MALICIOUS POST DETERMINATION WITH TRUST FACTOR IMPLEMENTATION USING MACHINE LEARNING

Shivangi Suresh Ratawa[1], Seema B. Rathod[2]

[1]*Post Graduate Student, Sipna College of Engineering & Technology, Maharashtra, India*
[2]*Professor, Sipna College of Engineering & Technology, Maharashtra, India*

---***---

Abstract: *The popularity of online social media is spread day by day for various online community purposes. Now, it has been found that social media is becoming use as tool for spreading harmful act in sophisticated manner. This thing is also done in web forum, chat room, etc. Some forum are used for a open discussions on a critical topics influenced by radical thoughts. The influential users dominate the mind of naive users using their malicious thoughts. Influential users divert the naive users to do wrong things. The main intension of this topic is to detect whether the post posted by the user is malicious or not and for that the post which is posted by user is checked by text comparison. Each word from the text is checked by using online dictionary Web API service swhether the meaning of the specific word is malicious or not. If the post contain malicious content so the user will be detected and he/she won't be able to post that malicious content. In this way it will predicate the category of the post and help to stop violent data on social media.*

***Key Words*: Malicious, Post detect, Machine learning.**

1. Introduction

Social media is used for spreading most of the terror message and activities through the link and posts. So that it is necessary to detect the user activities and post base on the content provided. We proposed a concept which help us to detect the user and stop spreading such malicious post on social media. Online social media is very popular for different kinds of activities such as online chat etc. There are hundreds of multimedia websites present on internet, these multimedia websites, online chat rooms play serious threats on our society as well as national security. These website provide support psychological war and extension of their agendas, whereas chat rooms and forums encourage their strategies and ideology through interacting with naive users. Many users available on the social media but some users generally avoid going through every comment posted by others. There always exist few users which maintain relationship of trust with other user and their comment attract by other users. These users are called as active users or influential users. These users sometimes called as community leaders. The agenda of influential user is to dominate the naive users to do the frauds as well as wrong things.

Machine Learning Approach

To overcome these problems, most of the researchers started applying machine learning techniques in detecting malicious post on social media. The machine learning can have high demand on Artificial Intelligence (AI). These technique's provide to learn by itself and improve from experience without having any specific program. The main goal of machine learning is to provide ability to the computer to learn automatically without any interference with humans. This technique provide a system to learn by itself and improve from experience without having any specific program.

The machine learning techniques does the following 1.To create a model, the various machine learning algorithm is trained by using a set of training data. 2. Once a new input data is entered into the machine learning algorithm it takes some prediction on the basis of the trained model. 3. The prediction taken in step 2 can be evaluated for checking accuracy. 4. If the estimated accuracy is once tolerable, then the machine learning algorithm is deployed. Otherwise using an enhanced set of training data the machine learning algorithm is again and again trained.

In the same way we are implementing machine learning in our project to detect malicious post on social media.

2. Literature Review

It is becoming increasingly difficult to ignore the importance of using online social networks (OSNs) for various purposes such as marketing, education, entertainment, and business. Online Social Networks also open the door for harmful activities and behaviors. It cause financial fraud and propagating malware and spam advertisements are very common criminal actions that people engage in by accessing uniform resource locators It has been reported that advanced attackers tend to exploit human flaws rather than system flaws thus, users are targeted in social media threats by hour. Fake news can cause lots of issues.

It is becoming increasingly difficult to ignore the importance of using online social networks for various purposes such as marketing, education, entertainment, and business. Online social networks facilitate the way that information is communicated and shared between people, and they have been a tremendously successful route for doing so. For example, Facebook has more than 900 million active users on average, with a 17% increase every year1. Twitter has 320 million monthly active users, 2 and 500 million tweets are posted every day. The more the number of user the more chance of spreading malicious content among users. A report released in 2016 by Proofpoint—a leading information security company—states that advanced attackers tend to exploit human rather than system flaws; thus, people are targeted in social media threats by hour by posting different kind of malicious post. With this huge growth in the amount of online social network data, it is challenging to distinguish malicious links from non-malicious links that use dynamic features that do not evolve over time. This paper will focus on two major aspects: URLs and OSNs.

CLASSIFICATION USING URL FEATURES

This section presents the existing extracted URL features that are related to the host, domain, and lexical characteristics. These features sometime overlap, especially the host and domain features, and also can be used together. This section first describes each feature separately, and then analyses and discusses the drawbacks of these features.

A. Lexical Features

Lexical features reflect some characteristic of a URL as a string; for example, the length of the URL, the length of the host name, and the number of dots present in the URL .Researchers primarily use lexical features to identify websites, blogs, and URLs. One advantage of using lexical features is that the content from the entire web page is not needed in order to analyze it. Therefore, it can be efficiently used in real-time detection. used lexical features for the aforementioned reason to classify phishing URLs. They argued that phishing links tend to have a certain pattern of URL length that differs from legitimate uniform resource locator. The lexical features were used initially by however, added more features extracted from the host name and the URL path. These features are strings, delimited by '/', '?', '.', '=', '-', and '_', to classify the URL. The Markov model used in this study to model these textual properties then different classifying algorithms were used resulting in accuracy of 95%. Very similar work was done by using the same features; however, they included a bigram language model to characterize the host name portion of each URL. As Feroz and Mengel (2014) noted, the key point of using the bigram is that the model has the ability to capture the randomness of the string in a particular URL. This classifier has an accuracy of 97%.

B. Host-based Features

Typically, host-based features are used with lexical features to enhance the detection algorithm and improve the classification accuracy. The classifiers used to distinguish malicious URLs from legitimate URLs are more accurate when the most relevant features are extracted. The host-based features of any URL has rich information about the website that hosts the uniform resource locator, and can be extracted by a simple query known as Whois. This query can provide information about the registrar, and who the registrant is, as well as data about the registration, updates, expiration, and other information. Fette, Sadeh, and Tomasic published a paper in which they described how to detect phishing URLs in an email . They used the IP for a URL, as they assumed that the phisher might store the website on a normal personal computer that did not have domain name system (DNS) entries. They also included the domain age and compared the registration data with the email that was sent. If the elapsed time was less than 60 days, they labelled the email as a phishing email. Additional features were used with the host based features mentioned previously, and this study achieved an accuracy of 99%.

C. Domain-based Features

Domain features and host features can partly overlap since they provide valuable information about the underlying infrastructure of a particular website. Based on the domain information such as IP, domain age and some DNS queries, a wide range of blacklist lookup services can be used to detect malicious URLs. These include Google Safe Browsing, Virus Total, Spamhaus and Web of Trust .Several studies have utilized domain information to detect malicious uniform resource locator used the page rank, domain name, and lexical features as the main features to classify phishing URLs. Page rank is the numeric value, ranging from 0 to 10, which determines the importance of a given web page in relation to other web pages. Based on this ranking they argued that phishing pages have short life spans and thus have a lower page rank. They used a logistic regression classifier and achieved an accuracy of 97.3%.

D. Issues Related to URL Features

All the URL features introduced in this section can be used in a classification algorithm either separately or in combination with one another. Although using URL features has been shown to result in a high percentage of overall accuracy, attackers use different evasion techniques, making it useless to detect URLs based on existing features.

Facebook Apps: The facebook app is used by millions of users . The facebook app is an online social network where users can easily post malicious data. Facebook enables third-party developers to offer services to its users by means of Facebook applications. Unlike typical desktop and smart phone applications, installation of a Facebook application by a user does not involve the user downloading and executing an application binary. Instead, when a user adds a

Facebook application to her profile, the user grants the application server: 1) permission to access a subset of the information listed on the user's Facebook profile (e.g., the user's e-mail address), and 2) permission to perform certain actions on behalf of the user (e.g., the ability to post on the user's wall). Facebook grants these permissions to any application by handing an OAuth 2.0 [17] token to the application server for each user who installs the application. Thereafter, the application can access

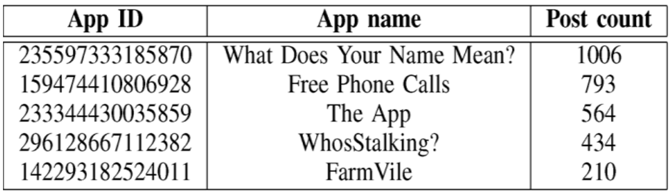
The data and perform the explicitly permitted actions on behalf *Operation of Malicious Applications:* Malicious Facebook applications typically operate as follows.

- Step 1: Hackers convince users to install the app, usually with some fake promise.

- Step 2: Once a user installs the app, it redirects the user to a Web page where the user is requested to perform tasks, such as completing a survey, again with the lure of fake rewards.

- Step 3: The app thereafter accesses personal information from the user's profile, which the hackers can potentially use to profit.

- Step 4: The app makes malicious posts on behalf of the user to lure the user's friends to install the same app



App ID	App name	Post count
235597333185870	What Does Your Name Mean?	1006
159474410806928	Free Phone Calls	793
233344430035859	The App	564
296128667112382	WhosStalking?	434
142293182524011	FarmVile	210

TABLE 1

The top five malicious applications, in terms of number of posts per application. The malicious post can create lots of misunderstanding. Although we infer the ground truth data about malicious applications from MyPage- Keeper, it is possible that MyPageKeeper itself has potential bias classifying malicious app's posts. For example, if a malicious application is very unpopular and therefore does not appear in many users' walls or news feeds, MyPageKeeper may fail to classify it as malicious (since it works on post level).

Detecting Spam on OSNs*:* Gao *et al.* analyzed posts on the walls of 3.5 million Face book users and showed that 10% of links posted on Face book walls are spam. They also presented techniques to identify compromised accounts and spam campaigns. In other work, Gao *et al.* and Rahman *et al.* develop differntt techniques for online spam filtering on OSNs such as Facebook. While Gao *et al*rely on having the whole social graph as input, and so is usable only by the OSN provider, Rahman *et al.* develop a third-party application for spam detection on Facebook. Others resent mechanisms for detection of spam URLs on Twitter. In contrast to all of these efforts, rather than classifying individual URLs or posts as spam, we focus on identifying malicious post that are the main source of spam on Facebook.

Detecting Spam Accounts*:* Yang *et al.* and Benevenuto *et al.* developed techniques to identify accounts of spammers on Twitter. Others have proposed a honey-pot-based approach to detect spam accounts on OSNs. Yardi analyzed behavioral an patterns among spam accounts in Twitter. Instead of focusing on accounts created by spammers, our work enables detection of malicious

Post posted by the user that propagate spam and malware by luring normal users to install them.

App Permission Exploitation*:* Chia *et al.* investigate risk signaling on the privacy intrusiveness of Facebook apps and conclude that current forms of community ratings are not reliable indicators of the privacy risks associated with app. Also, in keeping with our observation, they found that popular Facebook apps tend to request more permission's. To address privacy risks for using Facebook apps, some studies propose a new application policy and authentication dialog. Makridakis *et al.* use a real application named "Photo of the Day" to demonstrate how malicious content on Facebook can launch distributed denial-of-service (DDoS) attacks using the Facebook platform. King *et al.* conducted a survey to understand users' interaction with Facebook apps. Similarly, Gjoka *et al*study the user reach of popular Facebook applications. On the contrary, we quantify the prevalence of malicious apps and develop tools to identify malicious apps that use several features beyond the required permission set.

3. Problem Analysis

Malicious post from different users are getting posted on our social media platform. Any personal comments or religious comments are also affecting seriously on the overall media. To overcome this issue we are using online dictionary Web API Service of words. We are categorizing the collected data set so that we can efficiently compare it with users input. Our challenge is to set the trust factor for the user malicious posts. Here we are using machine learning extensively. To set that trust factor we are using Pattern Matching Algorithm which will calculate the frequency and occurrence of the malicious word coming in the post..

They have founded that a fraction of tweets particularly of some hot topics contains more number of URLs then other. This clearly highlights to what extend the attackers has been using. Though the research has came up the most efficient popular blacklisting in detecting spam, Using pattern matching algorithm the data posted by user will be compare to the online set(dictionary) web API service if there is any malicious content like any abusive, sexual, offensive or violent content in the text the count is calculated and the malicious percentage is taken out. The user once detected he/she won't be able to put malicious data on social media. By using this algorithm each and every word in the post will be compare and checked whether its malicious or not and then only the user will be able to post. The total percentage of the text will also be calculated. This will let us know the post containing the total percentage of malicious words used in the text. The words extracted from the text firstly will compare them with the words list present in database. As once the word is checked it wont get checked again and again it will get store in the database. If the words is not available in database than it will check and compare the word meaning in the online dictionary web API service that will detect whether the word is malicious or non malicious.

4. Propose Work

When the user wants to post any data it will be taken as input and according to our application the user should not be able to post the malicious data. The user input will be extracted and will split into single words and will be checked by using online dictionary Web API whether that input posted by user comes under malicious or not. As the process of checking of words will be taken place every time so we had made a database where the repeated words won't be checked again and again. If the words are not present in the database the words will be checked and verify from the online dictionary Web Api. If the words in the text is non malicious the post will be posted successfully. If the post contains malicious contents than the user will not be able to post that malicious post the detection is done in the same way as given in the below application diagram

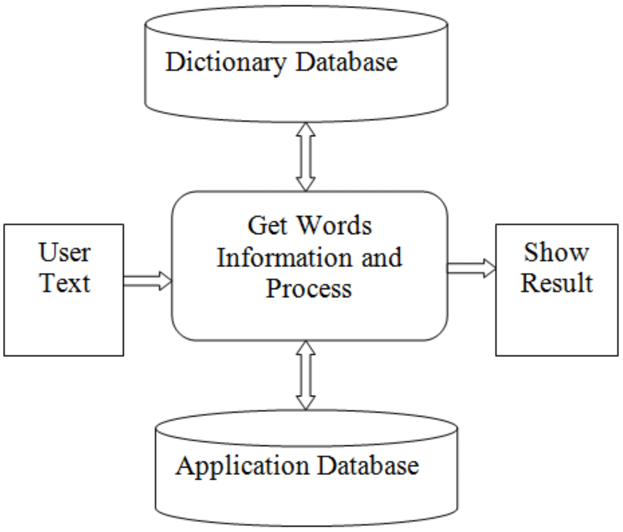


Fig:-Application Flow

A) User's Text : It is the text that will be considered as an input for our application. It can contain anything. We need to bifurcate and make the list of words and then process it.

B) Dictionary Database: In our application, we have used dictionary service for detecting context of the words that are being provided by a user.

C) App's Database: In this application, User's database is also defined to store the already processed words. Whenever this request to dictionary service. If dictionary service is user, it gets the data and application receives any text, it separates and extract all the words and then check first in application's database because it is not necessary to always use the dictionary service. If that data is not found in application's database, system sends a stores that info in application's database, so next time system won't send a request for same kind of words.

D) Show Result : This is the final step, in which application show its output. User's text contains any malicious thing, number of offensive/malicious/vulgar words, percentage, etc.

5. System Design

The user will have to register first than he can login the as his account is created. He can post or write any post once he get logged in. The post than will be detected before it get post whether its malicious post that is it contains any kind of offensive or abusive statements or whether it comes user non malicious.

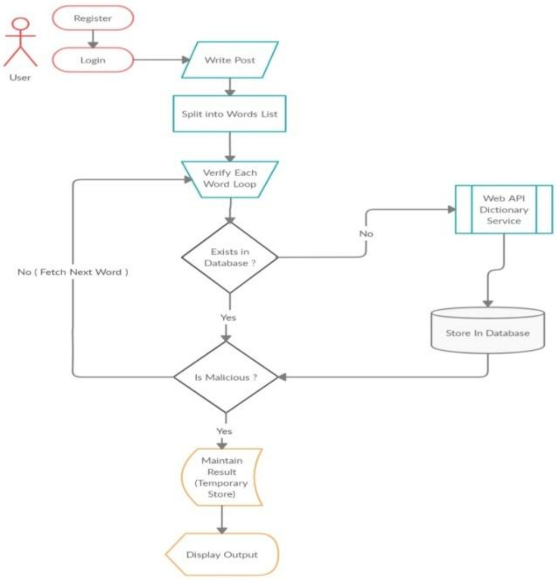


Fig:-Architecture Diagram

In the figure above the user first needs to create his account so he needs to do registration so that he can login to post and use the particular application User's Text will be used to extract the list of words .Each word from extracted word's list will be checked and processed from dictionary service and application database. Processing word will be compared with application's database words information .If processing word if found in application's database, it will maintain that result and switch to next word to process. If processing word if not found in application's database, it will try to fetch word's meaning from online dictionary using Web API service. In received information from dictionary's Web API, system will extract it context and map it with application's database and store it. So, it can be used for future reference. As the word's info is fetched and stored in application's database, application won't try to fetched that word's information from dictionary service. It will save web service calling efforts for the application.

6. Implementation

When user's inserting or posting the statement that time our system providing the check post will identifying that inserted post is statement or link non malicious or malicious. After identifying the things the system forwarding control to next level and step wise system will be executed. Following steps involved for the flow of the process:-

1) User's Text will be used to extract the list of words.

2) Each word from extract word's list will be checked and processed from dictionary service and application database.

3) Processing word will be compared with application's database words info.

4) If processing word if found in application's database, it will maintain that result and switch to next word to process.

5) If processing word if not found in application's database, it will try to fetch word's info from dictionary service using Web API.

6) In received info from dictionary's Web API, system will extract it context and map it with application's database and store it. So, it can be used for future reference.

As the word's info is fetched and stored in application's database, application won't try to fetch that word's information from dictionary service. It will save web service calling efforts for the application. The system also having the provision to collect data in to back end like storing the blacklisted links into the data base, so that in future if any link is found as same as which is having the similar characteristics of the link, it will be identified immediately and also it will be very helpful to reduced the complexity and increasing the accuracy of the system to detect the malicious post of the system.

In this we are detecting the malicious word from the text and the total percentage value is taken out. The percentage of the text is calculated by the number of malicious words posted in the text by the user. The more the malicious the words in the text the more will be the percentage. The total malicious percentage is taken out as the number of malicious divided by the total number of words in the text multiply by 100.

The simple formula to calculate the malicious content in the post is given below:-

Malicious Number of malicious words in the text

Count = *100 ————————————————————

Total number of words in the text

By using the above formula we can calculate the total percentage of malicious content in the text. The total percentage will detect the amount of malicious content in the post and he won't be able to post the text.

7. Result Analysis

It is observed that the user posting malicious text is being detected and below is the table showing the malicious words used by user and the total percentage that how much percent of data is malicious in the text posted by the user.

The user posting the text the text is being extracted and being checked that it is malicious or non malicious. If the post does not contain any malicious content in it the text will get posted directly. If the malicious content is found in the text the user will be detected and won't be able to post any kind of malicious data. We are using online web API dictionary service to check the text that the words used in the text is malicious or not. The malicious post posted contains asterisk symbol in it. Many users use asterisk while posting some kind of abusive words. We made our application in such a way that it should detect malicious words while its written in simple way or by adding asterisk

Sr. No	Text	Is Malicious ?	Malicious Percentage	Malicious/Offensive Words
1	Hello Everybody! I am using this new web application	No	0	
2	You Idiot! This application is under progress. Use it carefully.	No	0	
3	You bastard! How dare you to talk to me like this ?	Yes	9.09	bastard!
4	You bloody bast*ard!	Yes	66	Bloody, bast*ard!
5	You fuck*ing bast*ard!	Yes	66	fuck*ing, bast*ard!
6	Ok James! Leave it, we should not fight over here!	No	0	
7	Yeah Ronnie! That will be better. Otherwise, We will be behind bars due to this web application! :D	No	0	

TABLE :- Result Analysis

8. Conclusion

In this project malicious post determination with trust factor implementation using machine learning we have learned how to detect the malicious post. The user's post will be checked before posting. We have implemented machine learning as machine learning is nothing but our machine has learned the process of how the malicious post is detected and how the total percent of malicious content is posted by user. User is not able to post when the post is detected as malicious. This leads to increase the trust factor of the user on the application he/she is using.

9. References

[1] H. Gao, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," Internet Measurement Conference, November 1–3, 2010, Melbourne, Australia, pp 35-87.

[2] Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," In Proceedings of the International World Wide Web Conference (WWW), Banff, Alberta, Canada, May 2007.

[3] BENEVENUTO, F, RODRIGUES T, AND ALMEIDA V, "Detecting spammers and content promoters in online video social networks", In Proc. of Special Interest Group On Information Retrieval (Boston, Massachusetts, USA, July 2009).

[4] S. Lee and J. Kim, "Warningbird: Detecting suspicious urls in twitter stream," In NDSS, 2012.

[5] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Designand Evaluation of a Real-Time URL Spam Filtering Service," In Proceedings of the IEEE Symposium on Security and Privacy, 2011.

[6] C. Yang, R. Harkreader, "Die free or live hard? Empirical evaluation and new design for fighting evolving twitter spammers," In RAID, 2011.

[7] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," In CEAS, 2010.

[8] K. Lee, J. Caverlee, and S. Webb,"Uncovering social spammers: social honeypots + machine learning," In SIGIR, 2010.

[9] S. Webb, J. Caverlee, , and C.Pu," Social honeypots: Making friends with a spammer near you", In Conference on Email and Anti-Spam (CEAS 2008),2008.

[10] S. Yardi, D. Romero, G. Schoenebeck et al,"Detecting spam in a twitter network," First Monday, 2009.

[11] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove," Analyzing facebook privacy settings: user expectations vs. reality," In IMC, 2011.

[12]M. Gjoka, M. Sirivianos, A. Markopoulou, and X. Yang, "Characterization Of Osn applications," In Proceedings of the first workshop on Online social networks,WOSN, 2008.

[13] T. Stein, E. Chen, and K. Mangla," Facebook immune system," In Proceedings of the 4th Workshop on Social Network Systems, 2011.

[14] S. Yardi, D. Romero, G. Schoenebeck et al," Detecting spam in a twitter network", First Monday, 2009.