A Sybil Detection Method in OSN Based on DistilBERT and Double-SN-LSTM for Text Analysis

Xiaojie Xu[1], Jian Dong[2], Zhengyu Liu[1], Jin Yang[1(✉)], Bin Wang[3], and Zhaoyuan Wang[3]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China
yangjin66@scu.edu.cn
[2] Third Research Institute of Ministry of Public Security, Shanghai 200031, China
[3] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China

Abstract. Sybil attacks are increasingly rampant in online social networks (OSNs); thus, Sybil detection is one of the key issues in OSN security research. Sybils in OSNs are often used by attackers for public opinion intervention, topic flow filling, and dissemination of false and malicious messages. Therefore, if the credibility of the Sybil can be analyzed, then the harm of Sybil attacks can be prevented to a certain extent. Based on the analysis of existing Sybil detection research, this paper proposes an end-to-end Sybil detection model based on the Bidirectional Encoder Representations from Transformers (BERT) model that analyzes tweet text content. Considering the problems of the existing datasets, we built a dataset for text content analysis of tweets based on the hot political topic of the 2020 US presidential election. Accordingly, this study used a distilled version of BERT, DistilBERT, as the sentence embedding model, and the double self-normalizing long short-term memory (Double-SN-LSTM) recurrent neural network model as the classification detection model. The final experimental effect was greatly improved compared with the existing analysis methods, and it had a better detection effect for the more concealed Sybils.

Keywords: Sybil Attack · DistilBERT · Double-SN-LSTM

1 Introduction

A Sybil attack was originally used to describe a form of attack that acts on a peer-to-peer network (P2P), for example, when an attacker uses a single node in a P2P network to forge multiple identities, weaken network redundancy, reduce network robustness, and interfere with normal activities. Over time, Sybil attacks have also been targeted to the Tor network, Internet of Things (IoT), and Online Social Network (OSN). This paper focuses on Sybil attacks in OSNs, which specifically refers to the attacker using a few nodes in the social network to control multiple false identities, thereby using these identities to control or affect many normal nodes.

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 64–76, 2021.
https://doi.org/10.1007/978-3-030-90022-9_4

We use the method of detecting Sybils in the OSN to reduce the harm of Sybil attacks. The research object is the current popular online social media network, Twitter. Among the existing content-based Sybil detection methods, the most popular is the classification method of machine learning (ML), and its research focuses on feature engineering. The purpose of feature engineering is to construct effective features, that is, to collect and optimize multi-dimensional features from user profiles and user social graph structures to characterize each user. To improve the performance of the classifier, feature extraction requires not only professional experience to extract high-quality features but also manual determination of the contribution of the selected features. Moreover, attackers have proposed a series of bypass strategies based on existing feature engineering perspectives so that such Sybils often have personal information that is very similar to real accounts, such as avatars, a self-introduction, location, and established social network relationships. Overall, the existing content-based methods largely rely on manual intervention, and it is difficult for them to deal with highly disguised Sybils, which results in increased classification errors and high manual consumption.

In response to the above problems, this paper proposes an end-to-end classification model based on the analysis of user-published content for Sybil detection in OSNs. The proposed model can automatically extract features and learn directly from the original input. In other words, the proposed model greatly saves the workload of manual design, selection, and verification of features. Furthermore, judgments are only made based on the contents posted by users, and it achieves or even exceeds the classification accuracy of existing related research with less input, which allows it to better avoid the elaborate disguises of attackers.

The main contributions can be summarized as follows:

(1) We propose an end-to-end classification and detection model based on the content posted by users. This method avoids the highly user-friendly features of carefully forged Sybils, which not only improves the classification accuracy but also reduces unnecessary expenses.
(2) The proposed analysis model uses the Bidirectional Encoder Representations from Transformers (BERT) model that has outstanding performance in natural language processing (NLP) tasks for text analysis. At the same time, it considers the timeliness of BERT and compares and chooses its optimized distilled version of BERT, called the DistilBERT pre-training model. Moreover, to fully consider time features, the time-series association between each text content is extracted through the two-layer long short-term memory (LSTM) recurrent neural network (RNN) model, which was proven to be effective.
(3) Based on tweets regarding the 2020 US presidential election, we constructed a dataset with real accounts and Sybil tags. This dataset contains 222,802 tweets related to Biden or Trump during the election of the US president from October 15, 2020, to November 3, 2020, in which there are 949 real accounts and 987 Sybils. The dataset has now been published in the GitHub open-source warehouse, and it can be used by relevant researchers for further analysis and research.

The remainder of this paper is organized as follows: Section 2 introduces the related work, Section 3 describes the proposed system, Section 4 provides the experimental results, and finally, Section 5 gives a conclusion.

2 Related Work

The detection of Sybils is dynamic and iteratively updated. In this section, we will introduce structure-based methods and content-based detection methods.

2.1 Structure-based Inspection Method

Structure-based detection methods mainly refer to the structure based on the social network graph. These methods distinguish Sybil from human accounts by analyzing the edges and nodes of the social network graph. Yu et al. [1] proposed a decentralized protocol, SybilGuard, to use random routing to identify Sybil nodes and limit the impact of Sybil attacks. Based on SybilGuard, Yu et al. [2] proposed a decentralized protocol SybilLimit with the same idea as SybilGuard, but they applied a different random walk-based method. Danezis and Mittal [3] proposed a centralized Sybil detection algorithm, which calculates the probability of a Sybil by using a Bayesian algorithm. A new method that relies on the basic properties of social network graphs and ranks nodes according to the possibility of Sybil attacks perceived by users was proposed by Cao et al. [4]. Another study [5] mentioned that Sybils can be further detected by removing the edges of the Sybil attacks that have been perceived by the users. Compared with Cao et al.'s method [4], it achieves better detection accuracy. Wei et al. proposed a mechanism based on network topology and random walk to defend against Sybil attacks in large OSNs.

Yang et al. [6] proposed a system that further utilizes user interaction information. They use a trust-based voting distribution and global voting aggregation to evaluate whether a user is a Sybil. It is worth mentioning that the system performs better than many existing ranking systems in the actual OSN environment. To reduce the complexity of running time and reduce the dependence on known trusted nodes, Misra et al. [7] proposed a Sybil community detection algorithm based on the obvious possibility of a Sybil account community.

Bansal et al. [9] modified the previous structure-based methods by using the trust value between each user. Its false positive rate was 14% less than the results in the studies by Yu et al. [1] and other researchers [8]. Wang et al. [10] used some of the advantages of confidence-based propagation and random walk-based methods to make them orders of magnitude more scalable than semi-supervised learning methods [11]. Zhang et al. [12] used a combination of three RW-based algorithms to utilize user activities and detect Sybils.

2.2 Content-Based Inspection Method

Content-based detection methods mainly use ML methods. Many researchers have focused on studying how to make better use of the multi-dimensional features of users in OSNs and then use ML classifiers for classification. Wang et al. [13] proposed a

server-side clickstream model, which groups users whose clickstreams are close to each other into behavior clusters. However, when using an unbalanced training dataset, the experimental data shows that the false-positive rate (FPR) increases. Alsaleh et al. [14] established some classification models based on Twitter's user characteristics. They used four different ML algorithms: decision tree (C4.5), decision tree (random forest), support vector machine, and multilayer neural network. Kang et al. [15] combined the characteristics of different users and the reliability of the network structure and constructed a user discriminant formula to identify Sybils in OSNs, and the FPR fluctuated between 3.74% and 14.96%. Xia et al. [16] proposed a Sybil detection method based on the credibility of attribute information. They calculated the credibility of the user by using the Euclidean distance between the center of the Sybil attribute and users' attribute, and the degree is used as a key parameter to classify Sybils. Mulamba et al. [17] chose the ML models AdaBoost and KNN as the classifiers. Al-Qurishi [18] and others further used the user's characteristic information and deep neural network to establish a predictive model. This was the first time that deep learning methods were applied to the field of Sybil detection. Due to the rapid growth of OSNs, the traditional methods lack robustness, and Sybils can imitate human users to bypass detection. In this article, we try to build a hierarchical deep learning network structure to make better use of the user's text feature information and minimize Sybil detection error.

3 Methodology and System Design

The Sybil detection method proposed in this paper includes experimental data preprocessing, and end-to-end detection model classification. The detection model mainly includes two parts, as shown in Fig. 1. The first part is the sentence embedding module, which uses the pre-training model DistilBERT to extract the semantic features of each tweet text posted by the user. The second part is an RNN module, which uses LSTM to extract the inter-related features of multiple tweets posted by a single user to better capture the time period of the state of the user. Finally, it is input into the neuron whose activation function is sigmoid to give the correct judgment. This section will elaborate on each part of the process in detail.

3.1 Data Preprocessing

The number of tweets posted by each user in a period is unpredictable, and thus data preprocessing first needs to determine the maximum number of tweets of each user to build a text matrix. Then, the maximum length of each tweet must be determined. Due to Twitter's limitation on tweet length, the maximum length of the tweet was required to be 256 characters, and the part that exceeded the maximum number of characters was truncated. Furthermore, it was very important to normalize each tweet, that is, to replace all links in the tweets with a unified form of http://u, which helped the model ignore invalid content and optimize the classification effect of the model.

Then, the processing result was input into the tokenizer of DistilBERT for word segmentation and indexing. The tokenizer we chose was WordPiece, which uses Byte-Pair Encoding (BPE) to achieve more fine-grained segmentation. Specifically, in the

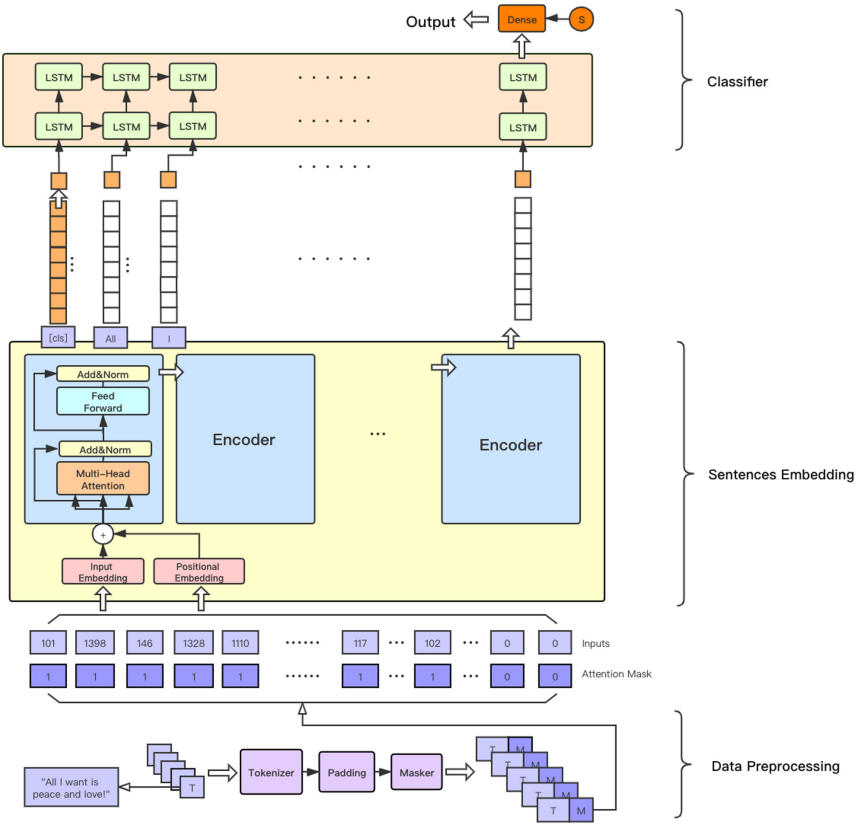

Fig. 1. Architecture of the proposed detection model.

processing of words with different suffixes only, this method divides words with different suffixes only into a collection of the word itself and different suffixes. This reduces the number of words in the vocabulary, thereby increasing the speed of subsequent processing. Indexing refers to replacing the word segmentation of the original sentence with the vocabulary index according to the input vocabulary. The sentence after word segmentation and indexing needs to be filled up to the maximum sentence length and the method used in this article fills in 0 at the end, and thus it is necessary to use Mask to distinguish between the filled part and the data part.

After data preprocessing, each user corresponded to a text matrix and a mask matrix with uniform dimensions. The dimensions of the matrix were the maximum number of tweets and the maximum length of each tweet. The matrix after data preprocessing was further used as the input of the next module for sentence embedding.

3.2 Sentence Embedding

Sentence embedding refers to the conversion of a fixed-length sentence expressed in the form of a word index value into a vector form with a fixed dimension. The embedded

vector must well understand and extract the semantic features of the text, as the input of the classifier model helps the classifier model achieve better classification results.

This study used DistilBERT [21] as the sentence embedding model. The DistilBERT model uses **Knowledge Distillation** based on the BERT model to achieve a faster, lighter, and very similar model to the original model. Knowledge distillation can be understood as a method that enables a more lightweight and compact student model to learn a more massive and excellent teacher model. When BERT and DistilBERT are trained in the teacher-student structure, the DistilBERT model simulates the output distribution of BERT, and DistilBERT can learn most of the experience of the BERT model. The objective function of this process is as follows:

$$\mathrm{L} = \alpha \mathrm{L}_{\mathrm{soft}} + \beta \mathrm{L}_{\mathrm{hard}} \tag{1}$$

Here, L_{soft} represents the cross-entropy of the SoftMax output of the student model and the teacher model at the same temperature. The formula is as follows:

$$\mathrm{L}_{\mathrm{soft}} = -\sum\nolimits_{\mathrm{j}}^{\mathrm{N}} \mathrm{p}_{\mathrm{j}}^{\mathrm{T}} \log \mathrm{q}_{\mathrm{j}}^{\mathrm{T}} \tag{2}$$

Here, p_j^T and q_j^T respectively represent the output probability values of the teacher model and the student model to the i label at the temperature T.

$$\mathrm{p}_{\mathrm{j}}^{\mathrm{T}} = \frac{\exp\left(\frac{\mathrm{v_i}}{\mathrm{T}}\right)}{\sum_{\mathrm{k}}^{\mathrm{N}} \exp\left(\frac{\mathrm{v_k}}{\mathrm{T}}\right)} \tag{3}$$

$$\mathrm{q}_{\mathrm{j}}^{\mathrm{T}} = \frac{\exp\left(\frac{\mathrm{z_i}}{\mathrm{T}}\right)}{\sum_{\mathrm{k}}^{\mathrm{N}} \exp\left(\frac{\mathrm{z_k}}{\mathrm{T}}\right)} \tag{4}$$

Here, L_{hard} Indicates the cross-entropy between the SoftMax output of the Student model and the ground truth when the $T = 1$, and the formula is as follows:

$$\mathrm{L}_{\mathrm{hard}} = -\sum\nolimits_{\mathrm{j}}^{\mathrm{N}} \mathrm{c}_{\mathrm{j}} \log \mathrm{q}_{\mathrm{j}}^{1} \tag{5}$$

Experiments showed that the number of parameters of DistilBERT was 40% less than that of the BERT model, but it still retained 97% of the performance.

The preprocessed text and mask are used as the input of DistilBERT. After the forward propagation of DistilBERT, the first participle "[CLS]" output by the model is taken as the input of the classification layer. This is because "[CLS]" as the starting tag of the text does not correspond to any word in the text. In the encoder structure of the transformer, it performs self-attention with other words of the input text. In other words, the vector of the "[CLS]" position can be represented as a vector of global text. After the sentence embedding part, the input text matrix is transformed into a vector matrix and the dimensions of the matrix are the maximum number of texts and the sentence embedding dimension.

3.3 Double-Layer LSTM Recurrent Network

The LSTM model structure is shown in Fig. 2. Long short-term memory is carefully designed to solve the long-term dependency problem of an RNN. Its main improvement method based on RNN is to add gate control, including a forget gate, input gate, and output gate.

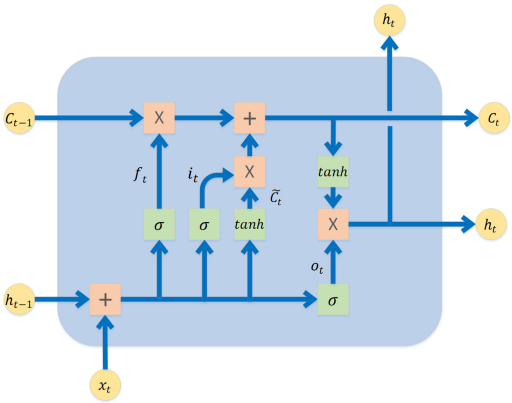

Fig. 2. Architecture of LSTM.

As shown in Fig. 2, the hidden state h_{t-1} at the last moment and the current input data x_t are activated by a sigmoid function to obtain an output f_t, which is the output of the forget gate. The output is a number between 0 and 1, where 1 represents complete retention, and 0 represents complete deletion, that is, the forget gate determines how much of the cell state C_{t-1} of the previous sequence is retained in the cell state C_t of the current sequence. The formula for the forget gate is given below.

$$\mathrm{f_t} = \sigma\left(\mathrm{W_f} * \left[\mathrm{h_{t-1}}, \mathrm{x_t}\right] + \mathrm{b_f}\right) \tag{6}$$

The input gate determines how much of the input data x_t of the current sequence is stored in the cell state C_t. It consists of two parts. The first part uses the sigmoid activation function to determine the retained value, and the output is i_t. The second part uses the tanh activation function, and the output is $\tilde{C}_t$. The formula for the input gate is given below.

$$\mathrm{i_t} = \sigma\left(\mathrm{W_i} * \left[\mathrm{h_{t-1}}, \mathrm{x_t}\right] + \mathrm{b_i}\right) \tag{7}$$

$$\tilde{\mathrm{C}}_\mathrm{t} = \tanh\left(\mathrm{W_C} * \left[\mathrm{h_{t-1}}, \mathrm{x_t}\right] + \mathrm{b_C}\right) \tag{8}$$

Displayed equations are centered and set on a separate line. At this point, the state value C_t can be updated, and C_t is composed of the previous state C_{t-1} multiplied by f_t to indicate the forgotten part, and i_t is multiplied by $\tilde{C}_t$ to indicate the new state value.

The formula is given below.

$$C_t = C_{t-1} * f_t + i_t * \widetilde{C}_t \tag{9}$$

Displayed equations are centered and set on a separate line. Finally, the output is obtained by multiplying the new state C_t after being activated by tanh with the output o_t of the sigmoid activation function. Here, the sigmoid activation function still determines the retained value, and the product indicates how much of the unit state of the current sequence affects the output value of the current sequence.

$$o_t = \sigma\left(W_o * \left[h_{t-1}, x_t\right] + b_o\right) \tag{10}$$

$$h_t = o_t * \tanh(C_t) \tag{11}$$

The double-layer LSTM RNN used in this model is shown in Fig. 1. The output of the last LSTM neuron in the network is sequentially input to the dropout layer, the fully connected layer, and a neuron whose activation function is sigmoid to output the classification result.

4 Experimental Result

In this section, we will elaborate and analyze the dataset construction, experimental preparation, evaluation indicators, and comparative experimental results.

4.1 Dataset Construction

The baseline dataset in this article was derived from Kaggle's public dataset ① **US Election 2020 Tweets dataset** [19] and ② **Political Tweets from Twitter Bots dataset** [20]. The public dataset ① provides a total of 1,727,003 tweets with Donald Trump and Joe Biden as relevant hashtags during the US general election from October 15, 2020, to November 3, 2020. The public dataset ② provides a total of 198,550 political-related tweets issued by robots during the US general election from October 19, 2020 to 2020.11.3.

To construct a positive sample, we manually selected 987 Sybils from public dataset ②. To construct a negative sample, considering that public dataset ① must include robots and Sybils, we first delete all the data of the users whose username matched the username of the robot and then deleted all the data of users who mentioned robots in the published tweets. Furthermore, the number of tweets published in this time period needed to exceed ten to meet experiment requirements. Finally, a total of 949 negative samples were obtained by manual screening.

It should also be noted that the text content in the positive sample and the negative sample was related to the political election, and the proportion of supporting or opposing a certain party in the positive sample and the negative sample was equal so that interference of the topic content can be excluded.

4.2 Experimental Preparation

In this experiment, we determined that the maximum number of texts per user was 50, and the maximum length of each tweet was 256 characters. For users with insufficient text number and text length, the number 0 was used to fill and mask. The number of neurons in the recurrent neural network was set to 256, the loss function was set to binary cross-entropy, and the optimizer was stochastic gradient descent (SGD). The batch size used during training was 32. The evaluation indicators of the model included confuse matrix, FPR, precision, recall, accuracy, and receiver operating characteristic curve (ROC_AUC). The version of the deep learning framework we chose was TensorFlow 2.4.0. The GPU used in the experiment is GeForce RTX2080Ti, the video memory size was 12 G, the CPU model selected is Intel(R) Core (TM) i3-9100F CPU @ 3.60 GHz, and the number of cores was 4.

4.3 Comparative Experimental Results

For the selection of classifiers, we chose common RNN models for comparative experiments, including SimpleRNN, Gated Recurrent Unit (GRU), LSTM, bi-directional long short-term memory (BiLSTM), double long short-term memory (Double-LSTM), and double self-normalizing long short-term memory (Double-SN-LSTM). The inputs of the above models were all the vectors after sentence embedding by DistilBERT. The evaluation indicators of the classifier included confusion matrix, false alarm rate, precision, recall rate, accuracy rate, and ROC_AUC. Through these indicators, the classification performance of the classifier on the problem can be comprehensively evaluated. After 70 epochs, the indicators of each classifier stabilized. The value of each evaluation index on the test set is shown in Table 1.

Table 1. Comparison of the results of the experiment.

Evaluating indicators	RNN	GRU	LSTM	Double-LSTM	BiLSTM	Double-SN-LSTM
FPR	0.0000	0.0000	0.0432	0.0000	0.0000	**0.0000**
Precision	1.0000	1.0000	0.9620	1.0000	1.0000	**1.0000**
Recall	0.9900	0.9901	1.0000	0.9753	0.9950	**0.9950**
Accuracy	0.9783	0.9951	0.9807	0.9711	0.9783	**0.9975**
ROC_AUC	0.9999	1.0000	1.0000	0.9999	0.9999	**1.0000**
Time(s)	**106**	308	438	690	895	681

From the experimental results, after sentence embedding through DistilBERT, the above recurrent neural network models learned the semantic features and mutual relations of sentences very well, and they gave correct judgments. Except for LSTM, each classifier performed extremely well in terms of accuracy and FPR, indicating that the classifier did not classify normal real users as malicious Sybils, which can provide an

excellent experience in practical applications. After comparison, the SimpleRNN model had a great advantage in time efficiency due to its simple structure. However, in a comprehensive comparison, Double-SN-LSTM had better performance in various performance indicators.

The loss curve of each classifier on the training set and validation set is shown in Fig. 3. From the observation and analysis in Fig. 3 above, the loss curve of the Double-SN-LSTM model is smoother, and the loss value was lower after reaching the same epoch.

Considering the evaluation index curves of each classifier on the training set and validation set, the Double-SN-LSTM model had better stability in each evaluation index dimension, and the curves are shown in Fig. 4 and Fig. 5. Furthermore, it should be noted

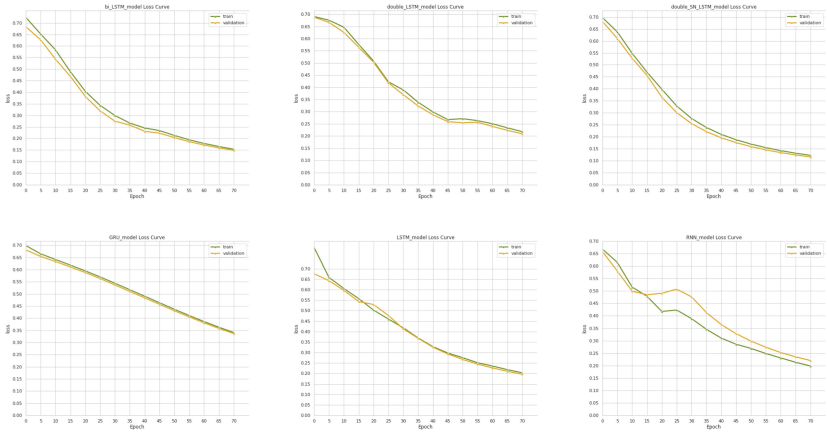

Fig. 3. The loss curve of each classifier.

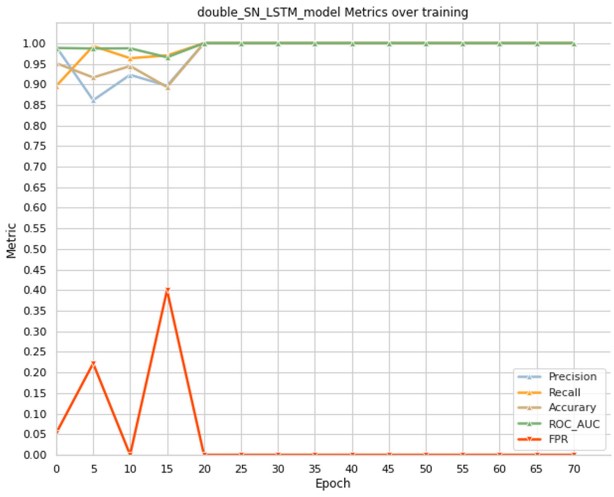

Fig. 4. Evaluation index curves of each classifier on the training set.

that the experiment used a 5-fold cross-validation method to divide the dataset, and it took the average of the five results obtained in a round as the output result of the epoch.

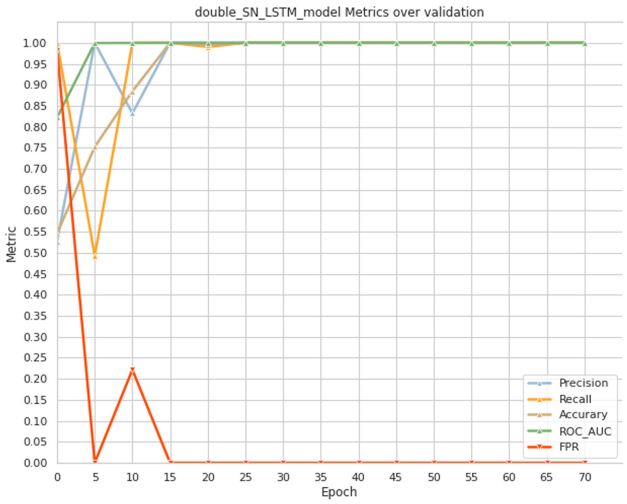

Fig. 5. Evaluation index curves of each classifier on the validation set.

5 Conclusion

In this article, we proposed an end-to-end model based only on text content analysis to detect Sybils in OSNs, that is, to detect such accounts only from the behavioral level. Static account information and social network relationships deliberately constructed by an attacker do not affect the detection results of this method, and thus it can better detect the Sybil accounts that are carefully forged by the attacker. Moreover, we constructed a new dataset that is more suitable for text analysis for other scholars to use.

We used DistilBERT to extract the semantic features of the text for embedding, which can better reflect the semantic features of the text compared to other embedding methods, such as Word2vec and Glove. After comparing experiments with various recurrent neural network models, we decided to use the Double-SN-LSTM model as the classifier. From the perspective of evaluation indicators, our model and method have a very high detection rate for Sybils.

Finally, our future work will focus on how to integrate structure-based features with content-based features to achieve a more in-depth and comprehensive detection to deal with the attackers' constantly changing bypass strategies.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant (No. 61872254), and the Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) (No.C20606), and

the Sichuan Science and Technology Program (2021JDRC0004). Xiaojie Xu and Jian Dong contribute equally to this work. We want to convey our grateful appreciation to the corresponding author of this paper, Jin Yang. He has offered advice with huge values in all stages when writing this essay to us.

References

1. Haifeng, Y., Kaminsky, M., Gibbons, P.B., Flaxman, A.D.: SybilGuard: defending against Sybil attacks via social networks. IEEE/ACM Trans. Netw. **16**(3), 576–589 (2008)
2. Haifeng, Y., Gibbons, P.B., Kaminsky, M., Xiao, F.: SybilLimit: a near-optimal social network defense against Sybil attacks. IEEE/ACM Trans. Netw. **18**(3), 885–898 (2010)
3. Danezis, G., Mittal, P.: Sybilinfer: detecting Sybil nodes using social networks. In: Proceedings of NDSS, pp. 1–15 (2009)
4. Cao, Q., et al.: Aiding the detection of fake accounts in large scale social online services. In: 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 197–210 (2012)
5. Cao, Q., Yang, X.: SybilFence: Improving social-graph-based Sybil defenses with user negative feedback (2013). arXiv:1304.3819. http://arxiv.org/abs/1304.3819
6. Yang, Z., Xue, J., Yang, X., Wang, X., Dai, Y.: VoteTrust: Leveraging friend invitation graph to defend against social network sybils. In: Proceedings of IEEE INFOCOM, pp. 2400–2408 (April 2016)
7. Misra, S., Tayeen, A.S.M., Xu, W.: SybilExposer: an effective scheme to detect sybil communities in Online social networks. In: Proceedings of IEEE International Conference Communications (ICC), pp. 1–6 (May 2016)
8. Shi, L., Yu, S., Lou, W., Hou, Y.T.: SybilShield: an agent-aided social network-based sybil defense among multiple communities. In: Proceedings of IEEE INFOCOM, pp. 1034–1042 (April 2013)
9. Bansal, H., Misra, M.: Sybil detection in Online social networks (OSNs). In: Proceedings of IEEE 6th International Conference on Advanced Computing (IACC), pp. 569–576 (February 2016)
10. Wang, B., Zhang, L., Gong, N.Z.: SybilSCAR: sybil detection in online social networks via local rule based propagation. In: Proceedings of IEEE Conference Computing Communications, pp. 1–9 (May 2017)
11. Gong, N.Z., Frank, M., Mittal, P.: SybilBelief: a semi-supervised learning approach for structure-based sybil detection. IEEE Trans. Inf. Forensics Secur. **9**(6), 976–987 (2014)
12. Zhang, X., Xie, H., Lui, J.C.S.: Sybil detection in social-activity networks: Modeling, algorithms and evaluations. In: Proceedings of IEEE 26th International Conference Network Protocols (ICNP), pp. 44–54 (September 2018)
13. Wang, G.: You are how you click: clickstream analysis for sybil detection. In: Proceedings of 22nd USENIX Security Symposium, pp. 241–256 (2013)
14. Alsaleh, M., Alarifi, A., Al-Salman, A.M., Alfayez, M., Almuhaysin, A.: TSD: Detecting sybil accounts in Twitter. In: Proc. 13th International Conference on Machine Learning and Applications, pp. 463–469 (December 2014)
15. Kang, K.: Compound approach for sybil users detection in social networks. Comput. Sci. **43**(1), 172–177 (2016)
16. Xia, Y., Pan, L., Shi, L., Zou, F.: Attribute credibility based sybil goup detection in Online social networks. In: Proc. IEEE 1st International Conference on Data Science Cyberspace (DSC), pp. 358–363 (June 2016)

17. D. Mulamba, I. Ray, and I. Ray.: On sybil classification in Online social networks using only structural features. In: Proceedings of 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1–10 (August 2018)
18. Al-Qurishi, M., Alrubaian, M., Mizanur, S.M., Rahman, A.A., Hassan, M.M.: A prediction system of sybil attack in social network using deep-regression model. Future Gener. Comput. Syst. **87**, 743–753 (2018)
19. https://www.kaggle.com/manchunhui/us-election-2020-tweets
20. https://www.kaggle.com/khanradcoder/political-tweets-from-twitter-bots
21. Sanh, V., Debut, L., Chaumond, J., et al.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)