Malicious Bot Detection

I.V.S.L Haritha[4]
Department of InformationTechnologyMLR Institute of Technology Hyderabad, India

Katamreddy Preethi
Department of Information Technology
MLR Institute of Technology
Hyderabad, India

Goli Thanmai
Department of Information Technology*MLR Institute of Technology* Hyderabad, India

Vennapureddy Sathwik
Department of Information Technology
MLR Institute of Technology
Hyderabad, India

Akash Reddy Aenugu
Department of Information Technology
MLR Institute of Technology
Hyderabad, India

Abstract— **Phishing attempts launched by rogue social bots have become a big worry for social media sites like Twitter. In order to trick people into clicking on their fake tweets and rerouting them to dangerous websites, these bots construct bogus tweets with unsafe URLs. Phishing is a form of online social engineering used to deceive people into disclosing their personal information, including email addresses, passwords, and credit card pins. Malware might potentially be installed on the victim's machine in some circumstances. A crucial task for the security of the Twitter network is the detection of harmful bots. The suggested approach to identifying these bots entails gathering data on the URLs shared in tweets, including URL redirection, link-sharing frequency, and the presence of spam. The behaviour of Twitter users is then assessed using this data, and harmful and honest tweets are separated using this data. The LA-based malicious social bot detection (LA-MSBD) technique was created in order to identify dangerous social bots and stop malicious attacks. This method efficiently detects malicious bots and stops assaults by fusing a trust computational model with URL-based indicators. As a result, user data security is seriously threatened by the increase in phishing attacks carried out by hostile social bots on social media platforms like Twitter. The suggested LA-MSBD technique offers a practical way to recognize and stop these assaults, protecting Twitter network users' online safety.**

Keywords- Malicious bot, Phishing attacks, URL redirection, Learning automata.

I. Introduction

Malicious bots are automated software program designed to perform malicious activities on the internet, such as spreading malware, stealing information, and spamming. These bots can be created by hackers or cybercriminals and can be programmed to mimic human behavior making it difficult for security systems to detect them.

Malicious bots can distribute fake news, sway people's opinions, and trick social media algorithms to promote particular material or agendas in the context of social media. For example, bots can be used to spread disinformation, influencing public opinion.

Detecting malicious bots posting URLs in the twitter network is extremely important for several reasons. First, malicious bots can use URLs to spread malware, phishing scams, and other types of malicious content. This can harm users devices, compromise their personal information, and lead to financial losses.

Second, these bots can use URLs to spread fake news, propaganda, and disinformation, which can have serious consequences on public opinion and democratic processes. For example, bots can be used to spread fake information about political candidates or issues, influencing public opinion and potentially affecting election outcomes.

Third, malicious bots can use URLs to manipulate social media algorithms, creating an unfair advantage for certain accounts or content. This can harm the integrity of the network and erode users trust in the platform.

Therefore, detecting malicious bots posting URLs in the networks like twitter is essential for maintaining the security, integrity, and trustworthiness of the platform. It can help prevent the spread of malicious content, ensure fair and democratic processes, and protect users privacy and security

II. Literature survey

In Evidence Fusion for Malicious Bot Detection in IoT paper, an evidence theory-based is proposed by Akbar Siami Namin et al for detecting malicious bots in IoT. The Dempster Shafer Theory (DST) is a probabilistic reasoning technique that excels at handling uncertainty, i.e. when there is no supporting data. It can be used effectively to locate a bot, especially when the behavior of the bots is dynamic or polymorphic. Key evidence for bot trails is extracted through the analysis of network flow characteristics. These facts are evaluated using the apriori process. DST is then used to identify any bots that may be present. To apply the apriori technique to the ISOT HTTP botnet dataset, it is transformed into a transaction dataset. For DST-based, this technique displayed an accuracy of roughly 87.73%.

The Decision Tree (DT) classifier was proven to be a very effective technique for identifying P2P botnets in the study

"Botnet Detection using Machine Learning" by Haq, Singh, et al. A dataset of 38,000 recordings of network activity, including both legal and attack data, was used by the study team to evaluate several classifiers and clustering approaches. Among the classifiers evaluated, the DT classifier exhibited the highest accuracy rate of 87.785% in detecting botnets, indicating its potential suitability for botnet identification.

In 2013, Feizollah et al. conducted a study on the identification of mobile botnets in Android malware by utilizing machine learning classifiers. The study used data sets from the Android Malware Genome Project and evaluated three classifiers: K-NN, DT, and SVM. To detect bots, the study considered three network characteristics: connection time, TCP size, and total number of GET/POST arguments. The results indicated that the KNN classifier had the highest accuracy in identifying specific botnets, with a true positive rate of up to 97.94% and a low false positive rate of just 0.06%. The accuracy of the DT classifier was comparable to that of the KNN. Overall, the study demonstrated the effectiveness of machine learning classifiers in detecting mobile botnets.
.

The researchers conducted a study on detecting HTTP botnets using machine learning algorithms. They analyzed the TCP packet characteristics of network traffic to distinguish between valid and malicious communications. They used classifiers such as DT, NB, KNN, and RF to assess the accuracy of the method against five HTTP botnets. The results showed that the random forest classifier was most effective, with detection rates above 90% for Dorkbot, Zeus, SpyEye, and Cutwall. However, the false positive rate was also high, meaning it might raise false alarms. The study highlights the potential of machine learning for detecting HTTP botnets and the importance of evaluating the effectiveness of different classifiers.

Guerra-Manzanares and his team proposed a study to detect botnets in IoT networks using machine learning. The study used a large dataset with 115 features and utilized filter, wrapper, and hybrid models to reduce the number of features to improve the accuracy of KNN and Random Forest models. The hybrid model combines the advantages of filter and wrapper models to reduce the number of features while preserving high accuracy levels. The study reported that using the hybrid model approach led to an accuracy level of 97%, indicating the potential of hybrid feature selection techniques for improving botnet detection in IoT networks.

In the research paper "Botnet Detection based on Machine Learning Techniques using DNS Query Data" by Hoang et al., a machine learning-based approach is suggested to identify botnets using DNS query data. The study compares the performance of several classifiers, such as K-NN, C4.5, random forests (RF), and NB, for botnet identification. The results indicate that RF is the most effective classifier, with an accuracy rate of 90%. The researchers note that using DNS data for botnet detection can be a promising strategy as it outperforms anomaly-based botnet detection algorithms. Overall, the study highlights the potential of machine learning in identifying botnets using DNS data.

Garg et al. conducted a study on identifying P2P botnets by analyzing network traffic patterns using machine learning algorithms. They selected key network traffic components and generated a vast number of test cases by combining them in different ways. The study compared the performance of three algorithms, namely NB, IBk, and J48, in detecting P2P botnets. The results revealed that both J48 and IBk algorithms outperformed NB in identifying P2P botnets, despite J48's long training period and IBk's long testing period. Both J48 and IBk algorithms showed a detection accuracy of above 95%, indicating the potential of machine learning-based techniques for botnet identification. The study emphasizes the importance of carefully selecting and evaluating machine learning algorithms for effective botnet detection.

III. Existing System

The current method of identifying malicious URLs is based on lexical and DNS characteristics. The drawback of this system is that the bots can use URL redirections in order to avoid getting detected by DNS properties.

Moreover, the malicious URLs are not directly posted in tweets as they are shortened and modified using advanced tools, which makes it difficult to detect them using lexical properties.

Other than above mentioned software-based approach, anti-phishing methods focused on hardware are currently in use. However, software-based methods are preferred due to cost and operational considerations.

These studies have demonstrated that machine learning is very helpful and effective at identifying botnets. Our research showed that, for a number of reasons, certain tactics are effective while others are not. Many algorithms, including DST, were useful in our research. Therefore, we made the decision to take this technique into account together with another approach like Bayesian Learning.

We employ all of the techniques listed above and determine the effective approach that produces the most accurate output for our model.

Disadvantages:

The capacity of malicious social bots to manipulate several elements of the content of their tweets, such as emoticons, emotive terms, and widely used words, makes it challenging to identify them only through statistical data.

On Twitter, however, user connections are generally solid and hard to manipulate by automated accounts.

The user's social interactions are not taken into account by the detecting techniques currently in use. Identification of temporal data trends in noisy data gets more difficult as rogue bots adapt their behavior to escape detection.

The ability of these algorithms to identify these bots may therefore not be particularly reliable.

IV. PROPOSED SYSTEM

An innovative method to recognize and stop damaging social bot attacks is the LA-based malicious social bot detection (LA-MSBD) technique. To find dangerous social bots, the approach blends a computational trust model with URL-based indicators. To evaluate a participant's behavior, the suggested method takes into account a number of URL-based characteristics, such as URL redirection, spam content in the URL, the frequency of shared URLs, and the relative position of the URL. Using Bayesian learning and the Dempster-Shafer theory, the veracity of tweets is assessed. The LA-MSBD approach offers a thorough framework for identifying harmful social bots by real-time analyzing user behavior and URLs, which can assist stop social engineering assaults on social media sites.

The project's goal is to create a machine learning solution for identifying harmful bots by first utilizing learning automata to identify malicious URLs on networks like Twitter. The project aims to collect and process data, design and implement a learning automata algorithm, evaluate algorithm performance, and improve the security of the twitter like networks. The ultimate objective is to support the creation of efficient protection against the danger posed by harmful URLs on social media platforms.

The scope of this project is to protect users from online threats such as phishing attacks, malware downloads, account takeovers, credential stuffing, and web scrapping. It is possible to train machine learning models to recognize the patterns and traits of known harmful URLs and bots as well as unexpected threats. This project has the potential to improve online security and protect users from a wide range of evolving threats

Advantages:

The suggested technique successfully and accurately detects dangerous social bots. It performs better in terms of precision than other well-known machine learning techniques, according to experimental findings.

In order to enable incremental learning and improve performance, the system is designed to update its action probability value through a limited number of learning sessions. This strategy makes the system a beneficial tool for combating hazardous social bot assaults by ensuring that it is always up to speed with the most recent botnet behaviors and is better equipped to recognize dangerous social bots.

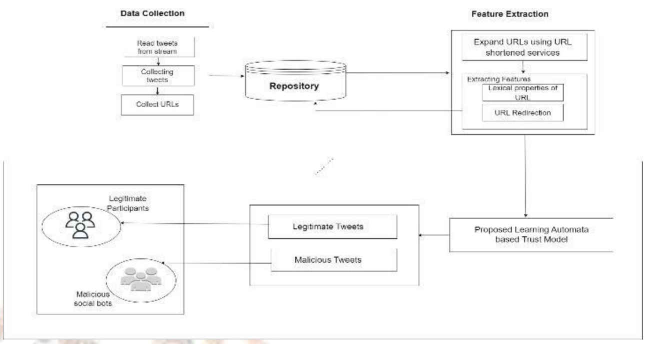

Fig-4.1: System architecture

V. MODULES

In this project we have identified the following modules:

- ❖ Admin Module
- ❖ User module
- ❖ Data Collection
- ❖ URL Extraction

A. *Admin Module :*

The primary super user of the system is admin. After logging in, the administrator will train the system and review all of the participants' malicious tweets. Once predicted, an administrator will put up a user interface for dangerous bot prediction. The harmful social bots list is also visible to the administrator, who can prohibit them for user security.

B. *User Module:*

Our system's end user is called user. Users can login and browse tweets and post tweets with URLs after creating an account. The system will detect malicious tweets by examining the URLs in the tweets, while the user can view the authentic tweets produced by the legitimate individuals.

C. *Data Collection:*

Data collection is where tweets are collected to analyse whether the URLs are malicious or not. The three subphases that make up the data collection phase include reading tweets from the system, gathering tweets, and finally gathering URLs. Moreover, a repository is used to store the gathered URLs and tweets.

D. URL Extraction:

URL extraction is the process where the software identifies the existence of a URL and extracts that URL in order to perform prediction operation to check if the URL is legitimate or not

VI. LA-MSBD ALOGRITHM

In addition to a trust model that assesses each user's trustworthiness based on URL-related factors, the article suggests a framework for analyzing the tweets of Twitter users. It is suggested to use the LA-MSBD approach to identify potentially harmful social bots. The LA model has a trust-measuring concept that scans tweets for potentially dangerous material, like URLs containing spam. A participant's series of tweets is scrutinized for malevolent intent and categorized as real or harmful. Malicious social bots are easier to spot, though, as they are more likely to publish objectionable content. The suggested approach uses a trust model to identify harmful social bots than existing machine learning approaches. The approach also employs incremental learning to raise the detection rate and update the action probability value.

Algorithm 1 *Feature Ranking*

```
1: for every features fi ∈ F do
2: Calculate weight wt fi = G( fi)/ $\sum_{i=1}^{n} G(fi)$
3: end for
4: $\vec{F}$→ Create a feature ranking vector with weights $wt_{fi}$ corresponding to each crucial feature fi
```

To find the most crucial characteristics in a dataset, algorithm 1 uses a weight function and a feature ranking algorithm. Using Shannon's entropy model, the algorithm determines the information gain value (G(fi)) for each feature. Then, each feature is given a weight (wt fi) based on the value of its information gain using the weight function. To determine the ~~relative importance of each trait to the others, the weights~~ are normalized. The features are then ordered according to their weights, with the more significant elements being labeled as important and the less significant features as less significant. This approach can increase the precision and effectiveness of machine learning models used for data analysis by identifying the crucial features that contribute the most to the dataset.

Algorithm 2 *Direct Trust Computation*

```
1: T D(t) = φ
2: for every tweet $t\, w_{ij}$ j=1 to m do
3: if the participant $p_i$ has shared jth tweet tw to one of its nearby participant (friend) then
4: F ← Retrieve feature set of $t\, w_{ij}$
5: $\vec{F}$ ← Feature_ranking(F)
6: for every feature $\vec{f}i \in \vec{F}$ do
7: Calculate pr( $f_1$, …, fn |C = malicious) using Eq (4)
8: end for
9: Calculate $T_{twij}$ (t) with help of Eq (5)
10: $T_{twij}$ (t) is concatenated with $T^D$(t), and $T^D$(t) is updated using the concatenated data
11: end if
12: end for
13: Calculate direct_trust $T^D_{p_i}$ (t) using Eq (6)
```

In the LA-MSBD approach, a set of attributes are taken from each tweet that the participant posts at time t in order to calculate direct trust. Based on the weights given to each feature, these characteristics are used to generate a feature ranking vector. After determining the possibility that a feature is in the hazardous category, the distrust value of a tweet is calculated by multiplying this likelihood by the user's trustworthiness score. Using equation (5), the ultimate trust value of a tweet is calculated. The participant's overall trust value is calculated by carrying out this process for each tweet in their timeline. Using this method, it is possible to identify dangerous social bots by identifying tweets with low trust levels, which indicate potentially destructive material.

Algorithm 2 combines a set of direct trust values for the participant at that particular time with the trust value of each tweet. The trust value of each tweet is concatenated to the original empty value of this set (Line 1) to update it (Line 10). Using equation (6), the participant's final direct trust value is calculated at that moment by taking into account all of the trust values of their tweets in that set. This process is repeated for all of the participant's tweets.

Algorithm 3 Indirect Trust Computation

```
1: T I D(t) = ϕ
2: if participant pi has one or more participants inside a single hop, then
3: for every pk ∈ N B(pi) do// N B(pi)-neighbors of pi
4: $T^D_{pk}$(t) ← Direct_Trust_Computation(pk)
5: Concatenation of $T^{ID}$ (t) with value $T^D_{pk}$ (t) and $T^D$ (t) is updated with the concatenated values
6: end for
```

```
Calculate indirect_trust $T_{pi}^{ID}$(t) by using Eq (11)
else
    $T_{pi}^{ID}(t) = 0$
end if
```

A participant's one-hop neighbors' degrees of belief are used to determine indirect trust. Direct trust is helpful in determining a participant's dependability, but taking into account the participants' beliefs about them can reveal more. Dempster's weighted combination rules are used to combine a participant's direct trust values with the opinions of their neighbors to arrive at the indirect trust value. It is believed that the participant's belief value in the network at time t is represented by the resulting indirect trust value.

Algorithm 4 *LA-MSBD*

Input:
Participants in the set P = {p1,..., pn} in Twitter, τ : Time slots available, Tf : Threshold value, ϵ: Reward parameter
Output:
T: a list of all legitimate participants and a set of trust values for those participants, Sb: a collection of harmful social bots
Assumptions:
Let LA be set to $L\ A\ = \{la1, la2, ...lan\}$, where lai denotes the learning automata for every participant.

```
begin
$S_b$ = φ, β = φ, T = φ
For each participant, a learning automaton is initiated by pi
pi ∈ P do for every participant
    for t = 1, 2,...,τ do
        $T_{pi}^{D}$ ← Direct_Trust_Computation()
        $T_{pi}^{ID}$ ← Indirect_Trust_Computation()
        Calculate trust value of pi ($T_{pi}$ (t)) using Eq (1)
        Calculate action probability value pr(t) = 1 − $T_{pi}$(t)
        if $T_{pi}$(t) < $T_f$ then
            Values from the concatenation of the set with the string 1 and β are updated.
        else
            Values from the concatenation of the set with the strings 0 and β are updated.
        end if
        Calculate $pr(t + 1)$ using Eq (12)
    end for
    if (no. of 1's in β > no. of 0's in β) then
        $S_b$= $S_b$ ∪ {$pi$} // $pi$-harmful social bot
        reward $pi$ using $T_{pi}(t)$ = $T_{pi}(t)$ − ε
    else
        $pi$ is legitimate and added into the legitimate list of participants.
        Concatenation of set $T$ with the value $T_{pi}(t)$ and $T$ is updated with the concatenated values
    end if
    β = φ
end for
return $T$ with list of legitimate participants and $S_b$
```

VII. CONCLUSION

In order to identify harmful social bots, our project suggests the LA-MSBD technique, which combines a trust computational model and URL-based features. For network security, identifying bots is essential, and a number of techniques have been developed to monitor and catch bot activity.

Our strategy has been evaluated and, in terms of accuracy, outperforms other current algorithms by up to 3%. Our method offers a more reliable and accurate way to identify harmful social bots in online social networks by utilizing a trust model and examining URL-based attributes.

VIII. FUTURE ENHANCEMENTS

Information technology needs to react to the quick changes in the market, thus security is crucial. One of the most significant threats to the new distributed computing models on the web, botnets function like a distributed network.

In the upcoming years, this application can be improved further and uploaded to real-time servers so that anyone in the world can use a web browser to access it. After scaling up the application many new features can be added such as alert displaying the type of phishing attack, automatic reporting of the identified malicious user to twitter.

.

IX. ACKNOWLEDGEMENT

This project was permitted and encouraged by our Institution, MLR Institute of Technology and our project guide Mrs. I.V.Sai Lakshmi Haritha. I thank all the people responsible for the same. I further thank our guide, I.V.S.L Haritha who provided insight that greatly assisted the research.

X. REFERENCES

[1] 2018 IEEE International Conference on BigData, 10-13 December 2018, Moitrayee Chatterjee, Akbar Siami Namin, Prerit Datta

[2] 2018,5th IEE International Conference on Parallel,Distributed and Grid Computing (PDGC-2018), 20-22 December 2018 - Shamsul Haq and Yashwant Singh

[3] 2013,Malaysian Journal of Computer Science– Ali Feizollah, Nor Badrul Anuar, Rosli Salleh, Fairuz Amalina, Rauf Ridzuan Ma'arof, Shahaboddin Shamshirband

[4] Journal of Telecommuncation, Electronic and Computer Engineering – Rudy Fadhlee Mohd Dollah, Faizal M.A., Fahmi Arif and Lee Kher Xin

[5] 2019,International Conference on Cyberworlds (CW), Kyoto, Japan, ppp. 324-327, 2019 - A. Guerra-Manzanares, H. Bahsi and S. Nõmm

[6] Future Internet 2018 – Xuan Dau Hoang and Quynh Chi Nguyen

[7] 2013 IEEE Department of Computer Science and Engineering – Shree Garg, Anukush K. Singh, Anil K. Sarjee and Sateesh K.Peddoju.