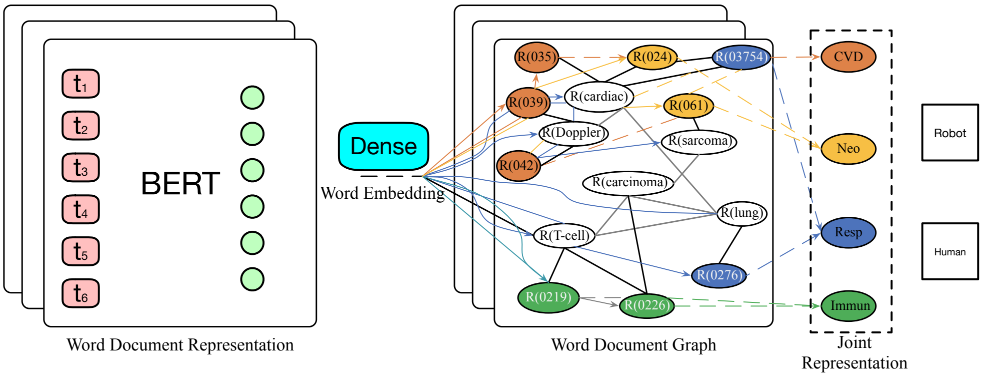

Figure 1. High-level illustration of BGSRD.

This major contribution includes:

- We combine pre-trained language model BERT and Graph Convolutional Networks to detect social bots;
- We can fuse semantic information by applying BERT multi-head attention, and a better-integrated representation can be generated by each text;
- We adopt a novel graph neural network method to detect social robots. This is research on embedding a heterogeneous graph and graph neural network to learn words and documents through the whole corpus modelling.

2. Related Works

With the spread of robot accounts on social networks, there are more and more studies on social robot account detection. With the development of related research, related methods can be divided into the following categories [10]: crowdsourced social machine account detection platform, detection technology via traditional machine learning, detection technology over deep learning, detection technology using social network graphs, and so forth.

2.1. Crowdsourcing Social Machine Account Detection Platform

Reference [11] proposes a crowdsourcing social machine account detection platform. It is considered that machine account detection is a relatively simple technology for human beings, so an online Turing detection platform is created. By employing a large number of workers and experts to test the account data in Facebook and Renren, the same account data are provided to multiple workers, and the opinions of the majority are taken as the final judgment.

However, its disadvantages are also very obvious. It would be better to do this in the early days of social networking, but the cost is almost unrealistic for established social networking platforms. The number of users of various mainstream social platforms has experienced explosive growth in the past few years. For example, the number of monthly active users of Twitter reached 336 million in 2019, which was an increase of 2.5 times compared with 2012 [12]. Compared with this high cost and inefficient service, it is not applicable. Due to the massive number of users and data every day, such a scheme can only stay in the process of theory and experiment, but cannot really be put into practical application.

2.2. Detection Technology Based on Machine Learning

The most common technology for the detection of machine accounts is based on machine learning, and is the mainstream detection technology at present. Taking this problem as a binary classification problem is the essence of machine account detection technology based on machine learning. After the required features are extracted from the account, the classification algorithm is used to analyze the data, and the detection model is trained. Then, the model is used to analyze the data of the account that needs to be classified and classify it.

2.3. Detection Technology Based on Deep Learning

With the development of deep learning, more and more studies have been applying it to machine account detection. Deep learning is a branch of machine learning. Deep learning takes artificial neural networks as the basic framework within which to conduct data representation learning [13]. Recently, with the rapid development of deep learning, more and more studies have also been applied to machine account detection. One branch of machine learning is deep learning. Deep learning learns data representation based on artificial neural networks [13]. Unlike with traditional machine learning, an in-depth study of the data needs more data and time to train the model; deep learning, at the same time, can use unsupervised, or characteristics of, semi-supervised learning and use a hierarchical feature extraction algorithm to replace the artificial nerual network [14] and obtain the characteristics, which can save time and discover some hidden features.

LSTM (Long Short-term Memory) is a kind of temporal cyclic neural network, first published in 1997 [15]. It is especially designed to solve the general cyclic neural network RNN (recurrent neural network, RNN). Suitable for processing and predicting events with long intervals and delays in time series, they are now often constructed as part of large deep neural networks. Researchers of machine account detection also use LSTM in correlation experiments and projects [16,17]. CNN (convolutional neural network) and LSTM networks have been used in machine account detection [16]. The CNN network is used to extract the characteristics and relations of the Twitter text content. The second layer regards the Twitter metadata as time information and uses the time information as the input to LSTM to extract the time characteristics of users' social activities. Finally, in the fusion feature layer, the previous content features and metadata features are fused to detect the machine account, and the final detection results are obtained.

Reference [17], using Twitter content and metadata, detected machine accounts at the level of tweets, extracted contextual features from user metadata, and provided them as auxiliary input to the LSTM network that processed the tweet's text. The model only needs one tweet to determine whether it is a machine account. Reference [18] used the BiLSTM (Bi-directional Long Short-Term Memory) algorithm to detect machine accounts. BiLSTM is an algorithm using bidirectional LSTM, and the two LSTMs are in opposite directions. Together they form the BiLSTM network. The model uses the context of tweets as input, and enters the BiLSTM network after word embedding. Finally, the outputs of forward LSTM and backward LSTM are stitched together, and then the normalized function is used for classification so as to obtain the required detection results. This model only uses the content of tweets as input, and does not use other features. The advantage of this method is that it saves a lot of working time of feature extraction, does not need manual features and prior knowledge, can improve work efficiency, and is more convenient to deploy in the scene of batch detection. Reference [19] proposed a two-stage, graph-based machine account detection system. The system utilizes supervised learning and unsupervised learning. Reference [20] uses incremental learning to process data in real-time. Although the convergence time of the model is longer, the final model produces a superior classification performance and is suitable for stream-based detection systems.

Similarly, the detection technology based on deep learning also has its disadvantages. When the dataset is not large enough, the effect of the neural network is often poor and the phenomenon of over-fitting easily occurs.

2.4. Detection Technology Based on Social Graph

The detection technology based on a social graph is mainly based on the social network graph formed between users in the social network. The social network graph can be used to understand and analyze the relationships between users on the social network platform. Therefore, the detection technology based on a social graph focuses on the relationships between users. After all, in a social network, no accounts exist in isolation, and they are all connected to each other. The social graph of normal users and machine accounts is often very different. For example, a large part of normal users' good friends come from real friends, who follow each other and interact often. Machine accounts, on the other hand, do not have such features. They will have fewer mutual friends, which is obvious in the social graph. They will also have fewer comments and likes, and most of them will tweet or retweet to expand their influence. There will also be a difference in the percentage of friends between normal users and computer accounts. Therefore, the structure of the social graph of normal users is significantly different from that of machine accounts, and the detection scheme based on the social graph uses this difference, together with the network characteristics of users, to identify and detect machine accounts.

SybilRank [21] represents an example of this framework: an opposing party can control multiple social machine accounts (often referred to as Sybils in this case) to impersonate different identities and launch attacks or infiltrations. Proposed strategies for detecting Sybil accounts often rely on examining the structure of the social graph. For example, SybilRank assumes that Sybil accounts only show a small number of links to legitimate users, rather than primarily to other Sybil accounts because they require a large number of social connections to show a trustworthy status. This feature can be used to identify dense, interconnected social machine accounts. In addition, research such as Sybilwalk [22], Gang [23], SybilScar [24], and Sybilfuse [25] are all machine account detection methods based on social interaction correlation diagrams.

3. Methods

The proposed BGSRD model uses the BERT model to initialize the representation of document nodes in the text graph. These are the representations used as input for GCN. Iterative updating based on the graph structure using GCN means that the social robot represents the posted text, and the final representation of the document node is its output, which needs to be sent to the softmax classifier when making predictions. In this way, we can make use of the complementary advantages of the pre-training model and graph models. (Replicating the experiment code is available at https://github.com/shanmon110/BGSRD (accessed on 22 November 2021)).

3.1. Textual Representation via BERT

The essence of BERT is to provide a better feature representation for word learning by running a self-supervised learning method on the basis of the massive corpus. As shown in Figure 2, the generalization ability of the word embedding model is further increased by the BERT model, and the relationship characteristics between character level, word level, sentence level and even sentences are also fully described. MLM (Mask Language Model) is used for multi-task training objectives, similar to the cloze test; although all position information is still seen, the words that need to be predicted have been replaced by special symbols, which can be bidirectional encoding. BERT uses Transformer as an encoder to achieve context correlation, and Transformer instead of BiLSTM as an encoder can have deeper layers and better parallelism. In addition, linear Transformer is more immune to the influence of mask markers than LSTM. All you need to do is reduce the weight of mask markers through self-attention, while LSTM is similar to the black-box model and it is difficult to determine its internal processing mode for mask markers. BERT adopted the NSP (Next Sentence Prediction, NSP) multi-task training goal to learn Sentence/Sentence pair relationship representation, and sentence level negative sampling. First, given a sentence, the model identifies whether the next sentence is a positive example (correct word), conducts random sampling of a negative example (random sampling word), and includes sentence-level dichotomies (that is, judge whether the sentence is the next sentence of the current sentence or noise), similar to word2vec word-level negative sampling.

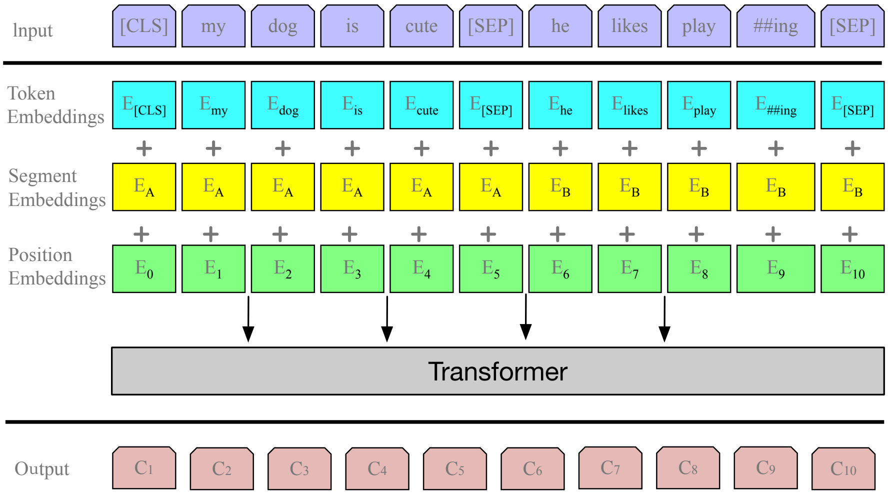
Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}
Transformer
Output
C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10}

Figure 2. BERT input–output representation [26]. We use BERT to generate word embeddings.

3.2. TextGCN

In order to model the global word co-occurrence more clearly, a large heterogeneous text graph containing word nodes and document nodes is constructed, as shown in Figure 3, so that the graph convolution can be easily adapted. The number of documents (corpus size) plus the number of unique words (vocabulary size) in the corpus is the number of nodes in the text graph $|V|$. For the input of Text GCN, a one-hot vector is every word or document, and the identity matrix simply sets the feature matrix $X = I$. The edge between nodes is constructed by word occurrence in documents (document–word edge) and word co-occurrence in the whole corpus (word–word edge). The word frequency-inverse document frequency (TF-IDF) of a word in a document is the weight of the edge between a document node and a word node, the frequency of its occurrence in a document is the word frequency, and the reciprocal of the logarithmic proportion of the number of documents containing the word is the inverse document frequency. It is better to use TF-IDF weight than just using word frequency. For all documents in the corpus, in order to make use of the global word co-occurrence information, collecting co-occurrence statistics mainly uses a sliding window of fixed size. We mainly calculate the weight between two-word nodes by PPMI (point-wise mutual information), a popular word association measure. In our preliminary experiment, PMI can produce better results, especially when word co-occurrence counting is used. Node i and node j formally define the weight as follows:

$$A_{ij} \begin{cases} PPMI(i,j) = i,j \; are \; words, i,j \; are \; words \; and \; i \neq j \\ TF - IDF_{ij}, i \; is \; document, j \; is \; word \\ 1, \; i = j \\ 0, \; otherwise. \end{cases} \tag{1}$$

The PPMI value of a word pair i, j is computed as:

$$PPMI(i,j) = \log \frac{p(i,j)}{p(i), p(j)} \tag{2}$$

$$p(i,j) = \frac{\#W(i,j)}{\#W} \tag{3}$$

$$p(i) = \frac{\#W(i)}{\#W}, \tag{4}$$

where $\#W(i)$ is the number of sliding windows in the corpus containing the word i. $\#W(i,j)$ is the number of sliding windows containing both the words i and j, where $\#W$ is the total number of sliding windows in the corpus. A positive PMI value means high semantic correlation of words in the corpus. Negative PMI values indicate little or few. There is no semantic correlation in the corpus. So, just add an edge between the word pairs using a positive PMI value.

After creating the text graph, feed the graph to a simple two-tier GCN. In Reference [27], the embedding of the second layer node (word/document) is the label and is set and sent to the softmax classifier.

$$Z = softmax(\hat{A}ReLU(\hat{A}XW_0)W_1), \tag{5}$$

where $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the same as in Equation (1), and $softmax(x_i) = \frac{1}{Z}exp(x_i)$ with $Z = \Sigma_i exp(x_i)$. The loss function is defined as the cross-entropy error over all labeled documents:

$$L = -\Sigma_{d \in Y_D} \Sigma_{f=1}^{F} Y_{df} \ln Z_{df}. \tag{6}$$

The document index set with labels is Y_d, and the output feature is F, which is equal to the number of classes. The label matrix is Y. The weight parameters W_0 and W_1 can be trained by gradient descent. Figure 3 is a schematic diagram of the overall Text GCN model.

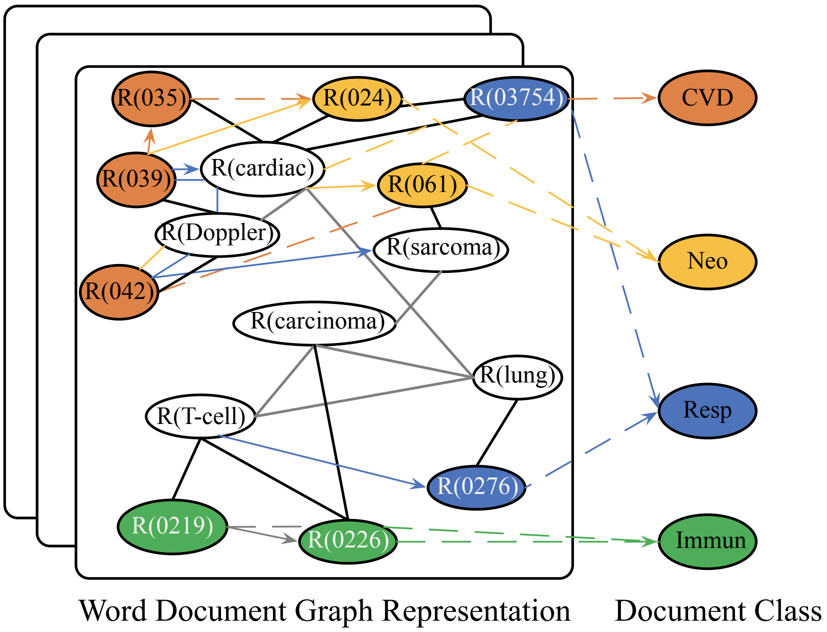

Figure 3. Text GCN is as follows [28]. A document is a node that begins with "O", and a word node is another node. The edges of the document are thick black edges and thin grey edges. The representation (embedding) of x is represented by R(x). Different document classes are represented by different colours (only four sample classes are shown to avoid confusion). CVD: cardiovascular disease; Neo: tumours; corresponding: respiratory diseases; Immun: immune diseases.

What is passed between nodes at most two steps away is the message that two layers of GCN can allow. Therefore, the information exchange between document pairs is allowed between two layers of GCN, and there is no direct connection between document and document edge. The performance of single-layer GCN is better. In our preliminary experiment, especially compared with two-layer GCN, it is concluded that more layers do not improve the performance. The results are similar to those in [27,29].

3.3. Interpolating BERT and GCN Predictions

In fact, the faster convergence and better performance of BGSRD are the reasons why BERT is directly optimized by using embedded auxiliary classifiers. The auxiliary classifier is mainly built by embedding the document (represented by X) specifically, directly feeding it to the dense layer with softmax activation:

$$Z_{GCN} = softmax(g(X, A)) \tag{7}$$

$$Z_{BERT} = softmax(WX). \tag{8}$$

The GCN model is represented by g. The joint optimization of BERT and GCN parameters is carried out by using the cross entropy loss at the nodes of the markup document. The linear interpolation of the prediction from BGSRD and the prediction from BERT is the final training goal, which is given by the following formula:

$$Z = \lambda Z_{GCN} + (1 - \lambda) Z_{BERT}. \tag{9}$$

The trade-off between two targets is controlled by λ. We use $\lambda = 1$ for the complete BGSRD model and $\lambda = 0$ for the BERT module only. The BGSRD model can be better optimized and we can balance the predictions of the other two models.

The explanation for obtaining a better performance can be explained by interpolation in the following: the input of GCN is adjusted and optimized for the target, which ensures that the input of GCN needs to be operated by Z_{BERT}. This is the reason a better performance can be obtained, and it is also beneficial for overcoming the inherent defects, such as gradient disappearance or excessive smoothing by the multi-layer GCN model [29].

4. Experiments

4.1. Datasets

We ran experiments on five widely-used social bot detection benchmarks: cresci-rtbust [30], botometer-feedback [31], gilani [32], cresci-stock-2018 [33,34], midterm [35]. These datasets are in the same format, including crawling time, user profile, description, followers, location, URL, and so forth. We have put these datasets with our code on GitHub (https://github.com/shanmon110/BGSRD (accessed on 22 November 2021)). The existing common datasets are summarized in Table 1. The difference between the number of accounts and the original number is caused by removing invalid accounts from the dataset. Stefano Crescis' research team and Reference [35] have collected many datasets, which are of great help to the study of machine accounts on social networks.

Table 1. A brief summary of the dataset.

Dataset	Year	Machine Account	Normal Users
cresci-rtbust [30]	2019	332	322
botometer-feedback [31]	2019	139	375
gilani [32]	2017	1090	1413
cresci-stock-2018 [33,34]	2019	6907	5992
midterm [35]	2018	41,395	7790

4.2. Baselines

Cresci-rtbust [30]: A new technology that only needs the time stamp of retweets for each analyzed account is used to detect the retweeting social robot, so there is no need to provide a complete user timeline or social graph.

Botometer [31]: A popular robot detection tool was developed by Indiana University. Botometer is based on Random Forest classifiers; given a Twitter account, Botometer extracts over 1000 features relative to the account from data easily provided by the Twitter API, and produces a classification score called a bot score: the higher the score, the greater the likelihood that the account is controlled completely or in part by software.

gilani: Reference [36] mentions three methods with which to conduct experiments on gilani; we will compare these three methods as a baseline. gilani has two main parts: bot and analyser. The bot fetches a trending topic or a popular tweet, disassembles the information in the topic or tweet, and the analyser is used for analysis.

cresci-stock: Reference [33,34] proposed a method for detecting social robots in the financial field. cresci-stock studies tweets related to the stocks of the five main financial markets in the US and bot detection techniques.

midterm [35,37]: Realization of efficient analysis and scalability to process all Twitter's public tweet streams in real time through a framework that uses minimal account metadata.

4.3. Experimental Setup

Document embedding is the output feature of using a [CLS] token. Compared with BERT and RoBERTa, it is the feedforward layer that obtains the final prediction. BGSRD is realized by using BERTbase and two layers of GCN. Learning rate initialization 1×10^{-3} is used for the GCN module, and 1×10^{-3} is used for fine-tuning the BERT module. Our model is realized mainly by using RoBERTa and GAT (Graphic Attention Network) [38]. Learning edge weights is not based on a predefined weight matrix but on the attention mechanism, especially when GAT variants are trained on the same graph as GCN variants. The input length for setting BERT is 18, 128 is the batch size, and 200 is the dimension of the GCN hidden layer. The number of attention heads of GAT is set as 8 and 0.5 is the default value of dropout. The parameter is updated by using the Adam optimizer.

4.4. Results and Analysis

The detection results of the robot can be seen in Tables 2–6. BGSRD technology achieves the best detection performance because BERT with GNN is used for feature extraction. In most evaluation indicators, in fact, BGSRD technology has defeated many other competitors. Extracting information features from our referral time series is one of the expected advantages of supporting GNN. The second-best overall result is obtained through each model. Most of the worst results are obtained by the evaluated technology in terms of accuracy index, which is interesting because there are many legal accounts that are wrongly classified as a robot. From a result comparison, this is different from the previous robot detection results.

Table 2. Bot detection results on the Cresci-rtbust dataset and comparison with a baseline and other techniques [30]. The best and second-best results for each metric are bold and underlined, respectively.

Model	Precision	Recall	F1	Accuracy
Botometer	0.6951	0.3098	0.4286	0.5830
HoloScope	0.2857	0.0049	0.0096	0.4908
Social fingerprinting	0.6562	0.8978	**0.7582**	0.7114
RTbust (handcrafted features)	0.5284	0.7707	0.6270	0.5364
RTbust (PCA)	0.5111	**0.9512**	0.6649	0.5154
BGSRD	**0.8842**	0.5926	0.7096	**0.78**

Table 3. Bot detection results on the botometer-feedback-2019 dataset and comparison with a baseline and other state-of-the-art techniques. The best results for each metric are bold.

Model	Precision	Recall	F1	Accuracy
Botometer-feedback [31]	0.6951	0.3098	0.4286	0.5830
BGSRD	**0.7336**	**0.6651**	**0.6977**	**0.8108**

Table 4. Bot detection results on the gilani dataset and comparison with a baseline and other techniques. The best and second-best results for each metric are bold and underlined, respectively.

Model	Precision	Recall	F1	Accuracy
Light [37]	0.681	0.172	0.274	0.615
D [37]	0.726	0.390	0.508	**0.670**
Botometer [37]	0.687	0.341	0.456	0.644
BGSRD	**0.7621**	**0.5036**	**0.6065**	0.5259

Table 5. Bot detection results on the stock dataset and comparison with a baseline and other techniques. The best and second-best results for each metric are bold and underlined, respectively.

Model	Precision	Recall	F1	Accuracy
Light [37]	0.548	0.285	0.375	0.495
D [37]	**0.714**	**0.960**	**0.819**	0.495
Botometer [37]	0.673	0.927	0.780	**0.719**
Cresci-stock [33,34]	0.5284	0.7707	0.6270	0.5364
BGSRD	0.666	0.6584	0.6622	0.6698

We also observe that the model with the BGSRD set of features performs consistently well overall, outperforming or obtaining similar results to the other models. The excellent performance of the model containing D in the stock dataset is also worth mentioning, where it performs the best. This provides evidence that the compression statistics extracted from the Digital DNA can detect bots that behave coordinately, as happens with stock. Moreover, by combining D with data selection it is possible to build a classifier that can generalise properly in different domains. Alternatively, the model with BGSRD, except for the stock dataset, produces results that outperform those of the other models on some occasions. Besides, it shows the best specificity in all cases and is scalable. BGSRD seems to be more robust against the bots in five datasets, probably because its features cover more aspects other than the user metadata, and BERT is used to study more semantic information. Results also confirm that is possible to obtain a competitive performance using just a small set of features, rather than a bigger one such as Botometer.

Table 6. Bot detection results on the midterm dataset and comparison with a baseline and other techniques. The best and second-best results for each metric are bold and underlined, respectively.

Model	Precision	Recall	F1	Accuracy
Light [37]	<u>0.099</u>	0.794	<u>0.176</u>	**0.964**
D [37]	0.027	<u>0.875</u>	0.051	0.859
Botometer [37]	0.054	**0.905**	0.101	<u>0.912</u>
BGSRD	**0.8304**	0.7884	**0.8089**	0.9026

4.5. Ablation Study

Figures 4–8 presents the various evaluation indicators of each model. We can see that BGSRD and RoBERTaGCN perform the best across all datasets. Using BERT or RoBERT with GCN generally performs better than using them with GAT, except for Gilani, which is due to content posted by social bots having the characteristics of propagation, while GCN can learn the propagation characteristics of fake content. Roberta-base and roberta-large improve the performance on datasets more significantly than bert-base-uncased and bert-large-uncased. The main reason for this is that the average length in the dataset is relatively long: long text may produce more document connections transmitted through intermediate word nodes because of the graph constructed by word document statistics and, at the same time, the messages transmitted by the graph will be more favorable to passing, and the performance will be better when combined with GCN. On cresci, botometer, stock and midterm datasets, the reason the GCN model performs better than the GAT model can be explained; compared with other datasets, datasets with shorter documents (such as Gilani) have less performance improvement because the ability of the graph structure is limited. BERTGAT and RoBERTaGAT also benefit from the graph structure. Their performance is not as good as that of the GCN variant because of the lack of edge weight information.

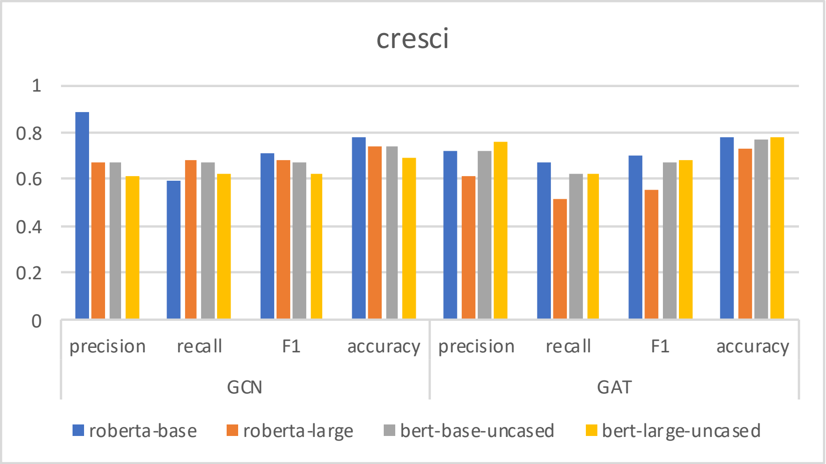

Figure 4. Results for different models on the transductive Socail bots detection cresci datasets. We ran all models 50 epochs and report the mean test result.

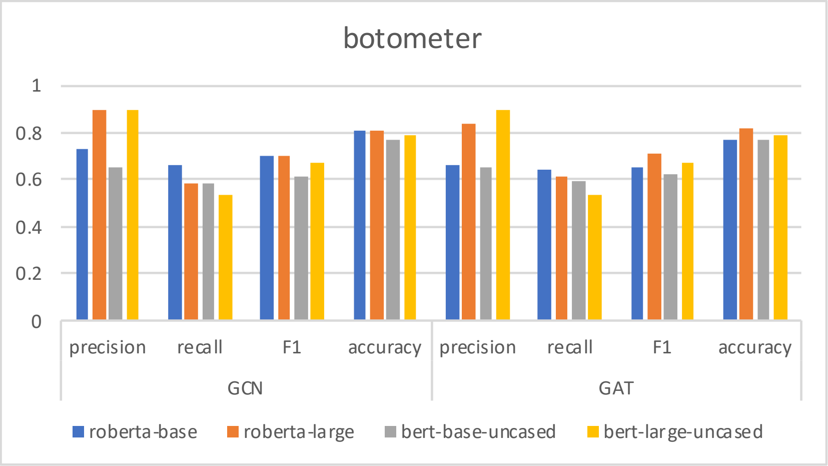

Figure 5. Results for different models on the transductive Socail bots detection botometer datasets. We ran all models 10 times and report the mean test accuracy.

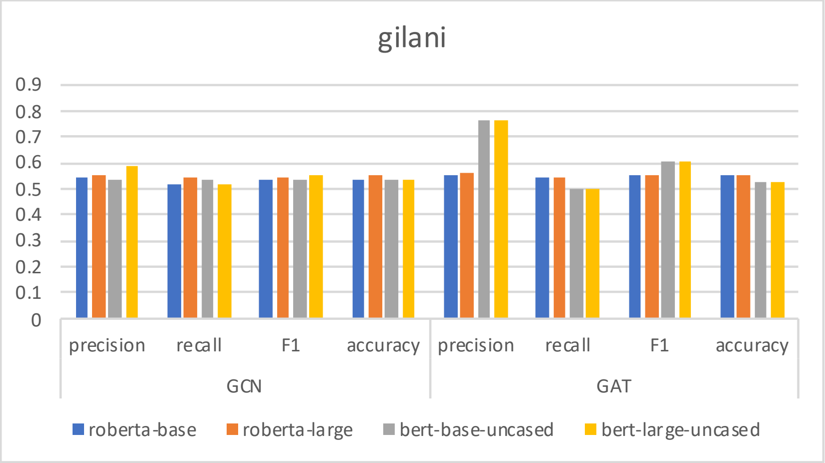

Figure 6. Results for different models on the transductive Socail bots detection gilani datasets. We ran all models 10 times and report the mean test accuracy.

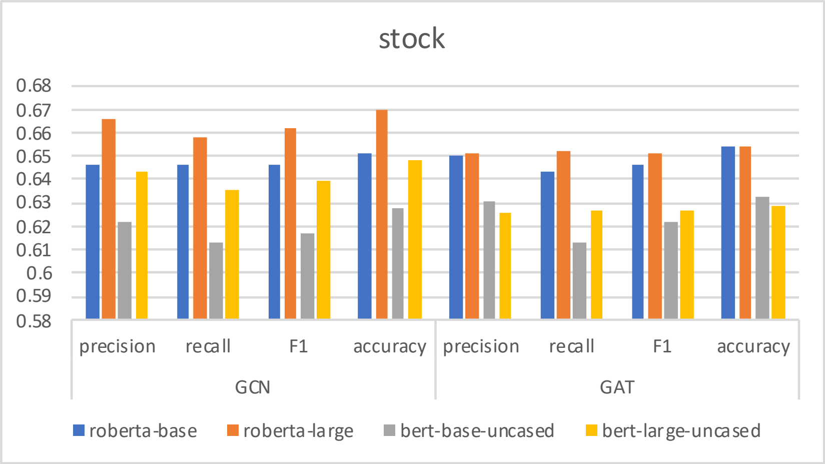

Figure 7. Results for different models on the transductive Socail bots detection stock datasets. We ran all models 10 times and report the mean test accuracy.

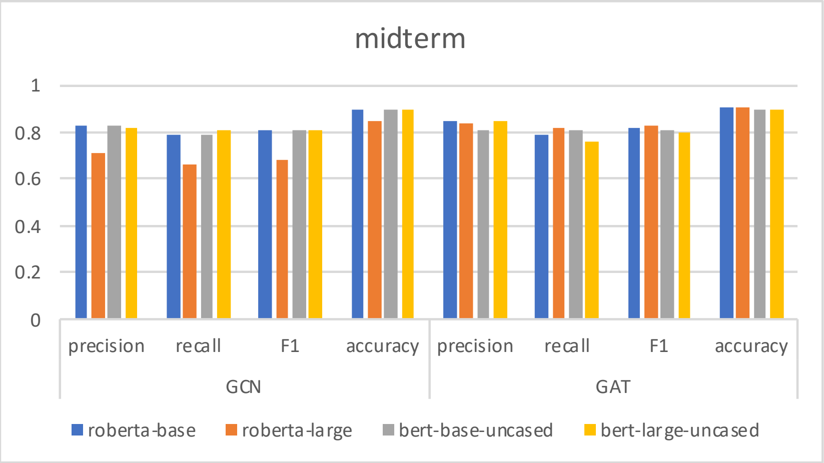

Figure 8. Results for different models on the transductive Socail bots detection midterm datasets. We ran all models 10 times and report the mean test accuracy.

4.6. The Effect of λ

The tradeoff between BGSRD and BERT is trained by λ control. The optimal value of λ will be different according to different tasks. The accuracy of RoBERTaGCN with different λ is mainly shown in Figure 9. The value of F1 is always higher on cresci, and the value of λ is larger at this time. The explanation for this is the high performance of the graph-based method. When $\lambda = 0.8$, the model achieves the best performance, which is slightly better than that of using the GCN prediction alone ($\lambda = 1$).

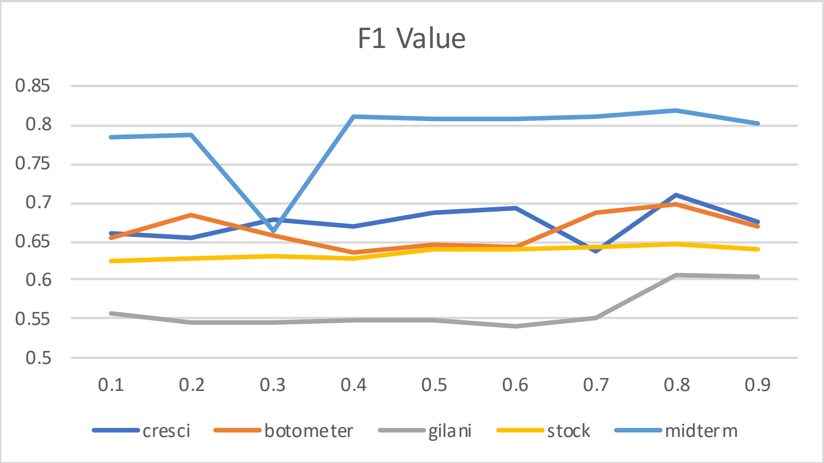

Figure 9. F1 value of RoBERTaGCN when varying λ on the test dataset. The corresponding RoBERTa baseline is represented by a dashed line.

4.7. Discussion

Powerful robots for detecting results and learning to predict documents and word embedding are mainly realized by BGSRD, which we can see from the experimental results. Among them, the GCN model is essentially transduction, which is a major limitation of this study because, in GCN training, document nodes are tested (without labels). Therefore, it is impossible for Text GCN to quickly generate embedding and predict invisible test documents. The best performance can be achieved only when a small learning rate is set by the RoBERTa module and when fine-tuned RoBERTa is used.

5. Conclusions and Future Work

BGSRD makes full use of the scale pre-training model and transduction learning for the classification of large social robots. The training of BGSRD is carried out by using a repository that stores all embedded documents. This is effective training, and some can be updated according to the small batch of samples. The detection of the classification problem of incoming text nodes is mainly carried out by constructing a heterogeneous whole corpus of generous word document maps and translating social robot texts. Limited tag documents are mainly realized through the framework of capturing global co-occurring words by BGSRD. It can be built on any document encoder and any graphic model. This method performs excellently on multiple benchmark datasets through a simple two-layer BERT combined with GCN.

We currently only detect social robots from semantic information and textual relationships and social robot detection requires more complex features to better recognize them. Future works may focus on digging for more account features under the surface, such as the sentiment analysis of tweets. The detection scheme also needs to be more comprehensive. For example, machine learning can be combined with social graphs to jointly analyze account characteristics and social network graphs, and human judgment mechanisms can be introduced into some joints. After all, humans can better identify the differences between machine accounts and human users. In order to further improve the robustness and detection capability of the detection technology, it is even necessary to further analyze the next possible update direction of the machine account and obtain the feature dimensions that can be used to detect the new machine account from the analysis results. Confrontational thinking leads to more powerful, generalized, and even preventative testing techniques.

Author Contributions: Writing—original draft preparation, Q.G.; writing—review and editing, H.X.; Data curation, Y.L. methodology, W.M.; Funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the anonymous reviewers for their valuable feedback.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Granik, M.; Mesyura, V. Fake news detection using naive Bayes classifier. In Proceedings of the 2017 IEEE first Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, 29 May–2 June 2017; pp. 900–903.
2. Cassa, C.A.; Chunara, R.; Mandl, K.; Brownstein, J.S. Twitter as a sentinel in emergency situations: Lessons from the Boston marathon explosions. *PLoS Curr.* **2013**, *5*. [CrossRef] [PubMed]
3. Conover, M.D.; Ratkiewicz, J.; Francisco, M.; Gonçalves, B.; Menczer, F.; Flammini, A. Political polarization on twitter. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.
4. Fu, Q.; Feng, B.; Guo, D.; Li, Q. Combating the evolving spammers in online social networks. *Comput. Secur.* **2018**, *72*, 60–73. [CrossRef]
5. Pan, J.; Liu, Y.; Liu, X.; Hu, H. Discriminating bot accounts based solely on temporal features of microblog behavior. *Phys. A Stat. Mech. Appl.* **2016**, *450*, 193–204. [CrossRef]
6. Chen, H.; Liu, J.; Lv, Y.; Li, M.H.; Liu, M.; Zheng, Q. Semi-supervised clue fusion for spammer detection in Sina Weibo. *Inf. Fusion* **2018**, *44*, 22–32. [CrossRef]
7. Wu, F.; Shu, J.; Huang, Y.; Yuan, Z. Co-detecting social spammers and spam messages in microblogging via exploiting social contexts. *Neurocomputing* **2016**, *201*, 51–65. [CrossRef]
8. Zheng, X.; Zhang, X.; Yu, Y.; Kechadi, T.; Rong, C. ELM-based spammer detection in social networks. *J. Supercomput.* **2016**, *72*, 2991–3005. [CrossRef]
9. Fu, H.; Xie, X.; Rui, Y. Leveraging careful microblog users for spammer detection. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 419–429.
10. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The rise of social bots. *Commun. ACM* **2016**, *59*, 96–104. [CrossRef]
11. Wang, G.A.; Mohanlal, M.; Wilson, C.; Wang, X.; Metzger, M.; Zheng, H.; Zhao, B.Y. Social Turing Tests: Crowdsourcing Sybil Detection. In Proceedings of the NDSS Symposium 2013, San Diego, CA, USA, 24–27 February 2013.

12. Twitter Inc. *Q1 2019 Letter to Shareholders*; Twitter Inc.: San Francisco, CA, USA, 2019.
13. Ahmad, J.; Farman, H.; Jan, Z. Deep learning methods and applications. In *Deep Learning: Convergence to Big Data Analytics*; Springer: New York, NY, USA, 2019; pp. 31–42.
14. Song, H.A.; Lee, S.Y. Hierarchical representation using NMF. In *International Conference on Neural Information Processing*; Springer: New York, NY, USA, 2013; pp. 466–473.
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
16. Ping, H.; Qin, S. A social bots detection model based on deep learning algorithm. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 1435–1439.
17. Kudugunta, S.; Ferrara, E. Deep neural networks for bot detection. *Inf. Sci.* **2018**, *467*, 312–322. [CrossRef]
18. Wei, F.; Nguyen, U.T. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In Proceedings of the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 12–14 December 2019; pp. 101–109.
19. Abou Daya, A.; Salahuddin, M.A.; Limam, N.; Boutaba, R. A graph-based machine learning approach for bot detection. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 8–12 April 2019; pp. 144–152.
20. Abou Daya, A.; Salahuddin, M.A.; Limam, N.; Boutaba, R. Botchase: Graph-based bot detection using machine learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 15–29. [CrossRef]
21. Cao, Q.; Sirivianos, M.; Yang, X.; Pregueiro, T. Aiding the detection of fake accounts in large scale social online services. In Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), San Jose, CA, USA, 25–27 April 2012; pp. 197–210.
22. Jia, J.; Wang, B.; Gong, N.Z. Random walk based fake account detection in online social networks. In Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 26–29 June 2017; pp. 273–284.
23. Wang, B.; Gong, N.Z.; Fu, H. GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; pp. 465–474.
24. Wang, B.; Jia, J.; Zhang, L.; Gong, N.Z. Structure-based sybil detection in social networks via local rule-based propagation. *IEEE Trans. Netw. Sci. Eng.* **2018**, *6*, 523–537. [CrossRef]
25. Gao, P.; Wang, B.; Gong, N.Z.; Kulkarni, S.R.; Thomas, K.; Mittal, P. Sybilfuse: Combining local attributes with global structure to perform robust sybil detection. In Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS), Beijing, China, 30 May–1 June 2018; pp. 1–9.
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
27. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
28. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton Hawalian Village, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
29. Li, Q.; Han, Z.; Wu, X.M. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In Proceedings of the AAAI, New Orleans, LA, USA, 2–7 February 2018.
30. Mazza, M.; Cresci, S.; Avvenuti, M.; Quattrociocchi, W.; Tesconi, M. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In Proceedings of the 10th ACM Conference on Web Science, Boston, MA, USA, 30 June–3 July 2019; pp. 183–192.
31. Yang, K.C.; Varol, O.; Davis, C.A.; Ferrara, E.; Flammini, A.; Menczer, F. Arming the public with artificial intelligence to counter social bots. *Hum. Behav. Emerg. Technol.* **2019**, *1*, 48–61. [CrossRef]
32. 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 2017, Sydney, Australia, 31 July–3 August 2017; pp. 349–354.
33. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. $FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In Proceedings of the Twelfth International AAAI Conference on Web and Social Media, New Orleans, LA, USA, 25–28 June 2018.
34. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Trans. Web (TWEB)* **2019**, *13*, 1–27. [CrossRef]
35. Yang, K.C.; Varol, O.; Hui, P.M.; Menczer, F. Scalable and generalizable social bot detection through data selection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1096–1103.
36. Gilani, Z.; Wang, L.; Crowcroft, J.; Almeida, M.; Farahbakhsh, R. Stweeler: A framework for twitter bot analysis. In Proceedings of the 25th International Conference Companion on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 37–38.
37. Antenore, M.; Camacho-Rodriguez, J.M.; Panizzi, E. A comparative study of Bot Detection techniques methods with an application related to COVID-19 discourse on Twitter. *arXiv* **2021**, arXiv:2102.01148.
38. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.