Tweet Based Bot Detection Using Big Data ABSTRACT

Twitter is one of the most popular micro-blogging social media platforms that has millions of users. Due to its popularity, Twitter has been targeted by different attacks such as spreading rumors, phishing links, and malware. Tweet-based botnets represent a serious threat to users as they can launch large-scale attacks and manipulation campaigns. To deal with these threats, big data analytics techniques, particularly shallow and deep learning techniques have been leveraged in order to accurately distinguish between human accounts and tweet-based bot accounts. In this paper, we discuss existing techniques, and provide a taxonomy that classifies the state-of-the-art of tweet-based bot detection techniques. We also describe the shallow and deep learning techniques for tweet-based bot detection, along with their performance results. Finally, we present and discuss the challenges and open issues in the area of tweet-based bot detection.

**EXISTING SYSTEM**

Kabakus and Kara [13] provided a short comparative survey of the research work in the field of Twitter spam detection within the year range of 2009-2015. They described different detection methods within four categories: account based, tweet-based, graph-based, and hybrid-based methods. The account-based methods were shown to leverage the user profile's metadata like followers and following count and other derived features such as age of the account. While in graph-based methods, features like distance and strength of connectivity between users were shown to be used for spam detection. However, in tweet-based methods, the survey mainly focused on detecting spam using URL and its derived features, such as length and domain name. To detect a spam user, posted URLs were analyzed and classified as malicious or benign. Besides this, the authors highlighted overlooked features that were argued to improve the spam detection.

Another comparative survey was presented by Chakraborty *et al.* [14] in the field of multiplatform spam user detection. The authors recognized that different platforms, such as e-mails, blogs, or microblogs, require different techniques and features to achieve accurate detection. Therefore, proposed techniques within the year range of 2011-2015 were classified based on the platform that the dataset lies within. A qualitative comparison was conducted for each group of methods under the same platform.

Besel *et al.* [21] observed that the botnet used a URL network shortening services and redirections to obfuscate the actual landing pages. They disclosed that users clicked on these URLs, found the botmaster establishing the Bursty botnet, and registering landing pages on phishing websites. They confirmed that the botmaster is still successful in owning Twitter bot-related services. This study includes a review and insight into Twitter's cyberspace infrastructure, cybercrime operation, and the dark markets.

Alothali *et al.* [15] summarized recent research work in the field of Twitter social botnet detection. They provided an analytical review of each proposed method with its limitations and advantages. The techniques were classified into three main categories, namely graph-based, machine learning-based, and crowd sourcing based techniques. The crowd sourcing technique uses human intelligence to identify various patterns, which is stated to be the most error prone out of the three techniques. It was also shown that machine learning methods and, more specifically, random forest classifiers are the most commonly used for detecting social bots in Twitter users.

Latah [16] presented a comprehensive review focusing on malicious social bots' stealthy manner and their detection techniques. The author precisely reviewed detection approaches, which are graph-based, machine learning based, and emerging approaches. Besides, the paper reviewed the strengths and weaknesses of these techniques and the means considered by the bots to avoid detection. Consequently, the paper suggested approaches that may enhance the defense procedures against malicious bots.

Disadvantages

1) .The system doesn’t have technique SHALLOW LEARNING-BASED DETECTION METHODS.

2). There is no technique deep neural networks are applied on Twitter data to determine the relevant content for users, and hence improve their experience on the platform

**PROPOSED SYSTEM**

The proposed model implemented the bidirectional strategy in which tweet sentences are processed both forward and backward for each layer enabling a better understanding of the overall text context. To train the model, a public dataset Cresci-2017 is used that consists of tweets from 3,474 human accounts and 1,455 bots, resulting in 11.4 million tweets in total. Before training, each tweet was preprocessed and tokenized to fit the word embedding model. A pre-trained GloVe model was used to convert text to numerical vectors that are acceptable by the network. The vectors were fed to a three-layer model with a decreasing dropout layer that was initially set to 0.5. Two subsets of testing datasets, composed of 1,982 and 928 accounts respectively, were used to evaluate their model resulting in precision and accuracy of 93% and 95%, respectively.

Mazza *et al.* [26] introduced a novel deep learning model to distinguish bots from humans using their retweet patterns termed RTbust. Before building the model, the authors analyzed the behavioral patterns of bots and humans alike. The analysis demonstrated a distinctive pattern of retweeting in terms of timing, and it was categorized into four patterns. The \_rst is the droplet pattern, which corresponds to normal users in which there exists a fair amount of time between the tweet being posted and the retweet operation. The three remaining patterns belonged to potential bots due to their suspicious and rapid retweeting pattern.

**Advantages**

1. The proposed system offered several innovative approaches that have vastly increased the efficiency of spam identification.
2. The system is more effective due to presence of tweet-based bot detection.

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* **Operating system :** Windows 7 Ultimate.
* **Coding Language :** Python.
* **Front-End :** Python.
* **Back-End :** Django-ORM
* **Designing :** Html, css, javascript.
* **Data Base :** MySQL (WAMP Server).