International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 08 | Aug - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Anomaly Detection in Social Media Using Machine Learning

[1] Mr. Siddesh K T , [2] Guruprasad H S

[1] Assistant Professor, Department of MCA, BIET, Davanagere
[2] Student, 4th Semester MCA, Department of MCA, BIET, Davanagere

ABSTRACT

In today's digital communication landscape, social media platforms like Twitter (X), Facebook, Instagram, and YouTube have emerged as influential instruments for global connectivity, instantaneous news dissemination, public expression, and business marketing.

With billions of users engaging on a daily basis, these platforms produce an immense amount of user-generated content, encompassing text, images, videos, and interactions. Although social media presents extensive opportunities, it simultaneously fosters an environment conducive to various forms of abnormal and malicious activities, including spam, counterfeit accounts, orchestrated disinformation efforts, hate speech, phishing links, and cyberbullying.

Identifying such anomalies in real-time poses a significant challenge due to the ever-changing nature of social media content and the vast scale of data produced. Traditional detection methods are often ineffective, labor-intensive, and lack scalability. Consequently, this project seeks to harness the capabilities of Machine Learning (ML) to create an intelligent system that can accurately identify anomalous behaviors or patterns on social media platforms with minimal human oversight.

Keywords: Anomaly Detection, Social Media, Machine Learning, Natural Language Processing, Cyberbullying, Fake Accounts, Misinformation, Supervised Learning, Unsupervised Learning, Sentiment Analysis

1. INTRODUCTION

In the digital era, social media platforms like Facebook, Twitter (X), Instagram, and Reddit have revolutionized the methods by which individuals communicate, share information, express their views, and form communities. These platforms produce enormous amounts of user-generated content every second, encompassing posts, comments, likes, shares, images, and videos. While social media has certainly facilitated quicker information sharing and worldwide connectivity, it also presents numerous challenges—most notably, anomalous behaviors such as cyberbullying, spamming, misinformation, fake accounts, hate speech, phishing links, extremist propaganda, and organized disinformation campaigns. Anomalies in social media are characterized as behaviors or patterns that significantly diverge from the standard. These anomalies can be malicious or may signify suspicious activities, rendering them a significant concern for platform moderators, governmental bodies, and cybersecurity teams. For example, unexpected surges in tweet volumes featuring similar hashtags could indicate bot activity or coordinated propaganda, whereas an individual account that rapidly sends messages to unrelated users might suggest spam or phishing attempts. The demand for an intelligent and automated system to detect such anomalies has increased dramatically. Manual moderation has become impractical due to the vast scale of social media data. This highlights the significance of Machine Learning (ML).

ML provides scalable, data-driven methodologies to identify patterns and flag irregularities with high accuracy, learning from extensive datasets and continuously adjusting to new types of anomalies. This project aims to design and develop a machine learning-based anomaly detection system tailored for social media platforms. The goal is to establish a model capable of analyzing real-time or historical social media data to automatically detect abnormal behavior or content, requiring minimal human oversight. The system utilizes algorithms such as Support Vector Machines (SVM) and Decision Trees.

I. RELATED WORK

Because abnormal activities like spamming, organized disinformation campaigns, and the spread of fake news are becoming more common, anomaly detection in social media has emerged as a major research subject. Researchers have looked into a number of techniques for automatically spotting abnormalities that diverge from typical user or content trends. In order to detect suspicious or outlier activities, traditional methods usually use machine learning algorithms, such as Support Vector Machines (SVM), Random Forest, and clustering techniques. These algorithms examine user activity, text content, and network data.

More complex analysis of social media text and multimedia information has been made possible in recent years by developments in deep learning and natural language processing. Techniques that make use of transformer topologies, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) have demonstrated enhanced performance in identifying minute irregularities in noisy and large-scale data. These methods aid in the comprehension of semantic abnormalities, attitude shifts, and contextual nuances that could point to malicious intent or the dissemination of false information. Furthermore, hybrid approaches that integrate social network topology and content-based analysis have been put forth to identify coordinated attacks or bot activity.

Numerous research have concentrated on identifying false information and fake news by examining transmission patterns, user credibility, and linguistic clues. Graph-based algorithms are frequently used in these methods to find unusual spreading patterns in social networks. Notwithstanding encouraging outcomes, scaling these techniques for real-time identification and managing the variety of languages and formats seen on social media sites continue to offer difficulties. By combining machine learning and network analysis methods, this study seeks to expand on previous findings.

The application of unsupervised and semi-supervised learning approaches is a significant new avenue in social media anomaly identification. These techniques are especially useful because tagged anomalous data is scarce and social media dangers are constantly changing. Clustering techniques, autoencoders, and isolation forests have been employed to find unique patterns without relying largely on predetermined labels. These methods look for aberrations that might indicate spam accounts, phony profiles, or coordinated efforts by analyzing engagement metrics, behavioral patterns, and content semantics. Because of their versatility, they can be used for ongoing social media stream monitoring, where irregularities frequently appear in unexpected ways.

Additionally, the incorporation of real-time anomaly detection frameworks customized for social media platforms has been investigated by researchers. These frameworks use incremental learning and streaming data processing to quickly spot questionable activity as it happens. Due to the large volume and velocity of social media data, effective algorithms that can quickly adapt are required. Fast anomaly scoring techniques are frequently combined with feature extraction from text, pictures, and network interactions in real-time systems. This lets platform moderators take prompt action in addition to lessening the effect of bad content. In order to balance detection accuracy with the computational limitations of streaming social

media data, this study attempts to integrate aspects of real-time detection.

III. METHODOLOGY

The basics of ML involve input data, the learning process itself and output data:

3.1. Input data

For machine learning, a broad range of data can be used as input.This data may be organized or unstructured and may originate from a number of sources, including mainframe databases, enterprise systems, and Internet of Things edge devices.Because more data frequently produces greater insights, machine learning algorithms are frequently fed extremely large volumes of data.The digital business era, when information sources and volumes are proliferating, makes this worse.

3.2. Learning

Though there are many various kinds of algorithms and ML routines that can be used to achieve different goals, most machine learning (ML) applied for business reasons is either supervised or unsupervised in nature. Furthermore, there are frequently disparate approaches to learning, including "lazy" and "eager" approaches. These techniques control how training data is processed, and that control will dictate the amount of computing power and storage needed.

3.3. Output data

Both predictive (i.e., forecasting) and prescriptive (i.e., action recommendations) outcomes can be obtained by machine learning. Additionally, the results can produce outputs that categorize data or point out areas that require further investigation. This output data can be presented as reports, saved for study, or entered into other business systems or apps.

The first key step in preparing to explore and exploit ML is to understand the basic stages involved:

- **Classify the problem**

Create a problem taxonomy that explains how to group the issue or business query that needs to be resolved.A sample taxonomy for grouping issues or business challenges that need to be resolved using machine learning is provided by the "cheat sheet" .

- **Acquire data**

Determine the sources of the data that support the issue you are attempting to resolve. ERP systems, IoT edge devices, and mainframe data are just a few of the sources of data that can be employed in machine learning. Either structured (like NoSQL database records) or unstructured (like emails) data may be utilized.

- **Process data**

Identify how to prepare data for ML execution. Steps here include data transformation, normalization and cleansing, as well as the selection of training sets (for supervised learning).

- **Model the problem**

Determine the best way to get data ready for ML execution. The selection of training sets (for supervised learning) and data transformation, normalization, and cleansing are steps in this process.

- **Validate and execute**

Verify the outcomes, choose the platform on which to run the models and algorithms, and then carry out the machine learning operations. Numerous cycles of executing the ML function and fine-tuning the results will probably make up the execution phase.

- **Deploy**

Ultimately, the ML process's output is used to deliver some kind of business benefit. This value could take the shape of information that will be used to guide choices, power systems or apps, or be saved for later examination. The output may also include new models or routines that enhance current systems or applications (such predictive models), depending on the kind of machine learning operations that are carried out.This stage involves

deciding how and where to use the results for decision-making and consumption, regardless of their format.

IV. TECHONOLOGIES USED

Python: The main programming language that drives the entire application is Python. Because of its ease of use, large library, and robust support for machine learning and data analysis, it is highly recommended. Python makes it easier to construct backend applications, preprocess data, and integrate machine learning models seamlessly, resulting in precise real-time personality predictions.

Flask:A lightweight Python web framework called Flask was utilized to build the application's web in terface.It effectively integrates the frontend with ba ckend machine learning models, processes HTTP r equests, and controls user sessions.Prediction servi ces may be deployed with ease because to Flask's s treamlined design, which makes web app develop ment flexible and scalable.

LightGBM: Microsoft created LightGBM, a sophisticated gradient boosting system designed for quick training times and great performance. Large datasets with plenty of features are easily handled by it, and decision tree methods enable effective multi-class categorization. Its leaf-wise tree development and gradient-based one-sided sampling help to cut down on training time without sacrificing accuracy.

CatBoost: Yandex's CatBoost gradient boosting library was created especially to manage categorical data efficiently. By training on permutations of data subsets, it employs ordered boosting to avoid overfitting and target leaking. CatBoost is especially reliable and accurate in predicting tasks involving mixed data types because of its symmetric tree architectures and its handling of categorical characteristics.

Pandas: Pandas is a robust Python package for analyzing and manipulating data. It makes cleaning, converting, and organizing structured data—like CSV files—easier. Pandas is used in this application to preprocess and prepare datasets, facilitating the effective feature extraction and exploration required for machine learning model training.

Tkinter: The Tk GUI toolkit is standard Python interface is called Tkinter. This project makes advantage of it to create the graphical user interface elements that enable smooth user interaction with the application. Tkinter offers a lightweight and user-friendly framework for developing desktop applications by supporting the creation of windows, buttons, labels, and other widgets.

V. RESULT

Fig 5.1. Admin Page

Description : When the project is run, this is the first page that shows up on the screen.The preprocess button to process the data is displayed, along with the headline indicating Twitter spam filtering.

Fig 5.2. Home Page

Description :
This is the preprocessing page where the browsing dataset is preprocessed after being seen from the ex isting location.

Fig 5.3. Dataset File

Description: The data set page is this one. displays a data set with a list of various qualities in a table format.

Fig 5.4. Spam Filtering Based On Two Attributes

Description: We can identify Twitter spam based on the two attributes displayed on this page.

Fig 5.5. Classification Based On Subject

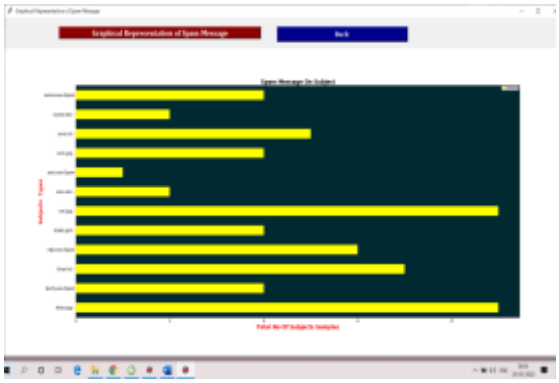

Fig 5.6. Classification Based On Extension

1. Fig 5.7. Spam Message Prediction

IV.CONCLUSION

The Twitter prototype is made to steer clear of harmful comments, like trolls, which damage users' reputations by sending them to spam messages. Twitter's division of spam and legitimate messages gave users a respectable level of privacy and security. Features can be added to the developing Twitter, which is currently in its initial state. A blocking tool may be put in place if the unapproved users keep tweeting offensive remarks. The system is able to recognize spam messages from unauthorized users; therefore, by recognizing a well-structured protocol mechanism, the same capability can be extended to authorized users.

REFERENCES

1) Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1–58.
2) Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.
3) Kumar, S., & Subbalakshmi, K. P. (2016). Detecting anomalies in social networks using machine learning techniques. In 2016 IEEE International Conference on Big Data (Big Data), 4086–4095.
4) Akoglu, L., Tong, H., & Koutra, D. (2015). Graph-based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 29(3), 626–688.
5) Salehi, M., Pourmohammad, M., & Ghaemi, B. (2018). Anomaly detection in social networks using machine learning algorithms. International Journal of Computer Applications, 179(47), 25–30.
6) Yu, H. F., Hsieh, C. J., Chang, K. W., & Lin, C. J. (2011). Large linear classification when data cannot fit in memory. ACM Transactions on Knowledge Discovery from Data (TKDD), 5(4), 1–23.
7) Pathak, M., & Lamba, A. (2021). Real-time anomaly detection in social media data using AI and machine learning models. Procedia Computer Science, 192, 487–496.
8) Ghosh, S., & Sanyal, S. (2012). Credit card fraud detection using machine learning models and anomaly detection approaches. International Journal of Computer Applications, 45(21), 39–44.
9) Zhang, Y., Zhou, Y., & Liu, Y. (2020). Fake news detection on social media: A data mining perspective. ACM Computing Surveys (CSUR), 53(5), 1–40.
10) Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), 1–21.