R. Jaiganesh, 2025, 13:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Fake Account Identification on Social Network Sites Using Machine Learning

R. Jaiganesh, Professor V. Sumalatha
Department of Computer Applications-pg
VISTAS

Abstract- At present social network sites are part of the life for most of the people and want to free-frank with new friends. Every day several people are creating their profiles on the social network platforms and they are interacting with others independent of the user's location and time, Technology is associated with online social networks which has become a part in every one's life in making new friends and keeping friends, and share a personal information with other users their interests are known easier. Traditional methods cannot differentiate between real and fake accounts efficiently. To analyze, who are encouraging threats in social networking sites with user profiles. There are numerous cases where produced accounts have been effectively distinguished utilizing machine adapting techniques characters made by people.

Keywords Social Media Security, Machine Learning.

I. INTRODUCTION

Fake accounts are now a major concern due to the social networking sites' explosive expansion. These accounts are frequently used for phishing, spamming, disseminating false information, and influencing public opinion. To guarantee a secure and reliable online environment, it is essential to recognize and address such risks.

The goal of this research is to employ machine learning techniques to detect phony accounts on social networking sites. The algorithm is trained to differentiate between real and fraudulent users by examining a variety of user behaviors and profile attributes, including account activity, content patterns, and link networks. To identify irregularities and trends suggestive of fraudulent activity, the system makes use of supervised learning techniques. By offering a reliable, scalable, and automated method for detecting phony accounts, the initiative hopes to improve the integrity of social media sites.

II. LITERATURE SURVEY

In order to differentiate between spammers and authentic users, Yang and colleagues examined Twitter users' behavioral patterns. Features including tweet frequency, follower/follower ratio, and content characteristics were their main focus. The use of feature-based classification for the detection of fraudulent accounts was made possible by their study.[1]. In order to identify spam profiles on social media sites like Facebook and Twitter, this study suggested a machine learning-based method that makes use of supervised algorithms like Random Forests and SVM. It underlined how crucial it is to use chat activity and friend request trends to spot unscrupulous people [2]. The researchers developed a classifier combining behavioral and profile-based features using actual data from social media platforms. Their model demonstrated the significance of temporal data, including posting frequency and session duration, and obtained high accuracy[3]. The idea of "Sybil accounts" was introduced in this work, along with the SybilRank

algorithm, which assesses accounts' trust scores according on their graph structure. The study showed that network-based features could be a useful tool for identifying fraudulent profiles [4]. Markov Clustering (MCL) was used by the researchers to identify spammer communities. They demonstrated that unsupervised learning might potentially be helpful in detecting fraudulent accounts with less labeled data by examining links and grouping related accounts [5]. This study investigated deep behavioral analytics through topic modeling, content sentiment analysis, and user interaction style tracking. It emphasized how useful psychological profiling and NLP techniques are for improving detection accuracy [6]. This study presented Botometer, a program that uses sentiment analysis, language signals, and temporal activity to assess Twitter accounts and determine if they are likely to be bots. The intricacy of hybrid (semi-human) false accounts was also highlighted [7]. Wang used feature engineering based on friend diversity, URL usage, and interaction count to show how well decision trees and logistic regression work in separating phony accounts from authentic ones [8]. By looking at timing patterns, device metadata, and retweet behavior, the authors divided Twitter users into three groups: humans, bots, and cyborgs. Their strategy demonstrated how machine learning-based detection techniques could be improved by time-series analysis [9]. This study examined a popular false account technique called link farming, which involves mass following and unfollowing in order to acquire followers. They identified abnormal growth patterns using anomaly detection and graph mining, and they suggested countermeasures[10].

III. IMPLEMENTATION

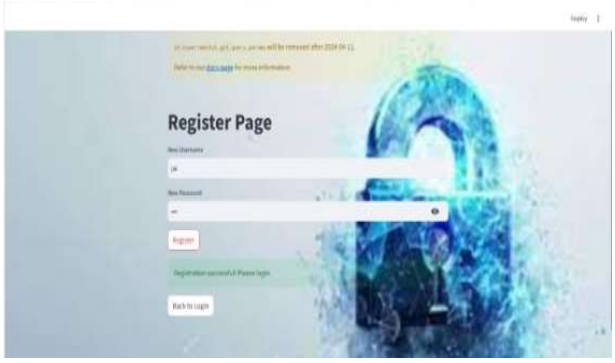

Fig 1: Register Page

Interface for User Registration: The registration page is where a new user can create a password and username.

Elements of the User Interface: Page Title: The "Register Page" noticeably indicates that the user is in the app's registration area.

Fields on the Form:

New Username: The username that Jai entered is displayed in the text field. New Password: Three hidden characters (seen by dots) in the password input box imply some sort of password concealing or input validation. Toggle password visibility with an eye symbol next to the password field (a popular UX feature for security). Buttons: To submit the registration form, click the "Register" button (red border). Return to Login (white button): Returns to the login screen. Message of Success: The message "Registration successful!" appears in a green notification window. Kindly log in.

This attests to the user's successful registration. Environment and Browser Details: Localhost:8501/?page=register is the URL. The URL shows that port 8501 is the local host for the application. Streamlit, which uses this port by default, is probably being used.

Name of Tab: app

The browser tab may have the name "app" for the application.

Additional Visual Components:

Background Picture: The use of a huge lock graphic most likely represents user authentication and security.

Notice of Caution: There is a yellow warning bar at the top: After 2024-04-11, yaml Copy Edit st.experimental_get_query_params will no longer be used. Streamlit is alerting developers to update deprecated functions with this message. Security Considerations: It's critical to save passwords safely, preferably using hashing algorithms like bcrypt. Ideally, the password visibility toggle should only be visible for a little moment or have the ability to rapidly hide it again. At least for registration and login, the program appears to have user authentication features configured.

Probable Technologies Employed: Technology for Components

Streamlit Localhost Deployment for Frontend User Interface Localhost server (default port 8501 in Streamlit)
Python is the backend language for user management. For user data, SQLite, Firebase, or local files may be used.
Logic for Authentication Python-based, potentially with simple libraries for authentication or hashing.

Fig 2: Login Page.

According to the URL (localhost:8501/?page=login), the image displays the login page of a locally hosted web application that was most likely created with Streamlit. The purpose of this interface is to authenticate users by confirming their login credentials.

Below is a thorough explanation of each page element:

Title and Design
The header "Login Page" is shown prominently on the page, suggesting its purpose.
Form elements are positioned in the center of the page for convenient access, and the layout is simple, clear, and easy to use.
Fields for Input
There are two input fields available:
Entering the user's registered username (which is already filled in in the screenshot) is known as the username.
To improve usability, Password features a masked input with an eye icon that lets users control password visibility.
When the "Login" button is clicked, the user authentication procedure is initiated.
Go to Register: This feature facilitates simple application navigation by sending users without accounts to the signup page.
Background with a Security Theme
A blue-toned 3D picture of a padlock on a keyboard serves as the background, highlighting the secure nature of the login procedure and signifying cybersecurity.
Extra Details: A warning regarding st.experimental_get_query_params deprecation appears at the top, indicating that the application is managing URL query parameters in Streamlit using an approach that may soon become obsolete. It shows that the developer is utilizing experimental capabilities of Streamlit, which might require an update in later iterations.

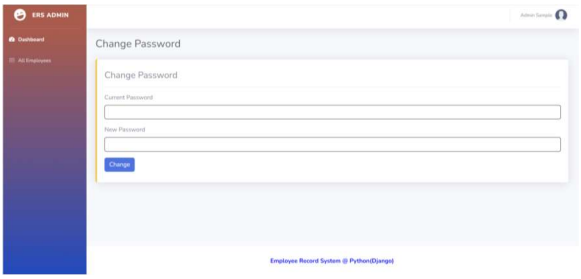

Fig 3: Fake Account Detection.

Based on specific user profile information, the image displays a "Fake vs. Real Account Prediction" page from a web application that was probably made with Streamlit to identify phony or suspicious online accounts.
In a nutshell, the page's goal is to determine whether a user account is authentic or fraudulent by examining particular attributes that the user has supplied.
Input Fields: Do You Have a Profile Picture?
Worth: 1
Probably a binary input that indicates whether the account has a profile image (1 = Yes, 0 = No).
Numbers and Username Length:
Value: 0.27 shows the proportion of numeric characters to the username's overall length.
Words from the whole name: Value: zero
It might show the number of distinct words in the whole name (0 could indicate no valid name).
Numbers and Full Name Length: Value: 0.00 shows the percentage of numbers in the entire name. Is Name the same as Username?
A partially hidden dropdown menu that probably asks if the user's login and complete name match.
Logout Button: This button, located on the left sidebar, indicates that only logged-in users can access this page.

Background and Style: A background image with a cybersecurity theme (keyboard, padlock, digital effects) interface that is clear, easy to use, and security-focused.

Note: As seen in the earlier screenshots, Streamlit's deprecation warning (st.experimental_get_query_params) is still visible. The developer is cautioned to upgrade code after April 11, 2024, by this.

IV. CONCLUSION

A machine learning-based method for precisely identifying phony accounts on social media sites was created for this research. The program successfully differentiates between legitimate and dubious identities by examining important profile characteristics like username patterns, the presence of a profile picture, and complete name information. The web application is accessible and useful for real-world use since it offers an easy-to-use interface for registration, login, and prediction. By proactively detecting or screening fraudulent individuals, this system shows how AI-driven techniques can improve digital safety and lessen the spread of criminal activity and false information.

REFERENCES

[1]. Yang, C., Harkreader, R., & Gu, G. (2011). Detecting spammers on social networks. Proceedings of the 26th Annual Computer Security Applications Conference, 1–10.

[2]. Stringhini, G., Kruegel, C., & Vigna, G. (2010) Detecting spammers on social networks. Proceedings of the 26th Annual Computer Security Applications Conference.

[3]. Cao, Q., Sirivianos, M., Yang, X., & Pregueiro, T. (2012). Aiding the detection of fake accounts in large scale social online services. USENIX Symposium on Networked Systems Design and Implementation (NSDI).

[4]. Ahmed, F., & Abulaish, M. (2013). A generic statistical approach for spam detection in online social networks. Computer Communications, 36(10–11), 1120–1129.

[5]. Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of Twitter accounts: Are you a human, bot, or cyborg? IEEE Transactions on Dependable and Secure Computing, 9(6), 811–824.

[6]. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017). The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. Proceedings of the 26th International Conference on World Wide Web Companion, 963–972.

[7]. Dickerson,J.P., Kagan, V., & Subrahmanian, V. S. (2014). Using sentiment to detect bots on Twitter: Are humans more opinionated than bots? IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 620–627.

[8]. Kumar, S., Spezzano, F., Subrahmanian, V., & Faloutsos, C. (2015). Edge weight prediction in weighted signed networks. IEEE 15th International Conference on Data Mining, 221–230.

[9]. Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. Communications of the ACM, 59(7), 96–104.

[10]. Botometer (Indiana University). (2024). Botometer: Check the activity of Twitter accounts.