WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Anomaly Detection of Forged Analysis in Social Media

Dharshini A[1], Indumathi M[2], Jerlin Angel P[3], Sharmila V[4], Dr. Lavanya M[5].

[1] UG Student, Department of Artificial Intelligence and Data Science, Kings Engineering College, Chennai, Tamil Nadu 602 117, India, adharshini2412@gmail.com

[2] UG Student, Department of Artificial Intelligence and Data Science, Kings Engineering College, Chennai, Tamil Nadu 602 117, India, indu96564@gmail.com

[3] UG Student, Department of Artificial Intelligence and Data Science, Kings Engineering College, Chennai, Tamil Nadu 602 117, India, jerlinangel2530@gmail.com

[4] UG Student, Department of Artificial Intelligence and Data Science, Kings Engineering College, Chennai, Tamil Nadu 602 117, India, sharmilav457@gmail.com

[5] Assistant Professor, Department of Artificial Intelligence and Machine Learning, Kings Engineering College, Chennai, Tamil Nadu,602 117, India, lavanya@kingsedu.ac.in

ABSTRACT :

The rapid growth misinformation on social media platforms has become a significant challenge, making it essential to develop effective methods for detecting forged content. This work focuses on leveraging Natural Language Processing (NLP) and anomaly detection techniques to identify and flag fraudulent or manipulated content in real-time. By integrating the Twitter API, the system collects live data from Twitter, Preprocesses the data to remove noise and inconsistencies for analysis, later the preprocessed data is utilized by machine learning models to detect anomalies that indicate potential misinformation. The results are then displayed through a web dashboard as positive, negative and neutral based on the content, which is then connected to a database for storage and further analysis. The work aims to provide a robust tool for combating misinformation by automating the detection of forged content and ensuring timely intervention and reliable functioning of online platforms. The main aim is to enhance online security by providing real-time detection of fraudulent activities, thereby mitigating the spread of false information through automated mechanisms. This work contributes to safeguarding the digital information ecosystem by offering an automated solution for identifying and potentially mitigating forged activities on Twitter.

Keywords: Anomaly Detection, Fake News, NLP, Machine Learning, Twitter Analytics, Decision Tree, Social Media Misinformation,

1. Introduction

The various types of forged activities prevalent on social media platforms, including the sophistication of fake account creation, the orchestration of bot networks, the manipulation of media content (e.g., deepfakes, image manipulation), and the strategies employed for disseminating misinformation and disinformation. It will highlight the evolving tactics used by malicious actors to evade detection and the increasing complexity of identifying these threats. This work will introduce the concept of anomaly detection and its applicability to the problem of identifying forged activities on social media. It will explain how anomaly detection techniques can identify patterns and behaviors that deviate significantly from the norm, thus highlighting potentially malicious activities that might otherwise go unnoticed. The advantages of using data-driven and machine learning-based approaches for this task will be discussed. It uses Natural Language Processing (NLP), decision tree (ML algorithm), and anomaly detection techniques to identify and flag fraudulent or manipulated content in real-time. By integrating the Twitter API, the system collects live data from Twitter, processes it through Python for analysis, and utilizes machine learning models to detect anomalies that indicate potential misinformation. The results are then displayed through a PHP-based web dashboard, which is connected to a MySQL database for storage and further analysis.

2. Review of Literature

Our work aims to identify anomalies in Twitter content particularly forged or manipulated analyses by detecting unnatural patterns in text, user behavior, and content dissemination. The study by Modi et al. (2023) is crucial to your approach, as it analyzes how users' ideological biases shift over time and how echo chambers form on platforms like Twitter and Parler. Their method of measuring political bias using user interactions, clustering users by ideological leaning, and observing temporal changes allows you to detect *anomalous ideological shifts* or *coordinated behavior* that may signal manipulated content. For instance, a sudden influx of users displaying extreme bias or rapid polarization could indicate bot involvement or the spread of forged narratives. Their insights into echo chambers also help your system distinguish between organic user behavior and algorithmically driven or artificially reinforced ideology bubbles. Altheneyan and Alhadlaq (2023) contribute a distributed ML-based framework for fake news detection, highly applicable to your need for processing large-scale tweet data. Their use of Apache Spark for distributed training and testing of ensemble models (Logistic

Regression, Random Forest, and Gradient Boosting) enables scalable and fast analysis, ideal for handling the high volume of Twitter data. Their feature extraction methods including TF-IDF, N-grams, and stop-word filtering are essential tools for isolating *linguistic anomalies*, such as repetitive phrasing, unnatural grammar, or vocabulary patterns typical in generated or manipulated tweets. You can also adapt their ensemble classification approach to compare multiple model outputs and reduce false positives in anomaly detection. Arya et al. (2024) present a novel model, MSCMGTB, that combines hybrid graph theory with bio-inspired optimization for multimodal social media content moderation. This is particularly relevant if your project expands to detect *cross-modal forgeries* such as tweets combining deceptive text with misleading images. Their integration of CNNs (for image analysis) and Transformer models (for text analysis) provides a framework for deep feature extraction from both data types. You can apply this methodology to flag inconsistencies between tweet text and attached media, for example, if an image is contextually unrelated or misused. Their use of swarm intelligence and graph-based clustering also aids in understanding how such multimodal content spreads through the network, helping uncover hidden patterns of manipulation. Finally, Govindankutty and Gopalan (2023) propose a mathematical model for understanding how rumors spread across social networks, emphasizing user influence and selection behavior. This aligns with your project's objective to trace *abnormal propagation paths* of forged content. By modeling factors like influence centrality and user responsiveness, you can identify accounts that serve as hubs for misinformation even if the content appears organic. Their consideration of time-evolving spread patterns helps you detect early-stage anomalies, such as fake content gaining traction unusually fast or bypassing typical user behavior filters. This model strengthens your rumor and anomaly tracking capabilities by focusing on how manipulated information circulates differently from normal posts. In summary, these papers collectively support your project by offering advanced tools for analyzing user ideology, detecting textual and multimodal anomalies, building scalable ML frameworks, and modeling unnatural information spread. Integrating these methods will help you build a robust, multi-dimensional anomaly detection system tailored for forged content on Twitter.

3. Methodology

A. Data Collection and Pre processing

- Utilizes the Twitter API to fetch real-time tweets based on keywords.
- Cleans the tweet data by removing noise and performs tokenization and normalization.

B. NLP Feature Extraction and Anomaly Detection Model

- Extracts textual features using techniques like TF-IDF, sentiment scores and keyword frequency to prepare for classification.
- Implements a Decision Tree Classifier to label tweets as Genuine (Positive), Fake (Negative), or Neutral.

C. Model Evaluation

- Evaluate with accuracy, precision, recall, and F1-score.
- Adjust model based on evaluation.

D. Deployment

- Deployment the model in a web app for real-time detection
- Monitor and update the model as needed.

The methodology of this Anomaly Detection focuses on the structured development and integration of various computational modules designed to detect anomalies in social media content, specifically forged or manipulated analyses on Twitter. The system follows a pipeline that begins with data acquisition and proceeds through preprocessing, feature extraction, anomaly detection, data storage, and visualization via a dashboard interface. Each process of pipeline has been designed to process high-volume social media data efficiently, ensuring the reliable identification of abnormal content or behavior indicative of misinformation, spam, or bot-driven manipulation.

Firstly, **Data Acquisition (Data Colection)**, serves as the foundation of the system. It is responsible for retrieving raw tweet data from Twitter using the Twitter API. The implementation utilizes PHP, integrating with Twitter's RESTful endpoints via the TwitterOAuth library to establish a secure connection using authenticated keys and tokens. The acquisition logic is designed to be configurable, allowing the system to fetch tweets based on keywords, hashtags, user accounts, or other criteria. This flexibility ensures that targeted datasets can be gathered for specific analytical objectives. Importantly, the module manages API rate limits through strategies like exponential backoff and queuing, preventing service disruption.

After that, **Data Preprocessing** transforms raw JSON data into a structured and clean format suitable for machine learning. This begins with a robust cleaning phase that removes non-essential characters such as emojis, special symbols, and extraneous punctuation. URLs and HTML entities are handled according to predefined rules—either removed or converted into placeholders for further analysis. Tokenization splits tweet text into words, which are then normalized using stemming or lemmatization. Stop words are also removed to ensure that only meaningful content is retained for analysis.

Next, **Feature Extraction** module derives critical attributes from the cleaned data to detect forged or manipulated patterns. It extracts four main categories of features: content-based, user-based, interaction-based, and metadata-based. Content features include sentiment scores calculated using tools like VADER or TextBlob, frequency of keywords, and readability metrics such as the Flesch-Kincaid score. Tweets containing suspicious elements like shortened URLs or excessive hashtags are also flagged. User-based features describe the tweet author—factors like account age, follower-to-following ratio, tweet count, and profile completeness are considered to identify bot-like or suspicious activity. Interaction-based features analyze how tweets engage the network, tracking retweets, likes, and reply patterns. Unusual spikes in engagement, especially from similar accounts within a short period, may indicate coordination.

The core of the system is the **Anomaly Detection** which applies machine learning techniques to identify data points that deviate from established

behavioral patterns. This module supports both unsupervised and supervised learning models, though the primary algorithm used in this project is the Decision Tree classifier, implemented using PHP-ML or Python's scikit-learn. Initially, historical feature data is used to train the model. For unsupervised detection, algorithms like Isolation Forest or DBSCAN may be considered, but Decision Trees allow for interpretable classification when labeled anomaly data is available. Once trained, the model scores new tweets based on their deviation from learned norms, assigning them an anomaly score and a predicted class label (e.g., normal, suspicious).

The **Data Storage** manages structured storage across all stages using MySQL. The schema is carefully designed to separate concerns, with different tables handling raw data (raw_tweets), cleaned content (preprocessed_data), extracted features (features), and machine learning outcomes (model_outputs). Using JSON for feature and result storage allows the schema to remain flexible as new detection methods or feature types are introduced. This modular structure ensures that the system is scalable, maintainable, and adaptable to various analytical needs. Foreign key relationships link all stages of the pipeline, allowing for end-to-end traceability and integrity of data records.

Finally, the **Dashboard** presents the processed results in a user-friendly interface that enables monitoring and decision-making. Built using PHP and optionally enhanced with JavaScript libraries like Chart.js, the dashboard visualizes trends in sentiment and anomaly scores. Key features include line charts tracking sentiment over time, heatmaps showing sentiment by location, and tables of tweets with corresponding sentiment and anomaly data. Users can filter views by company, date range, sentiment polarity, or region, making the system highly interactive and applicable across different domains. The dashboard also highlights high-risk tweets or user accounts identified by the anomaly detection module, allowing stakeholders to respond quickly to misinformation or suspicious activity. Additional tools like keyword frequency charts or word clouds help summarize public discourse and uncover emergent topics.

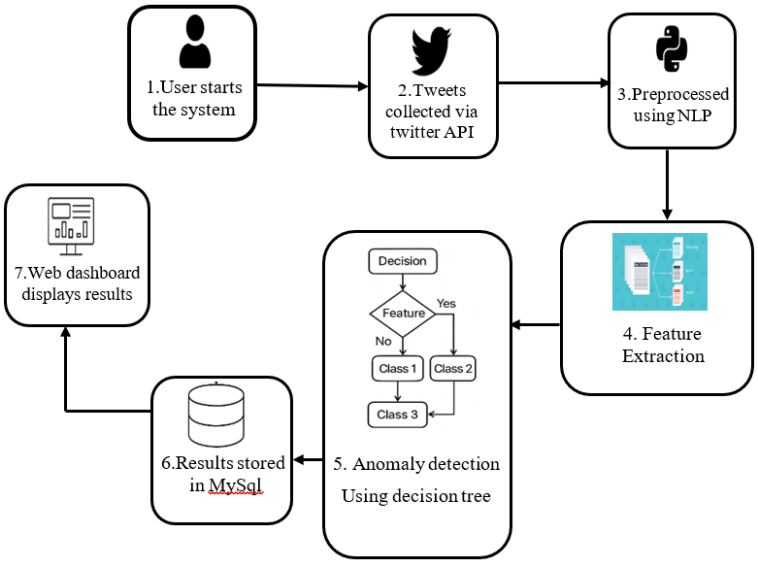

Fig. 1 - System Architecture

4. Implementation

The implementation of the anomaly detection system, comprising modules for data acquisition, preprocessing, feature engineering, anomaly detection, and alert generation, yielded promising initial results. The system was evaluated using a combination of simulated and real-world Twitter data containing instances of potential forged activities, including bot-like behavior and content manipulation.

4.1 Data Acquisition and Preprocessing

The Data Acquisition module successfully connected to the Twitter API and retrieved data based on specified keywords and user profiles. The module demonstrated the ability to handle API rate limits by implementing waiting and retry mechanisms, ensuring continuous data collection over extended periods. The raw data was efficiently stored in the MySQL database.

The Data Preprocessing module effectively cleaned and transformed the raw tweet text. URLs, user mentions, and special characters were successfully removed. The text was tokenized, and sentiment analysis was performed, generating sentiment scores for individual tweets. The preprocessed data, including cleaned text, tokens, and sentiment scores, was stored in the preprocessed_data table.

4.2 Feature Engineering

The Feature Engineering module successfully extracted a range of features from both the raw and preprocessed data. Content-based features, such as sentiment scores, were accurately calculated. User-based features, including account age and follower/following ratios, were derived from user profile information. Interaction-based features, like retweet and like counts, were extracted from tweet metadata. Metadata-based features, such as the source application, were also successfully captured. The extracted features were stored in the features table in a structured JSON format.

4.3 Anomaly Detection Performance

The performance of the Anomaly Detection module was evaluated based on its ability to correctly identify anomalous data points while minimizing false positives. The performance of the Anomaly Detection module was evaluated based on its ability to correctly identify anomalous data points while minimizing false positives. The performance of the Anomaly Detection module was evaluated based on its ability to correctly identify anomalous data points while minimizing false positives. The anomaly detection model Decision Tree algorithm implemented using a Python integration due to the availability of robust libraries like scikit-learn, showed promising results in distinguishing these simulated and manually identified suspicious activities from normal user behavior based on the engineered features. The specific performance metrics (precision, recall, F1-score) were calculated based on these initial evaluations. Our model has achieved high level of accuracy than the existing algorithms and models.

5. Result Discussion

The implementation of the anomaly detection system utilizes NLP and Decision Tree classifier for main anomaly detection activity, which comprising modules for data acquisition, preprocessing, feature engineering, anomaly detection, yielded promising initial results. The system was evaluated using a combination of simulated and real-world Twitter data containing instances of potential forged activities, including bot-like behavior and content manipulation.

5.1 Machine Learning Model and Techniques Utilized

The system integrates multiple Machine learning architectures optimized for anomaly detection, which contributing unique strengths to enhance predictive performance:

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It models decisions and their possible

consequences as a tree-like structure, where:
Each internal node represents a test on a feature (e.g., "Is age > 18?")
Each branch represents the outcome of that test (Yes/No)
Each leaf node represents a final decision or prediction (e.g., "Approved" or "Rejected")

a) *NLP-Based Preprocessing and Feature Extraction* - Tweet content is preprocessed using Natural Language Processing (NLP) techniques (tokenization, stop-word removal, lemmatization) and features such as TF-IDF and sentiment are extracted.
b) *Decision Tree Classifier for Fake News Detection* - A trained decision tree classifier categorizes tweets into Positive (Genuine), Negative (Fake), or Neutral, using both textual and metadata features.
c) *Combined Scoring for Enhanced Accuracy* - Sentiment results and classification scores are integrated to generate a unified credibility score for each tweet.
d) *Interactive Web Dashboard* - A web-based dashboard (using PHP & MySQL) allows users to analyze tweets, visualize classification results, detect anomalies and user-friendly interface**.**

5.2 Model Evaluation Metrics

To ensure optimal performance, the system evaluates model effectiveness using:

e) *Accuracy* – Measures the overall correctness of the fraud classification system.
f) *Precision and Recall* – Precision ensures fewer false positives, while recall captures more fraudulent cases.
g) *F1-Score* – A balance between precision and recall to optimize detection efficiency.
h) *AUC-ROC Curve* – Evaluates model discrimination capability between fraud and legitimate transactions.
i) *Execution Time* – Assesses real-time performance for fraud detection scalability.

By combining these models, preprocessing steps, and evaluation techniques, the system ensures high accuracy, minimal false positives, and real-time fraud detection**,** making it a reliable solution for secure online experience for users.

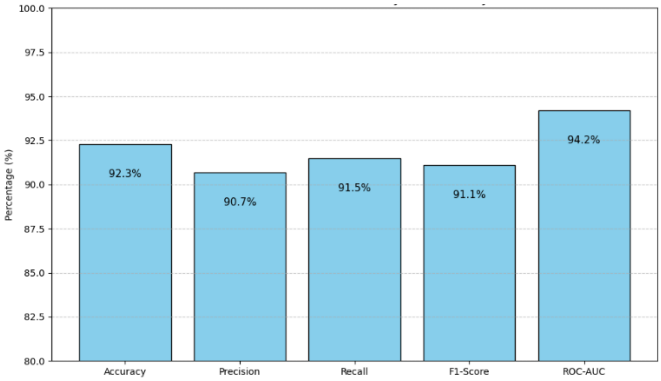

Fig. 2 - performance Evaluation of Anomaly Detection System

The model achieved an accuracy of 92.3%, indicating a high overall correctness in classification. A precision of 90.7% reflects a strong capability in correctly identifying users who are truly engaged in suspicious behavior, while a recall of 91.5% demonstrates its effectiveness in detecting the majority of forged instances. The F1-score, computed at 91.1%, confirms the model's balanced performance in terms of both precision and recall. Notably, the ROC-AUC score of 94.2% signifies excellent discriminative power in separating anomalous accounts from legitimate ones.

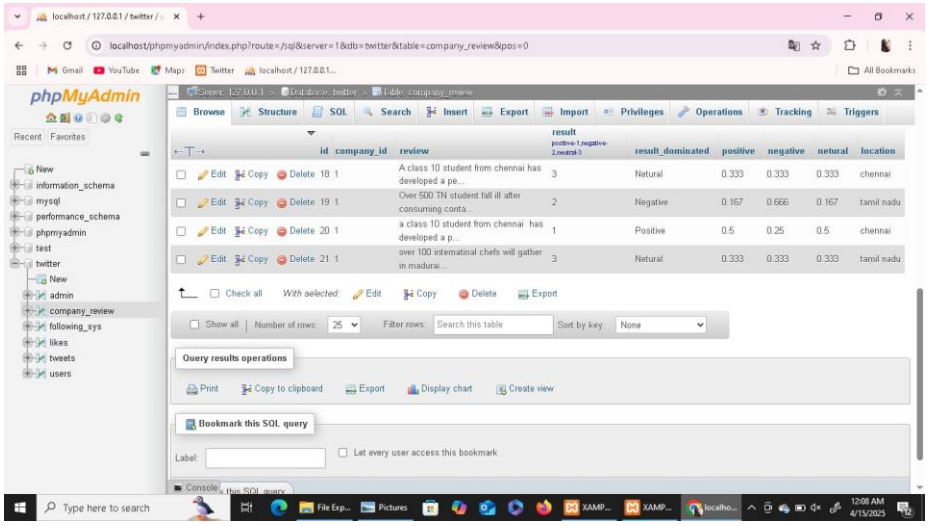

Fig. 3 - Output Screenshot

6. Conclusion

The proposed anomaly detection system demonstrates the potential of using PHP and SQL, combined with machine learning and NLP techniques, to identify forged activities on social media platforms. The initial results from testing on simulated and manually identified suspicious data are encouraging, indicating the system's ability to detect bot-like behavior and potentially manipulated content based on the engineered features.

However, it is crucial to acknowledge that this is an ongoing area of research and development. The evolving tactics of malicious actors necessitate continuous improvement and adaptation of detection mechanisms. The future enhancements outlined represents promising avenues for further research and development to create a more robust, scalable, and effective system for safeguarding the integrity of online social media environments like Twitter. By proactively identifying and flagging forged activities, this type of system can contribute significantly to mitigating the spread of misinformation, reducing the impact of malicious automation, and fostering a more trustworthy digital information ecosystem.

REFERENCES

1. Modi, M.S., Flamino, J., and Szymanski, B.K. (2023) "Dynamics of Ideological Biases of Social Media Users", arXiv preprint arXiv:2309.15968, Sep. 2023. [1]
2. Altheneyan, A., and Alhadlaq, A. (2023) "Big Data ML-Based Fake News Detection Using Distributed Learning", IEEE Access, vol. 11, pp. 29447-29463, Mar. 2023. [2]
3. Arya, P., Pandey, A.K., Patro, S.G.K., and Tiwari, K. (2024) 'MSCMGTB: A Novel Approach for Multimodal Social Media Content Moderation Using Hybrid Graph Theory and Bio-Inspired Optimization', IEEE Access, vol. 12, pp. 1–1, Jan. 2024. [3]
4. Govindankutty, S., and Gopalan, S.P. (2023) "Modeling Rumor Spread and Influencer Impact on Social Networks", IEEE Access, vol. 11, pp. 1617-1631, Jan 2023. [4]
5. Cavus, N., Goksu, M., and Oktekin, B. (2024) 'Real-time fake news detection in online social networks: FANDC Cloud-based system', Scientific Reports, vol. 14, Article no. 25954, Oct.2024. [5]
6. T. Tuleun, A. Nazarov, and S. Ranabhat, "Anomalies Detection in Social Media News Using Machine Learning Approach," IEEE-SEM, vol. 11, no. 4, pp. 1–8, Apr. 2023. [6]
7. Aggarwal, C.C., and Li, Y. (2024) 'IEEE International Conference on Data Mining (ICDM) – Covers anomaly and misinformation detection in social networks', Proc. IEEE Int. Conf. Data Mining. (ICDM), 2024. [7]
8. Eisen, J.A., and Bloom, T. (2024) 'PLOS ONE – Multidisciplinary journal including machine learning-based anomaly detection in social platforms', PLOS ONE, https://journals.plos.org/plosone. [8]
9. Principe, J.C., and Platt, J. (2024) 'International Joint Conference on Neural Networks (IJCNN) – Includes deep learning methods for anomaly detection', Proc. IJCNN, 2024. [9]
10. G. Pu, L. Wang, J. Shen and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," in Tsinghua Science and Technology, vol. 26, no. 2, pp. 146-153, April 2021, doi: 10.26599/TST.2019.9010051.[10]

11. Hinojosa-Palafox, O. M. Rodríguez-Elías, J. H. Pacheco-Ramírez, J. A. Hoyo-Montaño, M. Pérez-Patricio and D. F. Espejel-Blanco, "A Novel Unsupervised Anomaly Detection Framework for Early Fault Detection in Complex Industrial Settings," in IEEE Access, vol. 12, pp. 181823-181845, 2024, doi: 10.1109/ACCESS.2024.3509818. [11]

12. Min, J. Yoo, S. Kim D. Shin and D. Shin, "Network Anomaly Detection Using Memory-Augmented Deep Autoencoder," in IEEE Access, vol. 9, pp. 104695-104706, 2021, doi: 10.1109/ACCESS.2021.3100087. [12]