International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

DETECTION OF SOCIAL MALICIOUS BOTS

T.Sasi Vardhan[1], M. Sai Sarvani[2], Krovi Jaswitha[3], Mutnuri Sai Keerthana[4]
[1]Assistant Professor, Department of CSE, Malla Reddy Engineering College for Women, Hyderabad, Telangana, India.
[2,3,4]UG-Students, Department of CSE, Malla Reddy Engineering College for Women, Hyderabad, Telangana, 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 Social network. Furthermore, malicious social bots cannot easily manipulate URL redirection chains. To detect malicious social bots, a learning automata-based malicious social bot detection (LA-MSBD) algorithm is proposed by integrating a trust computation model.

INTRODUCTION

In online social networks, social bots are social accounts controlled by automated programs that can perform corresponding operations based on a set of procedures. The increasing use of mobile devices (e.g., Android and iOS devices) also contributed to an increase in the frequency and nature of user interaction via social networks. It is evidenced by the significant volume, velocity and variety of data generated from the large online social network user base. Social bots have been widely deployed to enhance the quality and efficiency of collecting and analyzing data from social network services. For example, the social bot SF Quake Bot is designed to generate earthquake reports in the San Francisco Bay, and it can analyze earthquake related information in social networks in real-time. However, public opinion about social networks and massive user data can also be mined or disseminated for malicious or nefarious purpose.

In online social networks, automatic social bots cannot represent the real desires and intentions of normal human beings, so they are usually looked upon malicious ones. For example, some fake social bots accounts created to imitate the profile of a normal user, steal user data and compromise their privacy, disseminate malicious or fake information, malicious comment, promote or advance certain political or ideology agenda and propaganda, and influence the stock market and other social and economic markets. Such activities can adversely impact the security and stability of social networking

platforms. In previous research, various methods were used to protect the security of online social network. User behavior is the most direct manifestation of user

intent, as different users have different habits, preferences, and online behavior (e.g., the way one clicks or types, as well as the speed of typing). In other words, we may be able to mine and analyze information hidden in user's online behavior to problem and identify different users. However, we also need to be conscious of situational factors that may play a role in changing user's online behavior. In other words, user behavior is dynamic and its environment is constantly changing i.e., external observable environment (e.g., environment and behavior) of application context and the hidden environment in user information. In order to distinguish social bots from normal users accurately, detect malicious social bots, and reduce the harm of malicious social bots, we need to acquire and analyze social situation of user behavior and compare and understand the differences of malicious social bots and normal users in dynamic behavior.

1.PURPOSE OF THE PROJECT

Specifically, in this paper, we aim to detect malicious social bots on social network platforms in real-time, by proposing the transition probability features between user contents based on the social situation analytics; and designing an algorithm for detecting malicious social bots based on spatiotemporal features.

2. BOTS: Internet bots, also known as web robots, WWW robots or simply bots, are software applications that run automated tasks over the Internet. Typically, bots perform tasks that are both simple and structurally repetitive, at a much higher rate than would be possible for a human alone. The largest use of bots is in web spidering, in which an automated script fetches, analyzes and files information from web servers at many times the speed of a human. Each server can have a file called robots.txt, containing rules for the spidering of that server that the bot is supposed to obey. In addition to their uses outlined above, bots may also be implemented where a response speed faster than that of humans is required (e.g., gaming bots and auction-site robots) or less commonly in situations where the emulation of human activity is required.

1. EXISTING SYSTEM

Most of the existing approaches 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. More- over, 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. 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 designed a framework for detecting spammers in the Twitter network using different machine learning algorithms. We focus to detect malicious social bots (who perform phishing attacks).

Moreover, these studies consider user profile features, which can easily be

modified by malicious bots. Moreover, profile features and social interaction features may not help in detecting malicious URLs that are posted by the

participants. Moreover, social bots may use malicious URL redirections in order to avoid detection. Thus, malicious social bots can attack legitimate users by misleading detectors.

1.DISADVANTAGES

- No efficient methods are used.
- No real time data is used.
- More complex.

1.PROPOSED SYSTEM

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. The proposed framework consists of three components: data collection, feature extraction, and LA model. To collect tweets posted by participants (users), the tweets can be crawled using Twitter Streaming APIs.

The data collection component consists of three subcomponents (i.e., subphases): reading tweets from Twitter streaming, collecting tweets, and URLs. Moreover, the collected tweets and collected URLs are stored in a repository. The feature extraction consists of two subcomponents: expanding shortened URLs and extracting feature set. Whenever

a feature extraction component obtains a shortened URL from the repository, it is converted into a long URL using URL shortened services (such as t.co, bit.ly, and tinyurl.com). For each URL (posted by the participant in the tweet), we extract several features that are based on the lexical properties of URLs (such as spam content and the presence of -, @, and symbols in the domain name) along with the features of URL redirection (such as URL redirection length and relative position of initial URL). Furthermore, we use

these features as input to the proposed LA model for MSBD. The proposed LA model is integrated with a trust evaluation model. Moreover, the trust model determines the probability of a tweet containing any malicious information (such as URL redirection, frequency of URLs, and spam content in URL). Finally, after evaluating the malicious behavior of a series of tweets posted by a participant, we classify tweets as malicious and legitimate tweets. However, malicious tweets are likely to be posted by malicious social bots. This helps in distinguishing malicious social bots from benign participants

ADVANTAGES

- This study includes the comparison of various previous methodologies proposed using different datasets and with different characteristics and accomplishments offerings and ingredients or not.

LITERATURE SURVEY

Title: Sensitive system calls based packed malware variants detection using principal component initialized Multi Layers neural networks

Year: 2018

Author: Raymond Canzanese

Methodology:

We propose a new method which first extracts a series of system calls which is sensitive to malicious behaviors, then use principal component analysis to extract features of these sensitive system calls, and finally adopt multi-layers neural networks to classify the features of malware variants and legitimate ones. Theoretical analysis and real-life experimental results show that our packed malware variants detection technique is comparable with the state-of-art methods in terms of accuracy. Our approach can achieve more than 95.6\% of detection accuracy and 0.048 s of classification time cost. We transform the packed malware variants detection problem to a system calls classification problem. To reduce the obfuscation which is caused by packers, we first extract sensitive system calls and abandon obfuscated system calls. Then we organize these sensitive system calls as a vector which will be sent to our neural net- works later. As system call is a coarse-gained and sparse representation of executables, it causes bad training approximation and feature generalization. So we next propose our principal component initialized multi-layers neural networks to efficiently and effectively train and detect malicious instance with these sparse vectors. Our approach contains the following two phases, a training phase and a detection phase. The work shown, in training phase, we monitor the system interactions of executables in Cuckoo sandbox to obtain the system calls. Each profile of executables we got from Cuckoo sandbox contains several fields: time-stamp, system call, base address, file name, executing times, etc. We only consider system calls since it can give us enough information to describe characteristics of behaviors of malware while reducing the noise and redundant.

Advantages:

Overcome the effect of unpacking behaviors of packers which add noisy information to the real behaviors of executables, which has a bad effect on accuracy.

Disadvantages:

It might be attacked by adversaries which causes security problem

Title: Multi-layer intrusion detection system with

ExtraTrees feature selection, extreme learning machine ensemble

Year: 2019

Author: Jivitesh Sharma, Charul Giri,

Methodology:

Recent advances in intrusion detection systems based on machine learning have indeed outperformed other techniques, but struggle with detecting multiple classes of attacks with high accuracy. We propose a method that works in three stages. First, the ExtraTrees classifier is used to select relevant features for seach type of attack individually for each (ELM). Then, an ensemble of ELMs is used to detect each type of attack separately. Finally, the results of all ELMs are combined using a softmax layer to refine the results and increase the accuracy further. The intuition behind our system is that multi-class classification is quite difficult compared to binary classification. So, we divide the multi-class problem into multiple binary classifications. We test our method on the UNSW and KDDcup99 datasets. we propose a novel approach based on neural networks for the problem of general purpose network intrusion detection. All previous approaches for intrusion detection either distinguish between normal traffic and attacks or can only detect one type of attack at a time. We propose to use an ensemble of ELMs for detecting all types of attacks simultaneously. Each ELM is trained for a specific type of attack, and each ELM is fed a different feature set consisting of features selected by an ExtraTree classifier for that specific attack. Training these ELMs on each and every type of attack takes less than 18 s. Our system is tested on the UNSW and NSL-KDD datasets and is able to outperform all previous machine learning-based intrusion detection systems. The results clearly show that our proposed method is able to outperform all the other methods, with a high margin. Our system is able to achieve 98.24% and 99.76% accuracy for multi-class classification on the UNSW and KDDcup99 datasets, respectively. Additionally, we use the weighted extreme learning machine to alleviate the problem of imbalance in classification of attacks, which further boosts performance. Lastly, we implement the ensemble of ELMs in parallel using GPUs to perform intrusion detection in real time.

Advantages:

Can detect multiple attacks at a time and the model uses considerably less number of features, with real-time detection due to parallel implementation, and gives state-of-the-art performance for intrusion detection

Disadvantages:

The system cannot determine the new type of attack if it is not trained on it.

Title: Apply machine learning techniques to detect malicious network traffic in cloud computing

Year:2019

Author: Amirah Alshammari

Methodology:

Computer networks target several kinds of attacks every hour and day; they evolved to make significant risks. They pass new attacks and trends; these attacks target every open port available on the network. Several tools are designed for this purpose, such as mapping networks and vulnerabilities scanning. Recently, machine learning (ML) is a widespread technique offered to feed the Intrusion Detection System (IDS) to detect malicious network traffic. The core of ML models' detection efficiency relies on the dataset's quality to train the model. This research proposes a detection framework with an ML model for feeding IDS to detect network traffic anomalies. This detection model uses a dataset constructed from malicious and normal traffic. This research's significant challenges are the extracted features used to train the ML model about various attacks to distinguish whether it is an anomaly or regular traffic. The dataset ISOT-CID network traffic part uses for the training ML model. We added some significant column features, and we approved that feature supports the ML model in the training phase. The ISOT-CID dataset traffic part contains two types of features, the first extracted from network traffic flow, and the others computed in specific interval time. We also presented a novel column feature added to the dataset and approved that it increases the detection quality.

The proposed dataset extracted from network traffic in different period and contains frame time, source MAC, destination MAC, source IP, source port, destination IP, source port, IP length, IP header length, TCP header length,

frame length, offset, TCP segment, TCP acknowledgment, in frequency number, and out frequency number. These attributes of network flow can specify packets, whether anomaly or normal. Other features that are vital and added to the ISOT-CID dataset are. APL is the average payload packet length for a time interval, PV is the variance of payload packet length for a time interval, and TBP means the average time between packets in the time interval. It consists of three stages. Stage 1 concerns the dataset preparation, and stage 2 builds the detection model. The last stage will consist of the evaluation stage, which ensures our approach accuracy for anomaly detection. This feature is depending on the rambling packet payload length in the traffic flow. Our presented results and experiment produced by this research are significant and encourage other researchers and us to expand the work as future work.

Advantages:

Provide an effective IDS that is proficient in protecting severe system components against intruders.

Disadvantages:

IDS security systems for computer networks must be very fast where it is deployed in real-time to extract the communication traffic characteristics and give its response in real-time and the deployment of this model in real networks will harm the speed required.

Title: Malware Analysis and Detection Using Data Mining and Machine Learning Classification

Year: 2017

Author: Mozammel Chowdhury, Azizur Rahman

Methodology:

Exfiltration of sensitive data by malicious software or malware is a serious cyber threat around the world that has catastrophic effect on businesses, research organizations, national intelligence, as well as individuals. Thousands of cyber criminals attempt every day to attack computer systems by employing malicious software with an intention to breach crucial data, damage or manipulate data, or to make illegal financial transfers. Protection of this data is therefore, a critical concern in the research community. We propose a comprehensive framework to classify and detect malicious software to protect sensitive data against malicious threats using data mining and machine learning classification techniques. A hybrid framework is used for malware classification integrating a binary associative memory (BAM) with a multilayer perceptron (MLP) neural network by using both signature-based and behavior-based features analysis. In this work, we employ signature-based n-gram features and behavior-based API (Application Programming Interface) call sequences for malware analysis In this work, we employ a robust and efficient approach for malware classification and detection by analyzing both signature-based and anomaly-based features. Experimental results confirm the superiority of the proposed approach over other similar methods. The proposed scheme for malware classification and detection is consisted of the following major components: (i) Pre-processing, (ii) Features extraction, (iii) Feature refinement/selection, (v) Classification, and (vi) Detection. Classification process is divided into two stages: training and

testing. In the training phase, a training set of malicious and benign files is provided to the system. The learning algorithm trains a classifier. The classifier learns from the labeled data samples. In the testing phase, a set of new malicious and benign files are fed into the classifier and classified as malware or cleanware. In this work, we propose a hybrid framework for malware classification integrating a binary associative memory (BAM) with a multilayer perceptron (MLP) neural network. This is a robust and efficient approach for malware classification and detection using a hybrid framework with combination of a binary associative memory (BAM) and a multilayer perceptron (MLP) neural network. The BAM network can significantly reduce feature dimensions collected from a large malware dataset. We employ hybrid features for malware analysis by integrating both signature-based and behavior-based features that clearly increases classification and detection accuracy.

Advantages:

The BAM network can significantly reduce feature dimensions collected from a large malware dataset

Disadvantages:

False positive is high

Title:MALCOM: Generating Malicious Comments to Attack Neural Fake News Detection Models

Year: 2020

Author: Thai Le, Suhang Wang

Methodology:

Circulation of fake news, i.e., false or misleading pieces of information, on social media is not only detrimental to individuals’ knowledge but is also creating an erosion of trust in society. Fake news has been promoted with deliberate intention to widen political divides, to undermine citizens’ confidence in public figures, and even to create confusion and doubts among communities. Hence, any quantity of fake news is intolerable and should be carefully examined and combated. Due to the high-stakes of fake news detection in practice, therefore, tremendous efforts have been taken to develop fake news detection models that can auto-detect fake news with high accuracies. In an attempt to solve these challenges, we propose MALCOM, a novel framework that can generate realistic and relevant comments in an end-to-end fashion to attack fake news detection models, that works for both black box and white box attacks. The main contributions are: This is the first work proposing an attack model against neural fake news detectors, in which adversaries can post malicious comments toward news articles to mislead cutting edge fake news detectors. Different from prior adversarial literature, our work generates adversarial texts (e.g., comments, replies) with high quality and relevancy at the sentence level in an end-to-end fashion (instead of the manipulation at the character or word level). Our model can fool five top-notch neural fake news detectors to always output real news and fake news 94% and 93.5% of the time on average. Moreover, our model can mislead black-box classifiers to always output real news 90% of the time on average. We also compare our attack model with four baselines across two real-world datasets, not only on attack performance but also on generated quality, coherency, transferability, and robustness

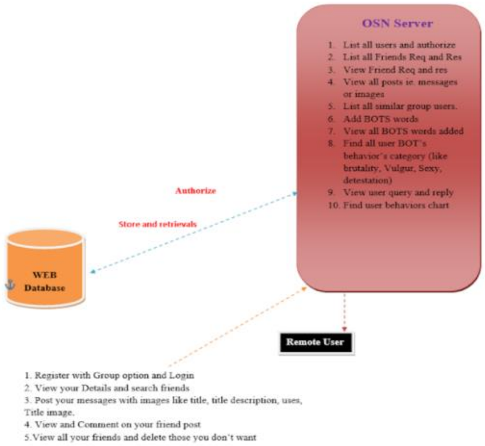
Advantages:

Malcom is shown to be more robust even under the condition when a rigorous defense system works against malicious comments

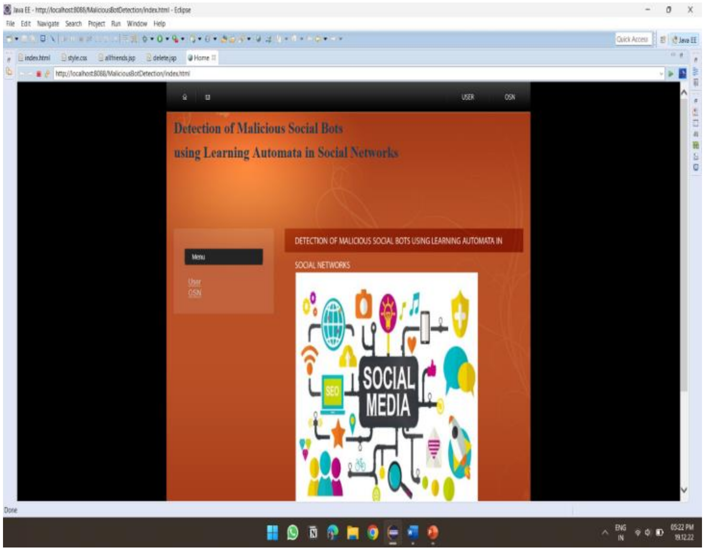
Disadvantages:

Whether or not comments generated using one sub-domain can be transferable to another is also out of scope of this model

ARCHITECTURE :


OSN Server
1. List all users and authorize
2. List all Friends Req and Res
3. View Friend Req and res
4. View all posts ie. messages or images
5. List all similar group users.
6. Add BOTS words
7. View all BOTS words added
8. Find all user BOT's behavior's category (like brutality, Vulgur, Sexy, detestation)
9. View user query and reply
10. Find user behaviors chart
Authorize
Store and retrievals
WEB Database
Remote User
1. Register with Group option and Login
2. View your Details and search friends
3. Post your messages with images like title, title description, uses, Title image.
4. View and Comment on your friend post
5.View all your friends and delete those you don't want

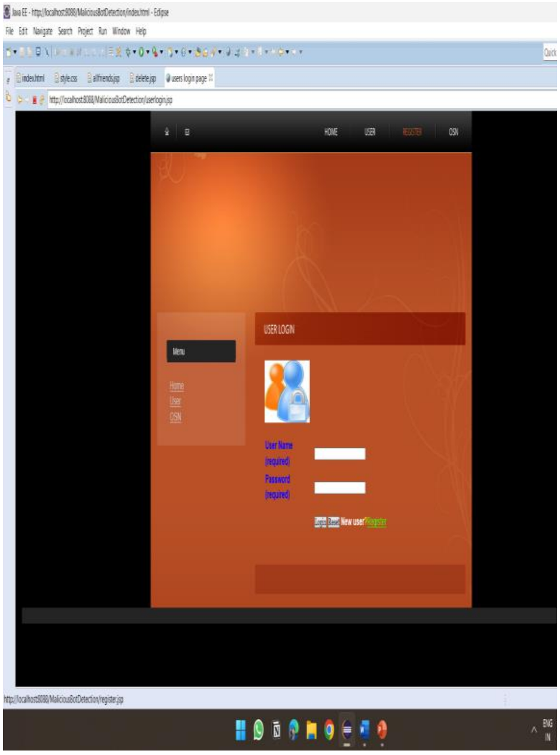
RESULT:
SCREENSHOTS:


Java EE - http://localhost:8088/MaliciousBotDetection/index.html - Eclipse

http://localhost:8088/MaliciousBotDetection/index.html

Detection of Malicious Social Bots
using Learning Automata in Social Networks

DETECTION OF MALICIOUS SOCIAL BOTS USING LEARNING AUTOMATA IN SOCIAL NETWORKS

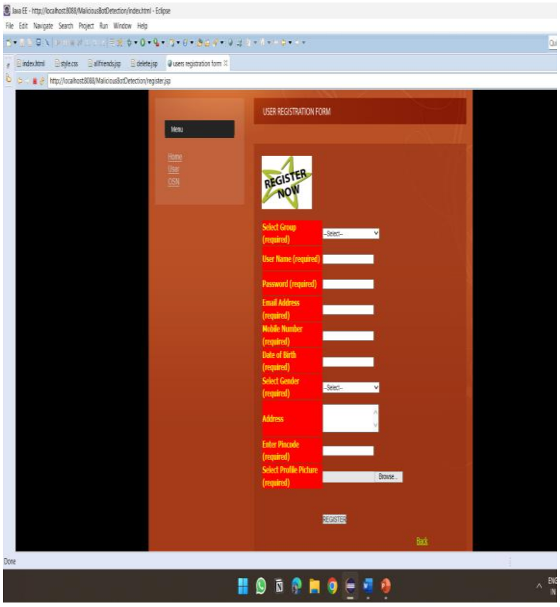

Java EE - http://localhost:8080/MaliciousBotDetection/index.html - Eclipse
File Edit Navigate Search Project Run Window Help
index.html style.css allfriends.jsp delete.jsp users login page
http://localhost:8080/MaliciousBotDetection/userlogin.jsp
HOME USER REGISTER OSN
Menu
Home
User
OSN
http://localhost:8080/MaliciousBotDetection/register.jsp

USER LOGIN

User Name (required)

Password (required)

Login Reset New user Register


Java EE - http://localhost:8080/MaliciousBotDetection/index.html - Eclipse
File Edit Navigate Search Project Run Window Help
index.html style.css allfriends.jsp delete.jsp users registration form
http://localhost:8080/MaliciousBotDetection/register.jsp
Menu
Home
User
OSN
REGISTER NOW
Done

USER REGISTRATION FORM

Select Group (required) --Select--

User Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Date of Birth (required)

Select Gender (required) --Select--

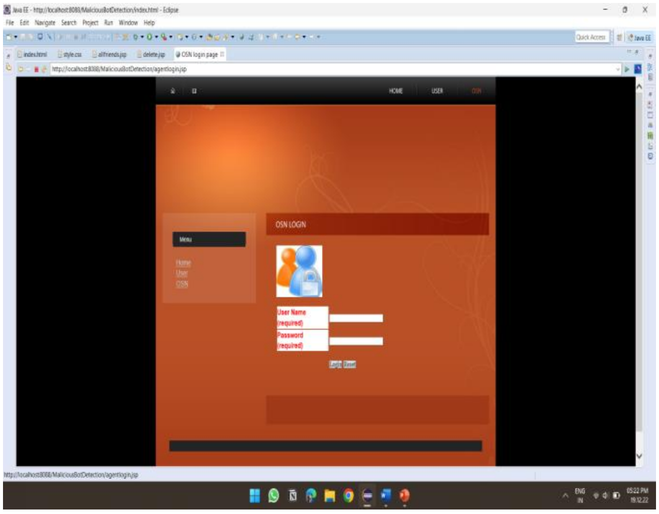
Address

Enter Pincode (required)

Select Profile Picture (required) Browse...

REGISTER

Back

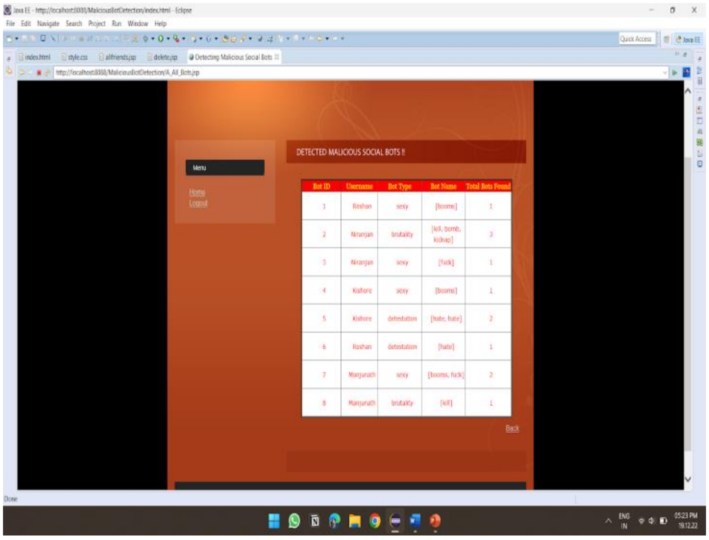

Java EE - http://localhost:8080/MaliciousBotDetection/index.html - Eclipse
File Edit Navigate Search Project Run Window Help
index.html style.css allfriends.jsp delete.jsp OSN login page
http://localhost:8080/MaliciousBotDetection/agentlogin.jsp
HOME USER OSN
Menu
Home
User
OSN
http://localhost:8080/MaliciousBotDetection/agentlogin.jsp
05:22 PM
19-12-22

OSN LOGIN

User Name (required)

Password (required)

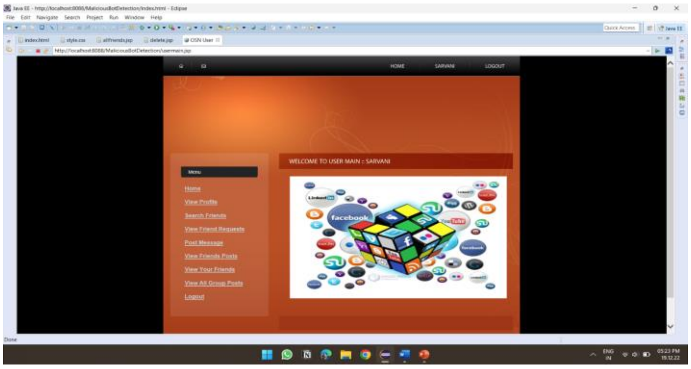
Login Reset


Java EE - http://localhost:8080/MaliciousBotDetection/index.html - Eclipse
File Edit Navigate Search Project Run Window Help
index.html style.css allfriends.jsp delete.jsp Detecting Malicious Social Bots
http://localhost:8080/MaliciousBotDetection/A_All_Bots.jsp
Menu
Home
Logout
Done
05:23 PM
19-12-22

DETECTED MALICIOUS SOCIAL BOTS !!

Bot ID	Username	Bot Type	Bot Name	Total Bots Found
1	Roshan	sexy	[boobs]	1
2	Niranjan	brutality	[kill, bomb, kidnap]	3
3	Niranjan	sexy	[fuck]	1
4	Kishore	sexy	[boobs]	1
5	Kishore	detestation	[hate, hate]	2
6	Roshan	detestation	[hate]	1
7	Manjunath	sexy	[boobs, fuck]	2
8	Manjunath	brutality	[kill]	1

Back


Java EE - http://localhost:8080/MaliciousBotDetection/index.html - Eclipse
File Edit Navigate Search Project Run Window Help
index.html style.css allfriends.jsp delete.jsp OSN User
http://localhost:8080/MaliciousBotDetection/usermain.jsp
HOME SARVANI LOGOUT
Menu
Home
View Profile
Search Friends
View Friend Requests
Post Message
View Friends Posts
View Your Friends
View All Group Posts
Logout
Done
05:23 PM
19-12-22

WELCOME TO USER MAIN : SARVANI

CONCLUSION

We proposed a novel method to accurately detect malicious social bots in online social networks. Experiments showed that transition probability between user click streams based on the social situation analytics can be used to detect malicious social bots in online social platforms accurately. In future research, additional behaviors of malicious social bots will be further considered and the proposed detection approach will be extended and optimized to identify specific intentions and purposes of a broader range of malicious social bots.

FUTURESCOPE:

In future research, additional behaviors of malicious social bots will be further considered and the proposed detection approach will be extended and optimized to identify specific intentions and purposes of a broader range of malicious social bots. Furthermore, as a future research challenge, we would like to investigate the dependence among the features and its impact on MSBD.

REFERENCES

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.

[1] 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. [2] 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.

[2] S. Madisetty and M. S. Desarkar, "A neural networkbased ensemble approach for spam detection in Twitter," IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 973–984, Dec. 2018.

[4] 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.

[5] 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

[6] 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.

[7] 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.

[8] 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.