International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284

e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

Machine Learning-Based Detection of Malicious Applications in App Stores

Mrs. Suvarna S. Wakchaure[1], Ms. Shruti R. Jadhav[2], Ms. Aditi B. Vishwas[3], Ms. Kalyani B. Suryawanshi [4], Ms. Shruti S. Tungar [5]

[1]Assistant Professor, Department of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India.

[2,3,4,5] Department of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India.

Email ID: *suvarna.jondhale@pravara.in[1], shrutijadhav132004@gmail.com[2], aditivishwas614@gmail.com[3], kalyanisuryawanshi39@gmail.com [4], tungarshruti04@gmail.com[5].*

Abstract

With the exponential growth of mobile applications, users increasingly rely on app stores for essential services, making them vulnerable to fraudulent and malicious applications. Detecting such apps manually is difficult due to the vast number of available applications and misleading ratings or descriptions. To address this challenge, this project presents a machine learning–based approach for detecting fraudulent applications using sentiment analysis of user reviews. The proposed system analyzes textual user feedback to classify sentiments as positive, negative, or neutral, helping to uncover hidden patterns of suspicious behaviour. In addition to sentiment scores, features such as app ratings, review consistency, and app metadata are considered to improve detection accuracy. Multiple classification models, including Naïve Bayes, Logistic Regression, and Decision Tree Classifier, are implemented and evaluated. The performance of these models is measured using accuracy, precision, recall, and F1-score to ensure reliable classification. Experimental results demonstrate that sentiment analysis significantly enhances fraud detection capability by capturing real user experiences. Among the evaluated models, the Decision Tree Classifier achieved superior performance with high accuracy and F1-score, making it the most effective approach for identifying potentially fraudulent applications. This system provides an efficient and scalable solution to improve user trust and ensure a safer mobile app ecosystem.

Keywords: *Fraud App Detection, Sentiment Analysis, Machine Learning, User Reviews, Naïve Bayes, Logistic Regression, Decision Tree, Natural Language Processing, App Security, F1-Score*

1. Introduction

With the advancement of technology, the use of mobile phones has increased significantly. The number of applications available on major platforms such as Android and iOS has grown rapidly. This growth has created a major challenge in the field of business intelligence due to continuous development, marketing, and daily usage. As a result, the market has become highly competitive[1]. Companies and developers compete intensely to demonstrate the quality of their applications and invest considerable time and resources to attract users and ensure long-term sustainability. Customer feedback, including app updates, ratings, and reviews, plays a crucial role in this process. It helps developers identify issues and improve their applications to better meet user needs[2]. However, instead of relying only on traditional marketing strategies, some developers promote their applications in unethical ways to

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

improve their ranking in app stores. This is often done using techniques such as bot farms or "water armies" to artificially increase downloads, reviews, and ratings. In some cases, groups of people are hired to post fake reviews and ratings for the benefit of developers. This practice is known as crowd-turfing. Therefore, it is important to provide users with accurate and genuine feedback before they install any application, helping them avoid poor decisions. Since each app receives a large number of comments and ratings, an automated system is required to process and analyze this data effectively[4]. With the increasing demand for mobile applications, it is necessary to identify and flag suspicious apps as fraudulent so that users can easily recognize them. Users are often unable to determine whether the reviews and ratings they read are real or fake. To address this issue, this project presents a method for detecting fraudulent applications on platforms like the Google Play Store and Apple App Store[5]. The proposed system uses four key features—in-app purchases, advertisements, ratings, and reviews—to determine whether an app is genuine or fraudulent. The process begins by selecting these important features, followed by training the collected data using different classification models. The models used include Naïve Bayes (83% accuracy), Logistic Regression (84% accuracy), and Decision Tree (85% accuracy), with the Decision Tree model providing the highest accuracy for the system[6].

2. Literature Survey

Nevon Projects propose a comprehensive framework for detecting quality fraud, which may be enhanced by domain-generated data. It is one of the most advanced initiatives for detecting fraudulent applications through information algorithms[7]. This tool detects fraudulent applications with 75-80% accuracy. On paper, they provided a thorough analysis of the facts and a proposed fraud detection methodology[8]. They evaluated three forms of verification: quality-based assurances, rating-based guarantees, and review-based validation. In the paper, they consider only updates as parameters with the naive bayes algorithm[3].

Table 1 Comparative Analysis of Existing Voting Systems and Proposed VoteChain Improvements

Sr. No.	Author & Year	Title / Approach	Limitations	Improvement in VoteChain
1	Esther Nowroji, Vanitha (2016)	Detects fraud app ranking using IP recognition	Only IP tracking; ignores fake reviews	Combine with sentiment analysis and user behaviour
2	Shashank Bajaj et al. (2018)	Uses sentiment analysis on reviews & ratings	Handles only sentiment; misses subtle fake reviews	Add ML classification + rating/download patterns
3	S.R. Srividhya, S. Sangeetha (2019)	Weighted sentiment analysis to classify apps	Sensitive to noisy/sarcastic data	Use advanced NLP (BERT/LSTM) + multi-feature integration
4	Navdeep Singh et al. (2016)	ML/statistical analysis of rating patterns	Only rating patterns; ignores review text	Combine sentiment, behaviour, and metadata features

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

5	Keerthana B et al. (2018)	Rabin-Karp algorithm for malware & rank fraud detection	Static pattern matching; not adaptive	Integrate ML/anomaly detection for dynamic fraud

3. Discussion

The proposed system for detecting fraud apps utilizes sentiment analysis on user reviews combined with features such as ratings, downloads, and in-app purchases to classify applications as genuine or fraudulent. The results show that positive, negative and neutral reviews generally correspond to authentic apps, whereas apps with inconsistent review patterns, overly repetitive positive reviews, or negative sentiment hidden in text are likely fraudulent. Compared to prior approaches like IP tracking or session-based detection, this method captures subtle manipulations in user feedback and provides a more comprehensive evaluation of app credibility. Challenges included handling noisy data, sarcasm, very short reviews, and cleverly written fake reviews that appear genuine, which occasionally affected accuracy. Nevertheless, the approach achieved promising results, demonstrating that sentiment analysis is an effective tool for fraud detection and can assist users in making informed decisions while helping developers maintain fair competition. Future improvements could involve using advanced NLP models such as BERT or LSTM to better understand context and sarcasm, as well as incorporating additional features like review timing patterns and user credibility scores to enhance detection accuracy[11].

4. Problem Statement

With the rapid growth of mobile applications, users increasingly rely on reviews, comments, and ratings to decide which apps to download. However, fraudulent developers often manipulate ratings and post fake or misleading reviews to artificially boost their app rankings, making it difficult for users to identify genuine applications. Existing detection methods largely focus on IP tracking or session patterns and fail to analyze the **actual content and sentiment** of user feedback. This project addresses this gap by **analyzing user comments, reviews, and ratings** to calculate the **percentage of positive, negative, and neutral feedback**, providing a clear and automated measure of an app's authenticity[9]. This approach helps users make informed decisions while promoting transparency and fair competition in app stores.

5. Objectives Of the Proposed System

The primary goal of this project is to develop an **automated and reliable system to detect fraudulent mobile applications** by analyzing user reviews, comments, and ratings using sentiment analysis, helping users make informed decisions and promoting fair competition among developers[10].

5.1.Collection and Processing of App Data

To gather user reviews, comments, and ratings from mobile apps on platforms like Google Play and Apple App Store, and preprocess the data for analysis.

5.2. Sentiment Analysis of Reviews

To implement sentiment analysis techniques that classify user feedback into positive, negative, and neutral categories, capturing the overall perception of the app.

5.3. Quantitative Feedback Measurement

To calculate the percentage of positive, negative, and neutral reviews for each app, providing a clear metric for evaluating app authenticity.

5.4. Fraud Detection

To identify potentially fraudulent apps by detecting inconsistencies between review sentiment, ratings, and other app metrics such as downloads or in-app purchases.

5.5. User-Friendly Reporting System

To provide an intuitive system that presents sentiment percentages and fraud detection results in a way that allows users to make informed download

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

decisions.

5.6. Support for Fair Competition

To assist developers and app stores in maintaining **transparency** by highlighting suspicious apps, thereby discouraging fraudulent practices[12].

6. Proposed System & Methodology

The proposed system is an automated fraud detection platform for mobile applications that evaluates the authenticity of apps by analyzing user reviews, ratings, and comments[13]. The system uses sentiment analysis to categorize reviews as positive, negative, or neutral and calculates the percentage distribution of each sentiment. By comparing the sentiment analysis results with the app's ratings, download counts, and in-app features, the system can identify suspicious or fraudulent applications. This approach provides users with reliable feedback, helping them make informed decisions, while promoting transparency and fair competition among app developers.

6.1. System Architecture

The architecture of the proposed fraud app detection system consists of multiple functional layers that process app data to identify fraudulent behaviour[14]. As shown in Figure 1, the system follows a modular pipeline including data collection, preprocessing, sentiment analysis, fraud detection, and final outcome generation.

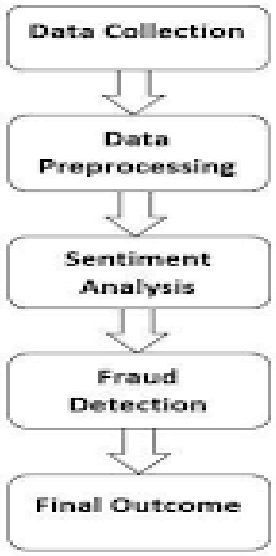

Figure 1 System Architecture

6.2.Main Layers of Architecture:

Data Collection

- This is the first step in the system

Data is gathered from different sources such as:

- User reviews
- Messages
- Complaints
- Transaction details

The quality of collected data directly affects the system's accuracy

Data Preprocessing

- Raw data is cleaned and prepared for analysis

Tasks performed:

- Removing stop words (like *is, the, and*)
- Removing punctuation and special characters
- Converting text to lowercase
- Tokenization (breaking text into words)
- This step ensures the data is structured and usable

Sentiment Analysis

- The processed text is analyzed to determine its sentiment

It classifies text into:

- Positive
- Negative
- Neutral
- Fraud-related content often shows negative or suspicious sentiment patterns

Fraud Detection

- Based on sentiment results and patterns, the system detects fraud
- Machine learning models or rules are applied
- The system decides whether the input is:
- Fraudulent
- Genuine

Final Outcome

The final result is displayed to the user

Output can be:

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

- Fraud detected
- No fraud detected

- Some systems may also show confidence level or alerts

6.3. System Workflow

The proposed system follows a systematic workflow to detect fraudulent mobile applications by integrating data preprocessing, sentiment analysis, and machine learning techniques.

Step 1: Data Collection

- Data is collected from app platforms such as Google Play Store and Apple App Store
- Includes user reviews, ratings, and basic app information[15]

Step 2: Data Preprocessing

- Raw data is cleaned by removing stop words, punctuation, and special characters
- Text is converted to lowercase and tokenized

Step 3: Sentiment Analysis

User reviews are analyzed using NLP techniques

Classified into: Positive, Negative and Neutral.

Step 4: Fraud Detection

- Machine learning models (Naïve Bayes, Logistic Regression, Decision Tree) are applied.
- Detects:
  oRating and sentiment mismatch
  o Repetitive or fake reviews

Step 5: Result Generation

- Application is classified as Fraudulent or Genuine
- Displays sentiment distribution (positive, negative, neutral)
- Results are visualized using a **pie chart** for better understanding.

6.4. Key Features Of The System

- **Sentiment Analysis of User Reviews** → Identifies positive, negative, and neutral feedback
- **Machine Learning-Based Classification** → Detects fraudulent apps using models like Naïve Bayes, Logistic Regression, and Decision Tree
- **Real-Time Fraud Detection** → Identifies suspicious patterns and fake reviews efficiently.
- **User-Friendly Interface** → Displays results using charts (e.g., pie chart) for easy understanding

6.5. Functional Objectives

The system is designed to achieve the following functional goals:

- Collect and preprocess app data from various platforms.
- Classify user reviews into positive, negative, and neutral sentiments.
- Analyze inconsistencies between ratings and review sentiment.
- Detect and classify applications as fraudulent or genuine.
- Provide clear visualization of sentiment distribution

7. Materials & Implementation

The implementation of the proposed fraud app detection system utilizes a combination of software tools and hardware components to ensure efficient data processing, sentiment analysis, and machine learning-based classification. The selected technologies support reliable data handling, accurate prediction, and effective visualization of results.

7.1. Hardware Requirements

Table 2 Hardware Components

Component	Description
Processor	Intel i3/i5 or higher
RAM	Minimum 4 GB (8 GB recommended for better performance)
Hard Disk	500GB

The hardware setup is minimal and practical,

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

making the system easy to deploy within educational institutions without requiring specialized equipment.

7.2. Software Requirements

Table 3 Software components

Component	Description
Operating System	Windows 10/11
Programming Language	Python (preferred for Machine Learning and NLP tasks)
Database	MySQL or PostgreSQL
ML/NLP Libraries	Scikit-learn, TensorFlow/PyTorch, NLTK/SpaCy
Development Tools	VS Code
Web Framework (if applicable)	Flask / Django

7.3. System Modules

The proposed fraud app detection system is divided into multiple modules, each responsible for a specific function in analyzing and identifying fraudulent applications.

1. Data Collection Module

- Collects app data from platforms such as Google Play Store and Apple App Store.
- Gathers user reviews, ratings, and app metadata

2.Data Preprocessing Module

- Cleans and prepares raw data for analysis
- Removes stop words, punctuation, and irrelevant data
- Performs tokenization and text normalization

3. Sentiment Analysis Module

- Analyzes user reviews using NLP techniques
- Classifies reviews into positive, negative, and neutral categories

4. Machine Learning Module

- Applies classification algorithms like Naïve Bayes, Logistic Regression, and Decision Tree
- Trains and tests models for accurate prediction

5. Fraud Detection Module

- Identifies fraudulent apps based on patterns and inconsistencies
- Detects fake reviews, rating mismatches, and abnormal behaviour

6. Result & Visualization Module

- Displays final classification (Fraudulent or Genuine)
- Shows sentiment distribution using charts (e.g., pie chart)

These modules work together to ensure a secure, efficient, and reliable voting system.

7.4. Implementation Details

- The proposed system for **Fraudulent Application Detection using Sentiment Analysis** is implemented as a **web-based application** using a client-server architecture. The frontend provides an interactive user interface through a web browser, while the backend processes user input, performs sentiment analysis on app reviews, and predicts whether an application is safe or fraudulent using machine learning models. The system is developed using Python, where the model is integrated with a web framework to deliver real-time results.
- **User Interface Dashboard:** Serves as the main entry point where users can input the application name or paste the app link to analyze its authenticity and risk level.
- **Data Collection and Review Extraction Interface:** Fetches user reviews and ratings of the selected application from the dataset or external sources, which are then used for further analysis.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

- **Text Preprocessing and Feature Extraction Module:** Cleans the collected reviews by removing stop words, punctuation, and noise, followed by feature extraction techniques such as TF-IDF to convert textual data into numerical format suitable for machine learning models.
- **Sentiment Analysis and Classification Module:** Applies machine learning algorithms such as Naïve Bayes, Logistic Regression, and Decision Tree to classify reviews into positive, negative, and neutral sentiments, and detect patterns indicating fraudulent behaviour.
- **Fraud Detection and Risk Prediction Interface:** Analyzes the aggregated sentiment results and predicts the risk level of the application (Safe / Suspicious / Fraudulent), providing users with a clear decision output.
- **Result Visualization Interface:** Displays the final prediction along with graphical representation such as accuracy comparison and sentiment distribution for better understanding of the analysis.

The system performance evaluation demonstrates improved accuracy and efficiency in detecting fraudulent applications compared to traditional manual inspection methods. The analysis of sentiment distribution and classification results highlights the capability of the system to identify suspicious patterns in user feedback. Overall, the proposed system ensures a reliable, scalable, and user-friendly solution for enhancing mobile application security.

Implementation Flow:

- User accesses the system through a web browser by running the application on a local server
- User enters the application name or provides the app link for analysis
- The system collects user reviews and ratings from the dataset or input source
- Extracted reviews are pre-processed by removing noise, stop words, and irrelevant data
- Feature extraction techniques such as TF-IDF are applied to convert text into numerical form
- The processed data is passed to machine learning models for sentiment classification
- The system analyzes sentiment results to identify suspicious patterns and detect fraudulent behaviour.
- Based on analysis, the system predicts the risk level of the application (Safe / Suspicious / Fraudulent)
- Final results along with visual representations are displayed to the user

The implementation ensures that the system operates efficiently by automating the process of analyzing user feedback and detecting fraudulent applications, while maintaining accuracy and ease of use for users.

7.5.Testing Scenarios

To validate the functionality of the proposed system, several test cases are considered:

- Analysis of a genuine application with predominantly positive reviews, where the system correctly classifies it as **Safe**
- Detection of a suspicious application with mixed reviews, where the system predicts a **moderate risk level**
- Identification of a fraudulent application with a high number of negative reviews, where the system classifies it as **Fraudulent**
- Handling of invalid input such as empty fields or incorrect application links to ensure proper error management
- Accurate sentiment classification and risk prediction using machine learning algorithms

These scenarios ensure the reliability, accuracy, and robustness of the system under different conditions.

8. Results And Discussion

8.1. Results

The proposed system for fraudulent application detection using sentiment analysis was evaluated

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

based on multiple performance parameters including classification accuracy, sentiment prediction efficiency, and overall system reliability. The results demonstrate that the system effectively identifies fraudulent, suspicious, and safe applications by analyzing user reviews. The machine learning models used in the system show satisfactory performance, with accurate sentiment classification and risk prediction. The system also ensures fast processing and user-friendly interaction, making it a reliable and efficient solution for enhancing mobile application security.

Figure 1 Home Page

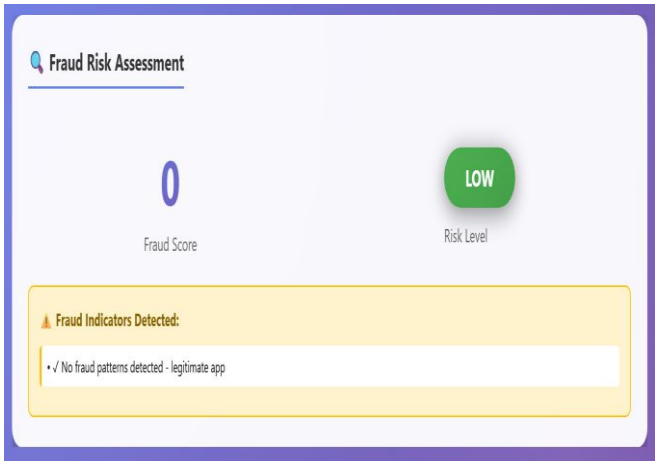

Figure 2 Positive Result

Figure 3 Result Graph

8.2. Integrity And Security

The proposed system maintains data integrity by ensuring accurate processing of user reviews and consistent sentiment classification without any data alteration. The use of preprocessing techniques and machine learning models helps in generating reliable and trustworthy predictions.

•Validates input data before analysis
•Ensures consistent and accurate sentiment classification
•Generates reliable and unmodified prediction results

The system demonstrates efficient performance by providing fast response time and accurate risk prediction, making it a dependable solution for detecting fraudulent applications.

8.3. Discussion

The proposed system highlights the effectiveness of sentiment analysis in identifying fraudulent applications by interpreting patterns in user reviews. The variation in sentiment distribution helps in understanding user trust and detecting suspicious behaviour in applications.

- High negative sentiment indicates potential fraudulent activity and poor user experience
- Mixed sentiment reflects uncertainty and possible risk associated with the application Consistent positive sentiment suggests reliability and user satisfaction.
- Model performance shows that combining multiple algorithms improves overall prediction reliability

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

The system demonstrates that analyzing user feedback can provide valuable insights into application credibility. However, the presence of fake or biased reviews may influence the prediction accuracy, indicating the need for more advanced techniques in future improvements.

Conclusion & Future Scope

Conclusion

The proposed system successfully demonstrates the use of sentiment analysis and machine learning techniques for detecting fraudulent mobile applications. By analyzing user reviews and classifying sentiments, the system is able to identify safe, suspicious, and fraudulent applications with good accuracy. The integration of multiple machine learning models improves the reliability and effectiveness of predictions. The web-based implementation ensures ease of use and real-time analysis, making the system practical for users. Overall, the proposed approach provides an efficient and scalable solution for enhancing mobile application security. Future improvements can include the use of advanced deep learning techniques and real-time data integration to further increase accuracy and robustness.

Future Scope

- **Integration with Advanced Deep Learning Models**
  The system can be enhanced by incorporating deep learning techniques such as LSTM or BERT for more accurate and context-aware sentiment analysis.
- **Real-Time Data Integration**
  Future versions can include real-time extraction of reviews from app stores to enable dynamic and up-to-date fraud detection.
- **Fake Review Detection Mechanism**
  Additional modules can be implemented to identify and filter fake or spam reviews, improving the reliability of predictions.
- **Mobile Application Development**
  A dedicated mobile application can be developed to increase accessibility and allow users to analyze apps directly from their smartphones.
- **Scalability and Dataset Expansion**
  The system can be expanded with larger and more diverse datasets to improve model performance and support large-scale application analysis.

References

[1]. S. R. Srividhya and S. Sangeetha, “A methodology to detect fraud apps using sentiment analysis.”
[2]. S. Bajaj, N. Nigam, P. Vandana, and S. Singh, “Detection of fraud apps using sentiment analysis,” International Journal of Innovative Science and Research Technology.
[3]. H. Zhu, H. Xiong, Y. Ge, and E. Chen, “Mining mobile app usage for fraud detection,” in Proc. ACM CIKM, 2013.
[4]. N. Jindal and B. Liu, “Opinion spam and analysis,” in Proc. WSDM, 2008.
[5]. E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, “Detecting product review spammers using rating behaviours,” in Proc. ACM CIKM, 2010, pp. 939–948.
[6]. J. Wan, M. Liu, J. Yi, and X. Zhang, “Detecting spam webpages through topic and semantics analysis,” in IEEE GSCIT, 2015.
[7]. N. Singh, P. K. Pandey, and Srinivasan, “Improved discovery of rating fake for mobile apps,” in IEEE ICONSTEM, 2016.
[8]. H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian, “Mining context-aware preferences for mobile users,” in IEEE ICDM, 2012.
[9]. B. Liu, Sentiment Analysis and Opinion Mining. Morgan & Claypool, 2012.
[10]. F. Sebastiani, “Machine learning in automated text categorization,” ACM Computing Surveys, 2002.
[11]. T. Mikolov et al., “Efficient estimation of word representations,” in ICLR, 2013.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0284
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 1934 - 1943

[12]. K. Ravi and V. Ravi, "A survey on sentiment analysis and opinion mining," Knowledge-Based Systems, 2015.
[13]. [13] S. B. Kotsiantis, "Supervised machine learning techniques," Informatica, 2007.
[14]. A. McCallum and K. Nigam, "Naive Bayes text classification comparison," in AAAI Workshop, 1998.
[15]. J. Leskovec, A. Rajaraman, and J. Ullman, Mining of Massive Datasets. Cambridge University Press, 2014.