Automated Telegram Group Management System for Spam and NSFW Content Prevention

[1]Jithin S, [2]Crisbin Joseph, [3]Pournami S, [4]Sayanth TM, [5]Theja Murali

[1]Assistant Professor, [2]Student, [3]Student, [4]student, [5]Student
[1]Department of Computer Science and Engineering,
[1]St. Thomas college of Engineering and Technology Mattanur, Kannur, India
[1]jithinsubhash@stthomaskannur.ac.in, [2]crisbinjoseph.dev@gmail.com,
[3]pournami236@gmail.com, [4]sayanth.t.m9090@gmail.com, [5]thejamurali21@gmail.com

Abstract—**In contemporary digital communication, ensuring the integrity of Telegram group communities presents significant challenges, particularly due to the proliferation of spam and NSFW (Not Safe For Work) content. The limitations of hu- man administrators in continuously monitoring group activities necessitate an automated solution. This project introduces an automated Telegram group management system, developed using Node.js in conjunction with Python and JavaScript libraries and models, to efficiently detect and mitigate undesirable content. The proposed system integrates a bot administrator capable of real-time monitoring of all incoming messages, including text, images, and videos. Leveraging advanced algorithms and libraries, the bot analyzes each message to identify spam and NSFW content. Upon detection, the bot automatically deletes the offending message for all group members and removes the sender from the group. The system also addresses messages containing malicious external links or unauthorized group in- vitations. Additionally, the bot provides comprehensive support for administrative commands, enabling human administrators to perform key management tasks such as adding members, retrieving the group link, removing users, and issuing warnings. This solution ensures a secure and well-regulated communication environment for Telegram group communities.**

***Index Terms*—Threat Intelligence, Fake Website Detection, Counterfeit Product Detection, Sentiment Analysis, NLP**

I. INTRODUCTION

In the digital age, the rapid growth of online communities has made platforms like *Telegram* essential for communication, networking, and information exchange. However, managing large, dynamic *Telegram* groups is challenging, especially with the increasing prevalence of spammers and **NSFW** (Not Safe For Work) content. These disruptions compromise group quality and safety, overwhelming human administrators. Thus, an automated solution is needed to maintain group integrity without constant human intervention.

This project proposes an automated *Telegram* group management system built with **Node.js**, enhanced by Python and JavaScript libraries. The system operates as an intelligent bot, acting as a virtual admin to monitor all incoming messages—text, images, or videos. Using advanced algorithms, the bot identifies and flags spam or **NSFW** content, deleting such messages and warning or removing the sender to ensure a safe environment.

Additionally, the bot detects harmful external links or invitations to problematic groups, providing an extra layer of security against external threats.

The bot also includes admin commands for tasks like adding members, retrieving group links, removing users, and issuing warnings, allowing administrators to maintain control without constant monitoring.

In summary, this automated system offers a scalable solution for maintaining *Telegram* group integrity. By automating content moderation, human administrators can focus on fostering a positive atmosphere while the bot handles spam and **NSFW** content, ensuring a secure environment for all members.

II. PROBLEM DEFINITION

With the rapid growth of online communities, managing *Telegram* groups has become increasingly challenging due to spam, **NSFW** (Not Safe for Work) content, and harmful external links. These issues threaten community integrity and user safety, while relying on human moderators alone is inefficient and often unable to provide real-time responses.

To address this, the project proposes an intelligent *Telegram* bot built with **Node.js**, **Python**, and **JavaScript** libraries. The bot leverages advanced **machine learning** models to analyze messages in real time, detect and remove spam, **NSFW** content, and harmful URLs, while also supporting core administrative tasks.

This solution reduces the manual workload for administrators, ensuring consistent moderation and improving user experience and security in *Telegram* groups.

III. RELATED WORKS

The literature review examines three key studies on spam detection, focusing on methodologies, advantages, and limitations. These studies highlight the growing need for advanced techniques to combat spam in various digital environments, including emails, SMS, and social networks. Each study offers unique insights into addressing spam through innovative approaches, such as Natural Language Processing (NLP), hybrid deep learning models, and graph- based machine learning. The review provides a comparative analysis of these methods, emphasizing their strengths and challenges in real-world applications.

A. A Novel Approach for Spam Detection Using NLP with AMALS Models

This study introduces a spam detection framework that leverages Natural Language Processing (NLP) and Altered Minimum Absolute Least Squares (AMALS) models. The approach combines gradient descent and AMALS to estimate missing data, addressing challenges like data sparsity. The framework achieves 98% accuracy, outperforming traditional methods like TF-IDF. Key components include preprocessing, feature extraction using Bag-of-Words (BOW) and BERT embeddings, and data imputation through the AMGD algorithm and AMALS model.

Despite its high accuracy, the method is computationally expensive and requires large labeled datasets. The architecture includes a prediction module for real-time classification and a feedback loop for continuous improvement. This study highlights the potential of advanced NLP techniques in spam detection but also emphasizes the need for efficient computational frameworks to handle large-scale data.

B. Multi-Type Feature Extraction and Early Fusion Frame- work for SMS Spam Detection

This paper proposes a hybrid deep learning model for SMS spam detection, combining lexical, syntactic, and semantic features. The framework achieves 99.56% accuracy on the UCI dataset, surpassing traditional methods like Random Forest (RF) and BERT. The model integrates local, temporal, and global text features to enhance detection robustness, making it effective for structured data like SMS.

However, the approach struggles with unstructured spam and has limited generalization. The framework involves preprocessing, feature extraction, and classification using deep learning models. Validated through cross-validation, the model demonstrates high accuracy and reliability. This study underscores the effectiveness of hybrid models in SMS spam detection but highlights the need for further research to address unstructured data challenges.

C. Spammer Detection in Social Networks Using ML and NLP

This research focuses on detecting spammers in social networks like Twitter and Facebook. It identifies spam through fake content, URL-based spam, trending topics, and fake users. The study leverages graph-based machine learning and NLP to capture relational data and scale for large networks. Key strategies include analyzing user authenticity, content quality, and time-based patterns.

While effective, the method relies heavily on graph quality and is limited to social networks. The study highlights the challenges of spam detection in dynamic environments, where spammers exploit easy account creation to spread harmful content. This research provides valuable insights into combating spam on social platforms but calls for more adaptable techniques to address evolving spam tactics.

TABLE I

COMPARISON OF SPAM DETECTION APPROACHES USING NLP

Methodology	Key Features	Advantages	Limitations
A Novel Approach for Spam Detection Using AMALS Models	Uses Attention Mechanism and Multi-layer Stacked Learning (AMALS).	- High accuracy. - Reduced false positives.	Computationally expensive. Needs large labeled datasets.
Multi-Type Feature Extraction and Early Fusion Framework for SMS Spam Detection	Combines lexical, syntactic, and semantic features.	- Robust feature extraction. - Handles imbalanced datasets.	- Struggles with unstructured spam. Limited generalization.
Spammer Detection in Social Networks Using ML and NLP	Uses graph-based ML models with NLP for social network spam.	- Captures relational data. - Scalable for large networks.	- Relies on graph quality. - Limited to social networks.

IV. PROPOSED SYSTEM

Managing large-scale groups on platforms like Telegram is challenging due to the prevalence of spam and NSFW (Not Safe for Work) content. Human moderators often struggle to maintain a secure and respectful environment, as manual over- sight cannot keep up with the high volume of messages. This issue is particularly critical in communities that rely on shared knowledge and interactions. To address these challenges, this project proposes an automated Telegram Group Management System using Node.js and JavaScript libraries. The system aims to detect and remove spam, NSFW content, and harmful links in real-time, ensuring a safer and more organized group environment.

The system integrates a bot to monitor messages, images, and videos, using advanced algorithms for rapid violation detection. It also supports admin commands, enabling administrators to manage groups efficiently. By automating content moderation, the system reduces the burden on human moderators, allowing them to focus on higher-level oversight. Drawing from advancements in spam detection, the project incorporates NLP techniques, text preprocessing, feature extraction, and malicious link classification

to create a comprehensive solution tailored for Telegram. This approach positions the system as a pioneering tool in automated community management, meeting the growing demand for effective moderation on digital platforms.

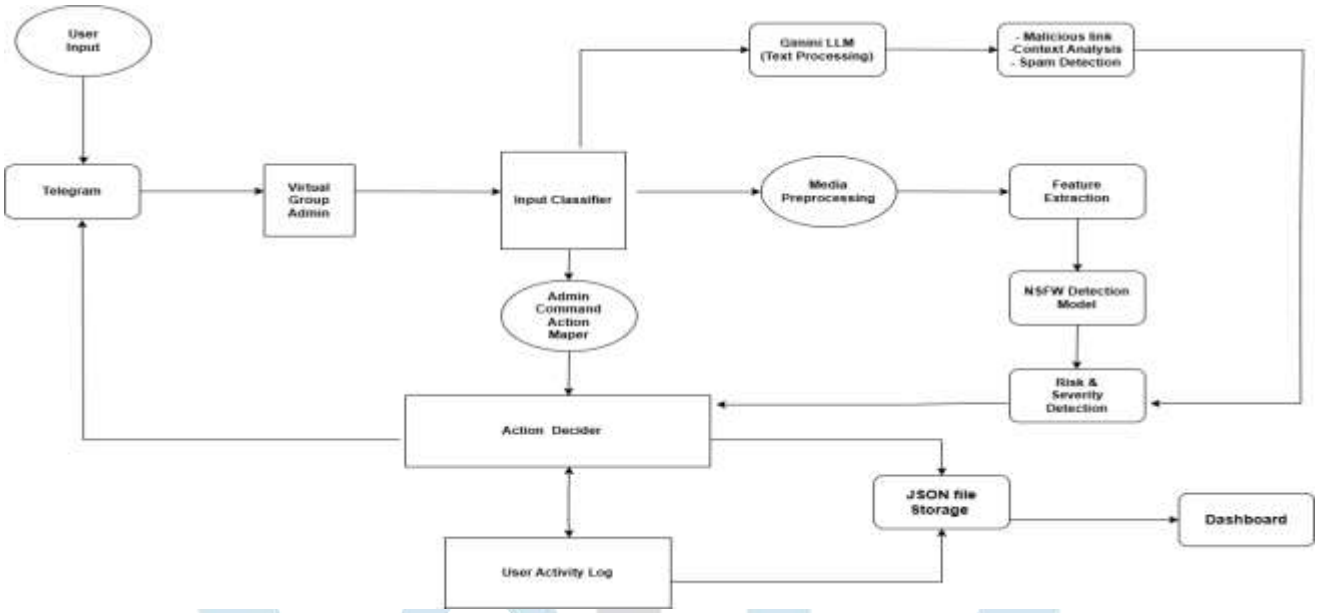

Fig. 1. Architecture Diagram

The flowchart outlines a sophisticated system for the automated moderation and classification of user input within a Telegram group. Designed with a robust architecture, it employs advanced artificial intelligence models and processes to ensure effective content management. The workflow is meticulously organized to handle various input types—text, audio, and media—and seamlessly integrate them into a unified moderation pipeline.

The process begins when a user submits input through Telegram. The Telegram platform forwards the input to the ***Virtual Group Admin***, which serves as the central controller, directing the data to the appropriate processing modules. The first step is performed by the ***Input Classifier***, which identifies the type of input (text, audio, or media) and routes it accordingly.

For text inputs, the system applies preprocessing to clean and normalize the data, ensuring consistency for downstream analysis. Key features are then extracted to facilitate the detection of malicious links via a ***Malicious Link Classifier***. Simultaneously, the processed text undergoes spam detection using a dedicated ***Spam Detection Model***. The system leverages ***Gemini LLM by Google***, a cutting-edge large language model, for the precise classification of textual content, ensuring reliable identification of harmful or inappropriate material.

In the case of audio inputs, the system integrates ***Assembly AI*** for converting speech to text. This transcription step ensures that spoken content is accurately translated into a textual format. The resulting text is analyzed by ***Gemini LLM***, enabling the system to detect spam, inappropriate language, and other potential violations. The dual use of speech-to-text conversion and large language model analysis enhances the system's ability to process audio content with both linguistic and contextual accuracy.

For media inputs, such as images and videos, the system initiates a Media Preprocessing stage to prepare the data for detailed examination. It employs ***Falcon AI***, a specialized model known for its efficiency in detecting inappropriate or NSFW content. Once flagged, the media content undergoes a ***Severity Detection*** process, assigning a criticality level to assess the impact and priority of the identified issues.

The system translates all classified inputs—text, audio, or media—into actionable responses via the ***Admin Command Action Mapper***. This component maps classification results to specific administrative commands, which are then passed to the ***Action Decider***. The ***Action Decider*** determines the most appropriate course of action, such as issuing a warning, removing content, or restricting a user's access to the group.

To maintain operational transparency and facilitate auditability, all actions and decisions are logged in the ***User Activity Log***. This record provides a comprehensive history of user interactions and system responses, supporting ongoing analysis and refinement.

The Automated Telegram Group Management System employs a modular, AI-driven approach to effectively manage and moderate group content, utilizing advanced spam and NSFW content detection techniques. Each component of the system is designed to enhance group safety by identifying and filtering harmful messages, images, and links in real-time, thereby fostering a secure and respectful environment for group members. The following outlines the core methodologies integrated into the system.

A. *Text Spam Detection Model*

The system begins with a text spam detection module that employs Natural Language Processing (NLP) and machine learning techniques to identify unwanted messages and malicious content within group conversations.

1) Text Preprocessing: Incoming messages are cleaned and standardized, removing unnecessary symbols, repeated characters, and stop words to prepare for analysis. This pre- processing step ensures that the model can effectively analyze the relevant content of each message.

2) Feature Extraction: Important linguistic features are extracted, focusing on word frequency, message patterns, and user behavior, thus creating a dataset that aids in identifying spam indicators. This stage leverages techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and n-grams to enhance feature representation.

3) Spam Classification: Utilizing a supervised learning model, the classifier categorizes messages as spam or genuine. The model is trained on labeled datasets, enabling it to filter out spam content quickly and efficiently, thereby reducing the workload on human moderators.

B. *Malicious Link Detection*

To protect against external threats, a malicious link classifier scans URLs for signs of phishing, malware, or spam.

1) Technique: Machine learning algorithms analyze each URL and compare it with a comprehensive blacklist of known malicious sites while assessing various risk indicators. This process includes checking URL length, the presence of suspicious parameters, and domain reputation scores.

2) Function: Probabilistic models assign a risk score to each link based on historical data, allowing the system to remove flagged links immediately and safeguard group members from potential online threats.

C. *NSFW Media Detection and Classification*

For multimedia files, an NSFW detection model employs computer vision techniques to automatically detect inappropriate or explicit content.

1) Media Preprocessing: Images and videos are resized and standardized to allow for efficient processing, ensuring that the model can handle a variety of media formats.

2) Feature Extraction and Classification: Utilizing Convolutional Neural Networks (CNNs), the system identifies features associated with NSFW content. The model is trained on large datasets containing both NSFW and safe media, allowing it to assign a confidence score for each piece of media based on its classification.

3) Warning and Removal:

If the NSFW probability exceeds a preset threshold, the bot automatically deletes the media and notifies the administrators. If a user exceeds a flag count of three, despite receiving three warnings—one for each detected NSFW violation—they are removed from the group without further warnings. All detected NSFW content is also removed. This entire process operates autonomously without human intervention, ensuring inappropriate content is promptly managed and maintaining the integrity of the group environment.

D. *Admin Command Integration*

The system incorporates an admin command module that enables real-time control and monitoring of group activities.

1) Function: Administrators can utilize predefined commands to adjust bot settings, manage filters, and review activity logs. This module directly links commands to bot functions, providing administrators with streamlined group management capabilities, including the ability to temporarily mute problematic users or escalate issues to more severe penalties, such as permanent bans.

E. *Action Decider Module*

A crucial component of the system, the action decider, coordinates responses by analyzing detected issues and determining the appropriate course of action.

1) Technique: This module evaluates user behavior, message patterns, and violation frequency to decide on the appropriate response, whether it involves deleting content, warning the user, or blocking the account. The decision- making process leverages heuristics and rule-based systems to tailor responses according to the severity and context of the violation.

2) Function: By dynamically assessing the severity of each detected offense, the action decider automates user management actions to maintain group order effectively, ensuring that appropriate measures are taken without unnecessary delays.

F. *Module Integration and Testing*

Finally, each module undergoes rigorous integration testing to ensure seamless functionality across different scenarios.

1) Technique: Modules are tested with various spam types, user behaviors, and NSFW media cases to validate the system's performance in real-world conditions. This comprehensive testing includes unit tests, integration tests, and user acceptance testing to ensure reliability.

2) Function: This iterative testing process helps refine the models and bot logic, guaranteeing the robustness of the Telegram Group Management System. Continuous feedback loops from administrators and users further enhance system performance, allowing for ongoing improvements in detection accuracy and user experience.

V. RESULT

The implementation phase of the Telegram Group Management System focuses on integrating AI-driven moderation techniques to enforce group policies in real-time. The system includes content moderation for text, images, videos, and audio to prevent the spread of explicit or inappropriate materials. Along with the bot, an admin dashboard provides insights into flagged content, user violations, and group activities. The system interacts with the Telegram API while maintaining accuracy and efficiency.

VI. TELEGRAM BOT IMPLEMENTATION

The Telegram bot is the core component of this system, responsible for real-time moderation and enforcement of group rules.

A. Bot Profile and Interface

The bot is registered in a Telegram group with a defined pro- file, including its name, description, and command structure. Users and administrators interact with the bot using commands or automated detections.

Fig. 2. Bot Profile and Interface

B. Text Moderation

The bot scans messages sent in the group to ensure a safe communication environment. - Messages undergo analysis using Google's Gemini language model, fine-tuned on Google AI Studio. - Inappropriate messages are removed, and the sender receives a warning. - Administrators are notified about repeated violations.

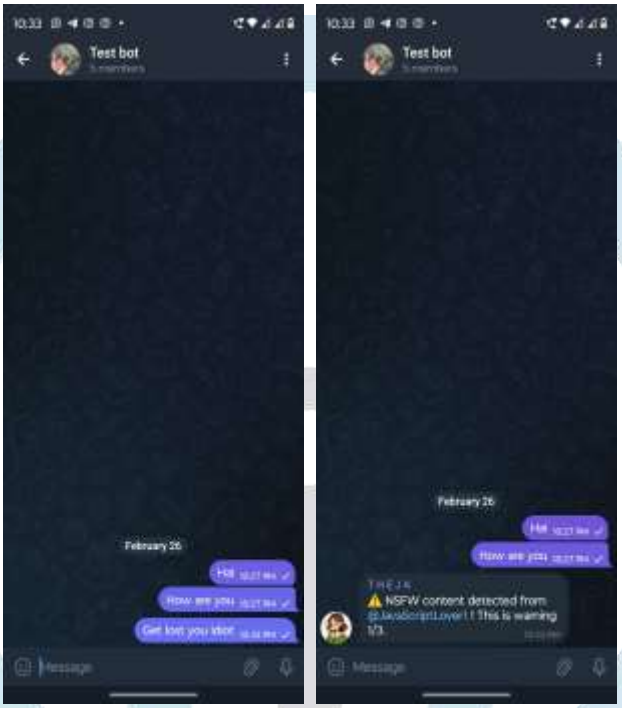

Fig. 3. Text Moderation - Message Sent vs. NSFW Text Detected

C. Image Moderation

Images shared in the group are screened using the Falcon AI model hosted on Hugging Face. - If an image is classified as inappropriate, it is removed immediately. - Users attempting to send restricted images receive a notification, and repeated violations may lead to bans

Fig. 4. Image Moderation - Image Sent vs. NSFW Image Detection

D. Video Moderation

Video moderation involves analyzing frames from different intervals to detect inappropriate content. - The bot extracts the video's thumbnail for a preliminary scan. - If flagged, the video is removed before it can be viewed by group members.

E. Audio Moderation

Audio messages are analyzed using speech-to-text conver- sion. - Each voice message is transcribed into text. - The transcribed content is checked for violations. - If flagged, the voice message is removed, and the user is notified.

Fig. 5. Audio Moderation - Audio Sent vs. NSFW Audio Detected

F. Warning and Removal

If a user sends NSFW content (video or audio), the bot detects and removes the content immediately. The user re- ceives a warning for each of the first three violations. How- ever, if the preset flag count of three is exceeded, the bot automatically removes the violator from the group without any further warnings. This process ensures swift action against repeated violations, maintaining a safe and appropriate group environment.

Fig. 6. Warning and Removal

VII. Admin Dashboard Implementation

The admin dashboard provides control and transparency for group owners to monitor flagged content and take action.

A. *Dashboard Overview*

The dashboard presents key statistics, including the total number of messages processed, flagged messages, and real- time engagement levels.

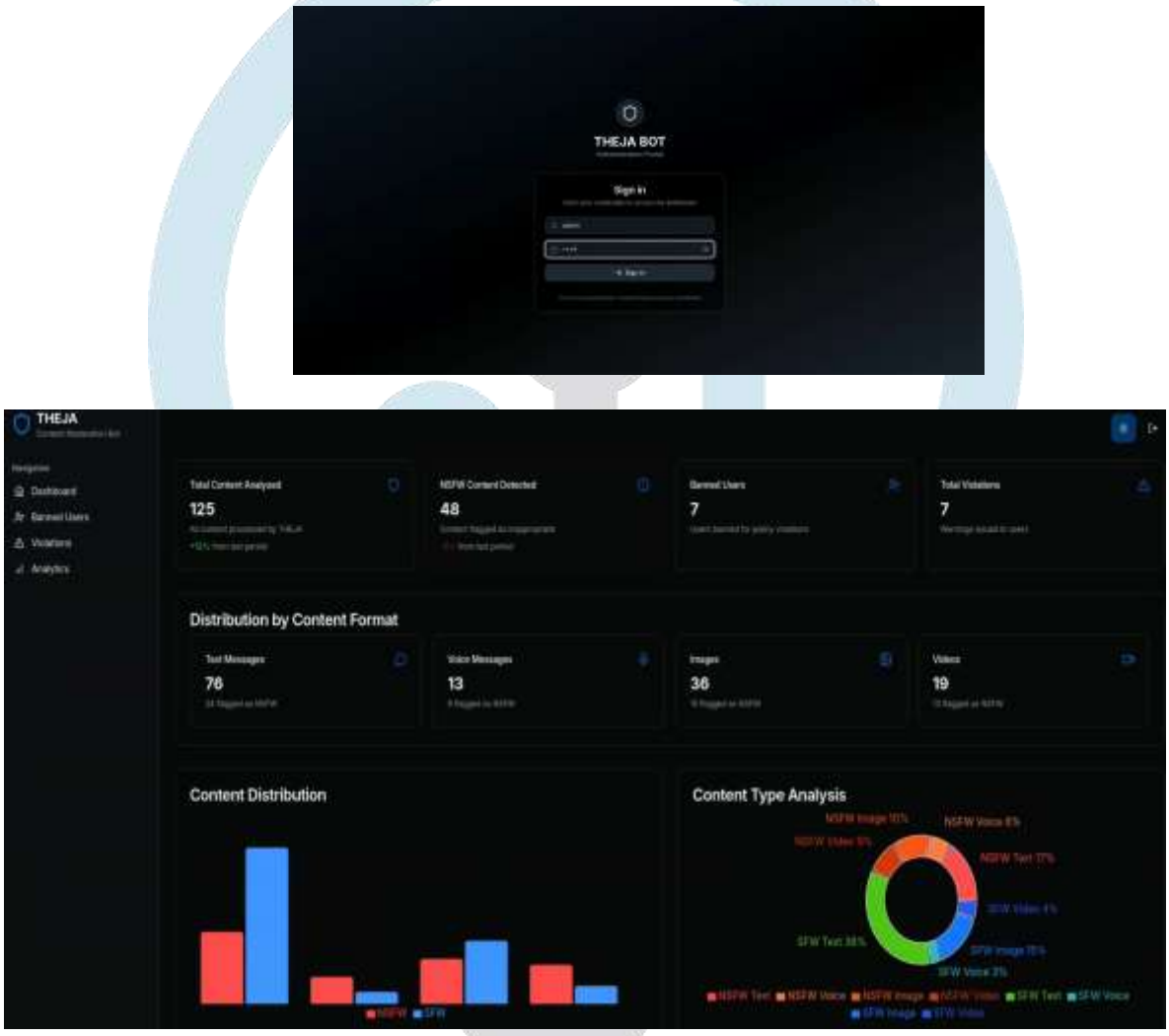

Fig. 7. Dashboard Overview - Login

Fig. 8. System Insights and NSFW Status

A. *User Violations and Banned Users*

The dashboard provides an organized view of flagged violations. - A dedicated section lists banned users with logs of repeated violations. - Graphical representations help evaluate group activity trends.

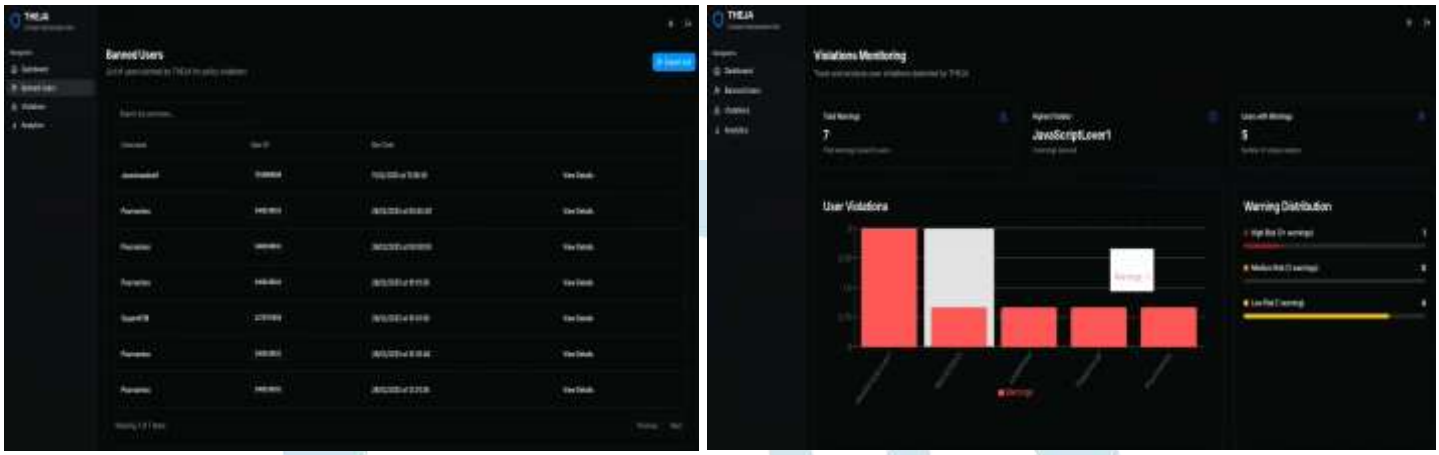

Fig. 9. Banned User List & user Violations

VIII. DEPLOYMENT AND SYSTEM INTEGRATION

The Telegram bot and dashboard are deployed using a cloud-based architecture to ensure scalability and efficiency. The backend is developed in Python and Node.js, handling AI model execution and message processing. - A database stores logs, flagged messages, and user interactions. - The bot interacts with the Telegram API for seamless communication. The dashboard runs on a web-based framework, allowing remote monitoring and control.

IX. CONCLUSION

The Telegram Group Management System was developed to address the increasing challenges of moderating large online communities. By integrating AI-driven content filtering mechanisms, the system effectively automates the detection and removal of inappropriate material, including text, images, videos, and audio.

If a user sends NSFW content (video or audio), the bot detects and removes the content immediately. The user receives a warning for each of the first three violations. However, if the preset flag count of three is exceeded, the bot automatically removes the violator from the group without any further warnings. All detected NSFW content is also removed.

Throughout the implementation and testing phases, the system demonstrated high accuracy, efficiency, and reliability in ensuring a secure and well-regulated digital environment. This proactive automation minimizes human intervention while maintaining the integrity of the group.

REFERENCES

[1] Pires and J. Borges, "Detecting Targeted Phishing Websites for Brand Protection and Cyber Defence Using Computer Vision," *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, Rome, Italy, 2023.

[2] Silpa, P. Prasanth, S. Sowmya, Y. Bhumika, C. H. S. Pavan, and M. Naveed, "Detection of Fake Online Reviews by using Machine Learning," *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Uttarakhand, India, 2023, pp. 71–77.

[3] Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji, and J. Porras, "Mitigation Strategies Against the Phishing Attacks: A Systematic Literature Review," *Computers & Security*, vol. 132, Sep. 2023.

[4] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka, "Survey and Taxonomy of Adversarial Reconnaissance Techniques," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–38, Jul. 2023.

[5] S. Khandelwal and R. Das, "Phishing Detection Using Computer Vision," *Computer Networks and Inventive Communication Technologies*, 2022.

[6] J. Zhou, Y.-F. Liu, and H.-L. Sun, "A Reputation Ranking Method Based on Rating Patterns and Rating Deviation," *2022 5th International Conference on Data Science and Information Technology (DSIT)*, pp. 1–6, 2022.

[7] G. Harris, "Combining Linguistic and Behavioral Clues to Detect Spam in Online Reviews," *2022 IEEE International Conference on e-Business Engineering (ICEBE)*, pp. 38–44, 2022.

[8] P. Rathore, J. Soni, N. Prabakar, M. Palaniswami, and P. Santi, "Identifying Groups of Fake Reviewers Using a Semisupervised Approach," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 6, pp. 1369–1378, Dec. 2021.

[9] S. Tang, L. Jin, and F. Cheng, "Fraud Detection in Online Product Review Systems via Heterogeneous Graph Transformer," *IEEE Access*, vol. 9, pp. 167364–167373, 2021.

[10] Conlin and U. Ruhi, "Current Research Landscape of Machine Learning Algorithms in Online Identity Fraud Prediction and Detection," *2021 IEEE International Conference on Technology Management Operations and Decisions (ICTMOD)*, pp. 1–6, 2021.

[11] L. Beltzung, A. Lindley, O. Dinica, N. Hermann, and R. Lindner, "Real-Time Detection of Fake-Shops Through Machine Learning," *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 2254–2263.

[12] Salahdine and N. Kaabouch, "Social Engineering Attacks: A Survey," *Future Internet*, vol. 11, no. 4, p. 89, 2019.

[13] Carta, G. Fenu, D. Reforgiato Recupero, and R. Saia, "Fraud Detection for E-commerce Transactions by Employing a Prudential Multiple Consensus Model," *Journal of Information Security and Applications*, vol. 46, pp. 13–22, June 2019.