DETECTION OF MALICIOUS SOCIAL BOTS USING LEARNING AUTOMATA WITH URL FEATURES IN TWITTER NETWORK

# PAPARAYUDU.N[1], Email:paparayudu.nagara
#JAISWAL ARYAMAN[2], Email:aryamanjaiswa104@gmail.com
#SRISAICHARAN[3], Email:charanksrisai@gmail.com
#CHINTA KOUSHIK[4],Asst. Professor, TKR College of Engineering and Technology, Hyderabad
#BTech Student, TKR College of Engineering and Technology, Hyderabad, INDIA.

Abstract: Malicious social bots generate fake tweets and automate their social relationships either by pretending like a follower or by creating multiple fake accounts with malicious activities. Moreover, malicious social bots post shortened malicious URLs in the tweet in order to redirect the requests of online social networking participants to some malicious servers. Hence, distinguishing malicious social bots from legitimate users is one of the most important tasks in the Twitter network. To detect malicious social bots, extracting URL-based features (such as URL redirection, frequency of shared URLs, and spam content in URL) consumes less amount of time in comparison with social graph-based features (which rely on the social interactions of users). Furthermore, malicious social bots cannot easily manipulate URL redirection chains. In this article, a learning automata-based malicious social bot detection (LA-MSBD) algorithm is proposed by integrating a trust computation model with URL-based features for identifying trustworthy participants (users) in the Twitter network. The proposed trust computation model contains two parameters, namely, direct trust and indirect trust. Moreover, the direct trust is derived from Bayes' theorem, and the indirect trust is derived from the Dempster-Shafer theory (DST) to determine the trustworthiness of each participant accurately. Experimentation has been performed on two Twitter data sets, and the results illustrate that the proposed algorithm achieves improvement in precision, recall, F-measure, and accuracy compared with existing approaches for MSBD.

KEYWORDS:Dempster-shefer ,MSBD, malicious URLs, malicious social bot detection

1.INTRODUCTION

Malicious social bot is a software program that pretends to be a real user in online social networks (OSNs). Moreover, malicious social bots perform several malicious attacks, such as spread social spam content, generate fake identities, manipulate online ratings, and perform phishing attacks. In Twitter, when a participant (user) wants to share a tweet containing URL(s) with the neighboring participants (i.e., followers or followees), the participant adapts URL shortened service in order to reduce the length of URL (because a tweet is restricted up to 140 characters). Moreover, a malicious social bot may post shortened phishing URLs in the tweet. When a participant clicks on a shortened phishing URL, the participant's request will be redirected to intermediate URLs associated with malicious servers that, in turn, redirect the user to malicious web pages. Then, the legitimate participant is exposed to an attacker. This leads to Twitter network suffering from several vulnerabilities (such as phishing attack). Several approaches have been proposed to detect spam in the Twitter network [5]–[8]. These approaches are based on tweet-content features, social relationship features, and user profile features. However, the malicious social bots can manipulate profile features, such as hashtag ratio, follower ratio, URL ratio, and the number of retweets. The malicious social bots can also manipulate tweet-content features, such as sentimental words, emoticons, and most frequent words used in the tweets, by manipulating the content of each tweet [9]. The social relationship-based features are highly robust because the malicious social bots cannot easily manipulate the social interactions of users in the Twitter network. However, extracting social relationship-based features consumes a huge amount of time due to the massive volume of social network graph [10]. Therefore, identifying the malicious social bots from the legitimate participants is a challenging task in the Twitter network. The existing malicious URL detection approaches [11], [12] are based on DNS information and lexical properties of URLs. The malicious social bots use URL redirections in order to avoid detection [13]. However, for detectors, identification of all malicious social bots is an issue because malicious social bots do not post malicious URLs directly in the tweets. Thus, it is important to identify malicious URLs (i.e., harmful URLs) posted by malicious social bots in Twitter. Most of the existing approaches [14], [15] are based on supervised learning algorithms, where the model is trained with the labeled data in order to detect malicious bots in OSNs. However, these approaches rely on statistical features instead of analyzing the social behavior of users [16]. Moreover, these approaches are not highly robust in detecting the temporal data patterns with noisy data (i.e., where the data is biased with untrustworthy or fake information) because the behavior of malicious bots changes over time in order to avoid detection [17], [18]. This motivated us to consider one of the reinforcement learning techniques (such as the learning automata (LA) model) to handle temporal data patterns. In this work, we design an LA model to detect malicious social bots with improved precision and recall. In this article, the malicious behavior of participants is analyzed by considering features extracted from the posted URLs (in the tweets), such as URL redirection, frequency

Related works:

Besel et al. analyzed social botnet attack on Twitter. The authors have presented that social bots use URL shortening services and URL redirection in order to redirect users to malicious web pages. Echeverria and Zhou presented methods to detect, retrieve, and analyze botnet over thousands of users to observe the social behavior of bots.

social bot hunter model has been presented based on the user behavioral features, such as follower ratio, the number of URLs, and reputation score.

a trust model has been designed to detect malicious activities in an OSN. The authors analyzed that the low trust value of a user indicates that the information spread by the user is considered as untrustworthy.

An MSBD approach has been proposed by considering user behavioral features, such as commenting, liking, and sharing. Madisetty and Desarkar have developed five different convolutional neural network models by considering tweet features.

Social botnet detection algorithm is proposed by considering spam content in tweets and trust to identify social bots. Gupta et al. designed a framework for detecting spammers in the Twitter network using different machine learning algorithms. In this article, we focus to detect malicious social bots (who perform phishing attacks) by considering various URL-based features using an LA model. Several spam-detection approaches have been proposed in the Twitter network to distinguish nonspam accounts and spam accounts. Moreover, these studies consider user profile features, which can easily be modified by malicious bots. To avoid feature manipulation, Yang et al. considered social relationships between malicious users and with their neighboring users based on closeness centrality. Moreover, profile features and social interaction features may not help in detecting malicious URLs that are posted by the participants. To address the above-mentioned problem, Janabi et al. considered URL-based features (such as URL length, Http-302 status code, and disabling right click) to distinguish legitimate URLs from suspicious URLs. RL-based approach is proposed to detect spam tweets in Twitter based on the tweet content and URL redirection chains. Patil and Patil used decision tree classifiers with statistical features in order to detect malicious URLs. Moreover, social bots may use malicious URL redirections in order to avoid detection. Thus, malicious social bots can attack legitimate users by misleading detectors. In this article, to protect against the malicious social bot attacks, we propose to identify the malicious tweets (which contain malicious URLs) in Twitter based on the lexical properties of URL and URL redirection chains.

Proposed System:

In this article, the malicious behavior of participants is analyzed by considering features extracted from the posted URLs (in the tweets), such as URL redirection, frequency of shared URLs, and spam content in URL, to distinguish between legitimate and malicious tweets. To protect against the malicious social bot attacks, our proposed LA-based malicious social bot detection (LA-MSBD) algorithm integrates a trust computational model with a set of URL-based features for the detection of malicious social bots. The proposed trust computational model contains two parameters, namely, direct trust and indirect trust. The direct trust value is derived from the Bayesian learning [19] (by considering URL-based features) to determine the trustworthiness of tweets posted by each participant. In addition to the direct trust, belief values (i.e., indicators for determining indirect trust) are collected from multiple neighbors of a participant. This is due to the fact that in case the neighbors of a participant are trustworthy, the participant is likely to be trustworthy. Furthermore, Dempster's combination rule aggregates the belief values provided by multiple one-hop neighboring participants in order to evaluate the indirect trust value of participants in the Twitter network. Moreover, in our work, the belief values provided by multiple neighboring participants are considered to be independent. The proposed LA-MSBD algorithm helps to detect malicious social bots accurately (in terms of precision, recall, F-measure, and accuracy) in Twitter. The major contributions are as follows.

Implementation:

We proposed as an alternative to the user-based neighborhood approach. We first consider the dimensions of the input and output of the neural network. In order to maximize the amount of training data we can feed to the network, we consider a training example to be a user profile (i.e. a row from the user-item matrix R) with one rating withheld. The loss of the network on that training example must be computed with respect to the single withheld rating. The consequence of this is that each individual rating in the training set corresponds to a training example, rather than each user. As we are interested in what is essentially a regression, we choose to use root mean squared error (RMSE) with respect to known ratings as our loss function. Compared to the mean absolute error, root mean squared error more heavily penalizes predictions which are further off. We reason that this is good in the context of recommender system because predicting a high rating for an item the user did not enjoy significantly impacts the quality of the recommendations. On the other hand, smaller errors in prediction likely result in recommendations that are still useful—perhaps the regression is not exactly correct, but at least the highest predicted rating are likely to be relevant to the user.

Data Processing is a task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images and many more, depending on the task we are performing and the requirements of the machine.

This might seem to be simple but when it comes to really big organizations like Twitter, Facebook, Administrative bodies like Paliament, UNESCO and health sector organizations, this entire process needs to be performed in a very structured manner.

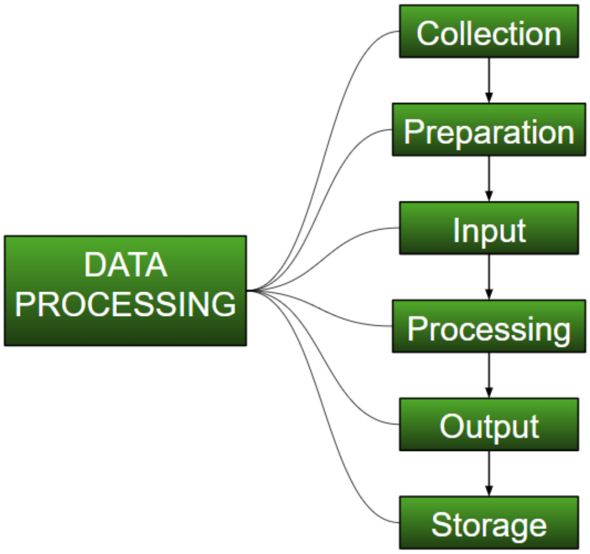


Figure 1:system arhitecture

Collection:

The most crucial step when starting with ML is to have data of good quality and accuracy. Data can be collected from any authenticated source like data.gov.in, Kaggle or UCI dataset repository.For example, while preparing for a competitive exam, students study from the best study material that they can access so that they learn the best to obtain the best results. In the same way, high-quality and accurate data will make the learning process of the model easier and better and at the time of testing, the model would yield state of the art results.

A huge amount of capital, time and resources are consumed in collecting data. Organizations or researchers have to decide what kind of data they need to execute their tasks or research.

Example: Working on the Facial Expression Recognizer, needs a large number of images having a variety of human expressions. Good data ensures that the results of the model are valid and can be trusted upon.

Preparation:

The collected data can be in a raw form which can’t be directly fed to the machine. So, this is a process of collecting datasets from different sources, analyzing these datasets and then constructing a new dataset for further processing and exploration. This preparation can be performed either manually or from the automatic approach. Data can also be prepared in numeric forms also which would fasten the model’s learning.

Example: An image can be converted to a matrix of N X N dimensions, the value of each cell will indicate image pixel.

Input:

Now the prepared data can be in the form that may not be machine-readable, so to convert this data to readable form, some conversion algorithms are needed. For this task to be executed, high computation and accuracy is needed. Example: Data can be collected through the sources like MNIST Digit data(images), twitter comments, audio files, video clips.

Processing:

This is the stage where algorithms and ML techniques are required to perform the instructions provided over a large volume of data with accuracy and optimal computation.

Output:

In this stage, results are procured by the machine in a meaningful manner which can be inferred easily by the user. Output can be in the form of reports, graphs, videos, etc

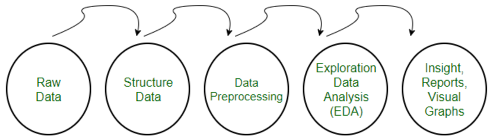
Storage:

This is the final step in which the obtained output and the data model data and all the useful information are saved for the future use.

Data Preprocessing for Machine learning in Python

- Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

- Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Need of Data Preprocessing

- For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

- Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

Rescale Data

- When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.
- This is useful for optimization algorithms in used in the core of machine learning algorithms like gradient descent.
- It is also useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like K-Nearest Neighbors.
- We can rescale your data using scikit-learn using the MinMaxScaler class.Binarize Data (Make Binary)
- We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.

- This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.
- We can create new binary attributes in Python using scikit-learn with the Binarizer class.

Standardize Data

- Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
- We can standardize data using scikit-learn with the StandardScaler class.

CONCLUSION

This article presents an LA-MSBD algorithm by integrating a trust computational model with a set of URL-based features for MSBD. In addition, we evaluate the trustworthiness of tweets (posted by each participant) by using the Bayesian learning and DST. Moreover, the proposed LA-MSBD algorithm executes a finite set of learning actions to update action probability value (i.e., probability of a participant posting malicious URLs in the tweets). The proposed LA-MSBD algorithm achieves the advantages of incremental learning. Two Twitter data sets are used to evaluate the performance of our proposed LA-MSBD algorithm. The experimental results show that the proposed LA-MSBD algorithm achieves up to 7% improvement of accuracy compared with other existing algorithms. For The Fake Project and Social Honeypot data sets, the proposed LA-MSBD algorithm has achieved precisions of 95.37% and 91.77% for MSBD, respectively. Furthermore, as a future research challenge, we would like to investigate the dependence among the features and its impact on MSBD.

References:

[1] P. Shi, Z. Zhang, and K.-K.-R. Choo, “Detecting malicious social bots based on clickstream sequences,” IEEE Access, vol. 7, pp. 28855–28862, 2019.

[2] G. Lingam, R. R. Rout, and D. V. L. N. Somayajulu, “Adaptive deep Q-learning model for detecting social bots and influential users in online social networks,” Appl. Intell., vol. 49, no. 11, pp. 3947–3964, Nov. 2019.

[3] D. Choi, J. Han, S. Chun, E. Rappos, S. Robert, and T. T. Kwon, “Bit.ly/practice: Uncovering content publishing and sharing through URL shortening services,” Telematics Inform., vol. 35, no. 5, pp. 1310–1323, 2018.

[4] S. Lee and J. Kim, “Fluxing botnet command and control channels with URL shortening services,” Comput. Commun., vol. 36, no. 3, pp. 320–332, Feb. 2013.

[5] S. Madisetty and M. S. Desarkar, “A neural network-based ensemble approach for spam

detection in Twitter," IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 973–984, Dec. 2018.

[6] H. B. Kazemian and S. Ahmed, "Comparisons of machine learning techniques for detecting malicious webpages," Expert Syst. Appl., vol. 42, no. 3, pp. 1166–1177, Feb. 2015.

[7] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in Twitter," in Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS), Jan. 2018, pp. 380–383.

[8] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), 2017, p. 3.

[9] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Key challenges in defending against malicious socialbots," Presented at the 5th USENIX Workshop Large-Scale Exploits Emergent Threats, 2012, pp. 1–4.

[10] G. Yan, "Peri-watchdog: Hunting for hidden botnets in the periphery of online social networks," Comput. Netw., vol. 57, no. 2, pp. 540–555, Feb. 2013.

[11] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A fast filter for the large-scale detection of malicious Web pages," in Proc. 20th Int. Conf. World Wide Web (WWW), 2011, pp. 197–206.

[12] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," J. Ambient Intell. Hum. Comput., vol. 10, no. 5, pp. 2015–2028, May 2019.

[13] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 million spam tweets: A large ground truth for timely Twitter spam detection," in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2015, pp. 7065–7070.

[14] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" IEEE Trans. Dependable Secure Comput., vol. 9, no. 6, pp. 811–824, Nov. 2012.

[15] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min, "Statistical features-based real-time detection of drifted Twitter spam," IEEE Trans. Inf. Forensics Security, vol. 12, no. 4, pp. 914–925, Apr. 2017.

[16] N. Rndic and P. Laskov, "Practical evasion of a learning-based classifier: A case study," in Proc. IEEE Symp. Secur. Privacy, May 2014, pp. 197–211.

[17] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "Learning-automatonbased online discovery and tracking of spatiotemporal event patterns," IEEE Trans. Cybern., vol. 43, no. 3, pp. 1118–1130, Jun. 2013.

[18] M. R. Khojasteh and M. R. Meybodi, "Evaluating learning automata as a model for cooperation in complex multi-agent domains," in Robot Soccer World Cup. Springer, 2006, pp. 410–417.

[19] C.-M. Chen, D. J. Guan, and Q.-K. Su, "Feature set identification for detecting suspicious URLs using Bayesian classification in social networks," Inf. Sci., vol. 289, pp. 133–147, Dec. 2014.

[20] T. M. Chen and V. Venkataramanan, "Dempster-shafer theory for intrusion detection in ad hoc networks," IEEE Internet Comput., vol. 9, no. 6, pp. 35–41, Nov. 2005.

[21] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in Proc. 26th Int. Conf. World Wide Web Companion- (WWW Companion), 2017, pp. 963–972.

[22] K. Lee, B. D. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on Twitter," in Proc. ICWSM, 2011, pp. 1–8.

[23] C. Besel, J. Echeverria, and S. Zhou, "Full cycle analysis of a largescale botnet attack on Twitter," in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2018, pp. 170–177.

[24] J. Echeverria and S. Zhou, "Discovery, retrieval, and analysis of the'star wars' botnet in twitter," in Proc. 2017 IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining 2017, 2017, pp. 1–8.

[25] A. Dorri, M. Abadi, and M. Dadfarnia, "SocialBotHunter: Botnet detection in Twitter-like social networking services using semisupervised collective classification," in Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervasive Intell. Comput., 4th Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech), Aug. 2018, pp. 496–503.

[26] M. Agarwal and B. Zhou, "Using trust model for detecting malicious activities in Twitter," in Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling, Predict. Springer, 2014, pp. 207–214.

[27] G. Lingam, R. R. Rout, and D. V. L. N. Somayajulu, "Detection of social botnet using a trust model based on spam content in Twitter network," in Proc. IEEE 13th Int. Conf. Ind. Inf. Syst. (ICIIS), Dec. 2018, pp. 280–285.