Siddhesh Mengade, 2025, 13:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Anonymous Communication on Social Networks using AI-Powered Content Moderation: A Review

Siddhesh Mengade, Pranjali Chopade, Parth Tate, Shraddha Patil
Department of Computer Engineering,
Genba Sopanrao Moze College of Engineering, Balewadi, Pune, India

Abstract- The Anonify application facilitates anonymous message exchange, offering a platform for users to receive feedback or questions without fear of judgment. Despite the prevalence of anonymous platforms, a significant gap exists in content moderation to ensure safe and constructive communication. This project aims to address this gap by incorporating AI-based content moderation and fraud detection mechanisms. Leveraging technologies such as Next.js, NLP, Netlify, Auth.js, and Tailwind CSS, the application employs machine learning to filter inappropriate content and prevent abuse. The findings highlight the importance of maintaining a secure, anonymous feedback system, enhancing interactions in professional and social settings.

Keywords- Anonymous Communication, Schema, Validation, Auth.js, Zod, Spam & Fraud Detection, Feedback Collection, Content Moderation, Next.js, Foul Language Detection.

I. INTRODUCTION

The rise of digital communication has transformed the way individuals interact, yet it has also introduced barriers to honest and open communication. In contexts where feedback is vital for growth, anonymity can serve as a powerful tool to encourage openness and mitigate fear.

Anonify is an anonymous messaging platform designed to foster open, honest communication without the fear of identity disclosure. As digital interactions evolve, the need for secure, candid conversations has grown, especially in environments where feedback is crucial for personal and professional growth. Anonify offers a clean, user-friendly interface that allows individuals to ask questions, share thoughts, and receive feedback anonymously. Additionally, to maintain a safe and respectful environment for registered users, the platform integrates Artificial Intelligence for content moderation. This feature ensures that any sort of inappropriate, offensive, or spam/fraud messages are filtered out and blocked in real-time, protecting users from harmful content while encouraging constructive, honest exchanges. Built as a mobile-first, responsive web application, Anonify employs cutting-edge technologies to ensure smooth, secure interactions and tailored user experiences.

II. PROBLEM STATEMENT

In today's digital landscape, the lack of anonymous communication channels discourages individuals from sharing their honest opinions, particularly in workplaces, educational settings, and personal interactions. This absence hampers open dialogue, creating a culture of silence around sensitive topics, which stifles innovation and limits personal development. When individuals feel unsafe expressing dissenting or unconventional viewpoints, the exchange of ideas is hindered, preventing creative thinking and the emergence of new solutions. Additionally, without the option for anonymity, individuals miss out on valuable feedback essential for personal and professional growth.

© 2025 Siddhesh Mengade. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

To address these challenges, our application offers a secure, anonymous messaging platform that encourages open dialogue and fosters honest communication, enabling users to share their thoughts freely while ensuring privacy and safety through content moderation using Artificial Intelligence.

Objectives & Motivation

The primary objective is to provide a secure platform for anonymous messaging that fosters open dialogue and encourages users to express their thoughts freely. By minimizing the fear of exposure, users are more likely to share genuine feedback and engage in meaningful conversations. To maintain a safe environment, an essential objective is to implement AI-powered content moderation which will filter out inappropriate, offensive, or spam messages, ensuring that users can interact in a respectful space without the threat of harmful content.

Enhance User Communication: The primary objective is to provide a secure platform for anonymous messaging that fosters open dialogue and encourages users to express their thoughts freely. By minimizing the fear of exposure, users are more likely to share genuine feedback and engage in meaningful conversations.

Promote Constructive Feedback: Anonify aims to create an environment where users can give and receive constructive feedback. This objective is crucial for personal and professional growth, as it enables individuals to learn from one another in a supportive setting.

Integrate AI-Powered Moderation: To maintain a safe environment, an essential objective is to implement AI-powered content moderation. This feature will filter out inappropriate, offensive, or spam messages, ensuring that users can interact in a respectful space without the threat of harmful content.

One of the motivations behind Anonify is the increasing demand for secure, anonymous communication channels. With many individuals hesitant to voice their opinions due to social pressures, providing an anonymous platform helps bridge this gap. As digital communication evolves, there is a growing recognition of the need for tools that facilitate candid conversations. Anonify is motivated by the potential to contribute positively to this landscape by offering a dedicated space for open dialogue. The motivation to use advanced front-end and back-end technologies is rooted in the desire to create a responsive, user-friendly application whilst prioritizing seamless interactions and data security. Some motives can be briefed as follows.

- Address the Need for Anonymity: Anonify provides a secure platform for anonymous communication, allowing users to express opinions without fear.
- Support Digital Communication: The app promotes candid conversations, contributing positively to the evolving landscape of digital interactions.

Leverage Technology for User Experience: Advanced technologies create a responsive, user-friendly application that enhances interaction and ensures data security.

III. LITERATURE SURVEY

The concept of anonymous feedback platforms is not new. Early platforms such as Ask.fm and Ngl.link paved the way for similar tools. However, issues related to cyberbullying and misuse were prevalent in these systems, leading to a growing need for improved security and better user interfaces. Research on anonymous feedback platforms emphasizes the importance of features like robust content moderation and user-friendly interfaces. This is where Anonify differentiates itself by integrating advanced technologies like Zod for real-time database management and Auth.js for seamless and secure user authentication and anonymous session handling. And artificial intelligence algorithms like sentimental analysis and spam and fraud detection for moderating anonymous feedbacks.[III][IV][V]

These platforms have been studied for their impact on social behavior, with research indicating both positive and negative outcomes. While anonymous messaging can promote open communication, it also has potential risks, such as cyberbullying.

NGL.link (Not Gonna Lie): Inspired by older anonymous platforms, NGL rose to popularity by focusing on Instagram integration, allowing users to receive anonymous feedback via stories. It successfully handled privacy concerns, offering moderation tools to filter inappropriate content. But privacy concerns are well handled only on receiver's side. It currently provides a sneaky monthly subscription to receiver-users which allows them to view the identity of senders. Thus, eliminating the entire concept of 'anonymity'.

Sarahah: This platform became popular for anonymous constructive feedback but faced significant backlash due to its association with online harassment, ultimately leading to its removal from app stores.

WhisperLink: A Novel Anonymous Messaging Service for a Secured Data Communication (2024 IEEE/ACIS)

Authors: Martin Morales, Amit Boyina, Dev Kothari, Md Moniruzzaman and Ajmery Sultana

Description: WhisperLink is an anonymous messaging service designed to enhance privacy in digital communication. It operates on Google Cloud, allowing users to create temporary chat rooms that self-destruct after 24 hours, ensuring confidentiality without the need for logins. WhisperLink uses end-to-end encryption, ensuring that only intended recipients can read messages, and offers additional authentication through security questions to restrict access. By not storing any data beyond 24 hours, the platform guarantees private and transient conversations. Its flexible, user-friendly design makes it suitable for social, professional, and private interactions, standing out as a leading solution for secure, anonymous communication.[I]

Paper Findings: WhisperLink incorporates several key features to ensure a high level of privacy and security in digital communication.

- One such feature is end-to-end encryption, which ensures that only the intended recipients can read the messages, preventing unauthorized access even from the service provider.
- Another significant feature is the no-login requirement, which allows users to communicate without having to provide personal information or create accounts.
- Additionally, the temporary chat rooms and self-destructing messages provide an extra layer of security. These chat rooms automatically delete after 24 hours, ensuring that no residual data is stored on the platform.

Relevant Concepts: The concept of no-login requirement is implemented on anonymous senders' side in Anonify. This feature is particularly useful in maintaining anonymity and minimizing the amount of personal data stored by the platform. By eliminating the need for usernames or passwords, users' privacy is maintained and the risk of identity exposure or data leaks is reduced.

The Impact of Anonymity on Communication in the Metaverse (2024 IEEE COMPSAC)

Authors: Yoshifuru Nakayama and Kaoru Sumi

Description: The research paper investigates the impact of anonymity on communication in the metaverse, using virtual avatars to explore how immersion affects interactions. Four groups were tested under varying levels of anonymity (anonymous-anonymous, anonymous-non-anonymous, etc.). The results revealed no significant differences in immersive communication across the groups, suggesting that anonymity did not influence communication behavior. The study concludes that primitive visual anonymity helps prevent rude or violent behaviors, while higher levels of anonymity are necessary for de-individualization in the metaverse, potentially opening up different applications for anonymous interactions in virtual environments.[II]

Paper Findings: In this study, four distinct communication groups were created to examine how varying levels of anonymity affect interaction through virtual avatars. These groups were:

- Anonymous-Anonymous, where both participants communicated without revealing their identities;
- Anonymous-Non-Anonymous, where one participant remained anonymous while the other was identifiable;
- Non-Anonymous-Anonymous, where the roles were reversed, with one person identifiable and the other anonymous;
- and Non-Anonymous-Non-Anonymous, where both participants communicated without anonymity.

The research aimed to analyze how these different combinations influence immersion in communication within the metaverse. This suggests that other factors, such as visual anonymity, play a more critical role in inhibiting negative behaviors like rudeness or violence in virtual environments. The results imply that even superficial anonymity, where personal details are not shared, can foster respectful communication in immersive settings like the metaverse.

Relevant Concepts: Hence in a virtual environment like Anonify, negative behaviors like rudeness or violence could be prominent. The AI algorithms are implemented in order to avoid this negativity in real time. The relevant finding from this paper is the anonymous to non-anonymous communication scheme which has helped us structure our web application and its interactions.

IV. SYSTEM ARCHITECTURE & DESIGN DETAILS

The lack of anonymous communication channels hinders open dialogue, stifles innovation, and limits personal growth, as individuals often fear expressing honest opinions on sensitive topics. Without anonymity, important discussions are left unaddressed, and creative thinking is suppressed. To solve this, the anonymous messaging receiver application provides a secure platform for users to share thoughts freely while ensuring privacy. By integrating AI-powered content moderation, it filters inappropriate or harmful content, promoting a respectful environment where users can engage in honest, meaningful conversations without fear of judgment or privacy concerns.

1. System Architecture

The diagram below represents the system architecture demonstrating the flow of an anonymous messaging application. Here's a brief information about the working process:

- **User Registration:** A new user signs up, they confirm their account via a verification code sent on their e-mail. Once verified, their data is saved in the database.
- **Login & Dashboard:** Users log in and access their dashboard, where they generate a shareable link which can be used to receive messages from anonymous senders.
- **Link Sharing:** The user shares the link, allowing anonymous senders to submit messages.
- **AI Content Moderation:** Messages undergo AI-powered moderation to filter inappropriate content in real time. Thus, blocking any inappropriate content and not sending it to the user at all.
- **Message Reception:** The user receives, views, and reads messages (that pass the AI Content Moderator) through the platform.

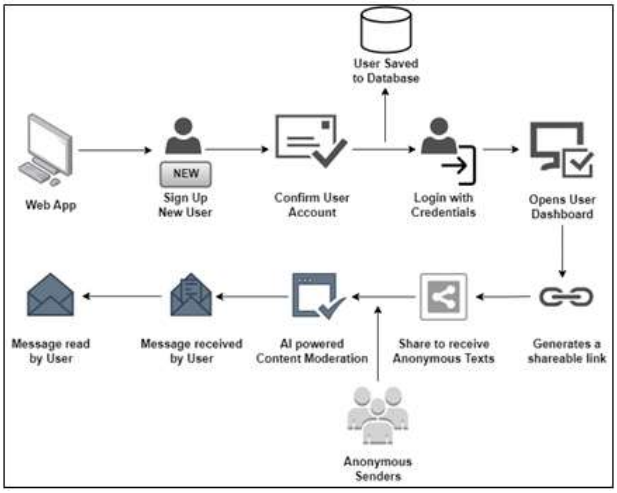

Figure 1 System Architecture

2. Database Design

The two schemas present in the database are, User Table and Message Table. And they outline the database schema for the anonymous messaging web application. Following explanation describes these schemas.

The User Table manages the user-related data such as login credentials, account status (verified or not), and whether the user is open to receiving anonymous messages. Whereas the Message Table manages the messages that are sent anonymously to users. Each message is linked to a user through a foreign key (user_id), and the content is stored securely.

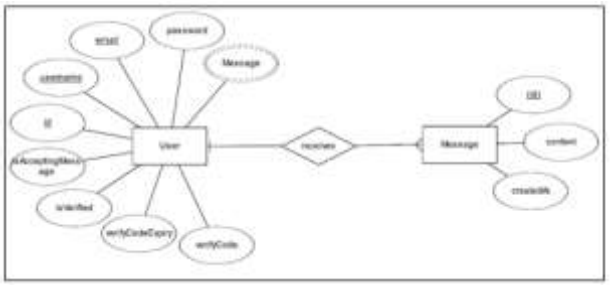

Figure 2 ER Diagram

3. AI-Powered Content Moderation

In the future, AI can play a crucial role in moderating content sent anonymously to non-anonymous receivers on the platform. A key method involves utilizing Sentiment Analysis to detect and filter harmful or inappropriate messages before they reach the recipient. AI models can analyze the sentiment of each message and classify it as positive, neutral, or negative, enabling the system to automatically flag offensive, abusive, or inappropriate messages.

Key Benefits of AI-Based Moderation

- **Real-time Detection:** AI algorithms like Naive Bayes, SVM, or LSTM can analyze messages instantly, flagging harmful content in real time.
- **Automated Filtering:** AI uses supervised learning to automate moderation, flagging extreme negative content like hate speech for blocking or review.
- **Context Awareness:** NLP models like BERT and GPT understand message context, reducing false positives by distinguishing between harmful and harmless expressions.
- **Scalability:** Deep learning models and RNNs enable efficient processing of large message volumes, ensuring timely moderation even with high traffic.
- **Adaptation to Evolving Language:** Word embeddings (e.g., Word2Vec, FastText) allow AI to adapt to new slang and evolving language, ensuring continued accuracy in detecting harmful content.

Additional AI Techniques for Moderation

- **Anomaly Detection:** Algorithms like K-Means clustering, Isolation Forest, or Autoencoders can be used to detect unusual patterns in messaging behavior, such as a sudden increase in negative content or signs of bullying, and flag these as potential threats.
- **Contextual Relevance:** AI models like BERT or T5 (Text-to-Text Transfer Transformer) can assess whether a message aligns with the overall conversation, helping to identify messages that are out of context or disruptive, such as spam or malicious content.

By incorporating AI-based content moderation techniques powered by advanced algorithms such as transformers, RNNs, SVMs, and deep learning models, the platform can ensure that messages sent anonymously are carefully filtered for harmful content before being delivered to non-anonymous receivers. This will create a safer and more positive user experience, reducing the likelihood of inappropriate or abusive interactions.

V. PROGRESS REPORT

Backend Development

In Next.js, API routing and route handlers are fundamental for building server-side functionality.

API Routes

API Routes allow you to create backend endpoints in your Next.js application, all within the /src/api directory. Each file in this directory becomes an API endpoint. These routes enable handling various HTTP methods such as GET, POST, PUT, DELETE, etc.

Example: A file at src/api/sign-up.ts becomes available at the endpoint /api/sign-up.

- **Request & Response:** Next.js API routes provide access to the req and res objects from Node.js, allowing you to handle requests and responses easily.
- **Middleware:** You can add middleware to authenticate users or protect routes.
- **Error Handling:** You can handle errors by returning appropriate HTTP status codes.
- **Limitations:** API routes don't support features like request body parsing by default; you need to manually parse them for certain content types.

Route Handlers

Route Handlers in Next.js are specific to the new app directory architecture, allowing more granular control over how a route responds to different HTTP methods. These handlers can be used to define custom logic for different HTTP verbs, such as GET, POST, and PUT, within the same file. They allow for easy handling of dynamic routes by using parameters.

Example: You can define route handlers by creating a file in app/api/route.js, which handles HTTP requests.[VIII]

- **Streaming and Buffering:** Route handlers support both, making it easier to handle large amounts of data, such as file uploads or real-time updates.
- **Advanced Capabilities:** They also offer advanced features like incremental static regeneration, server-side rendering, and dynamic rendering based on request parameters.

In summary, API routes provide a simple way to handle server-side logic in a Next.js app, while route handlers offer more granular control over handling HTTP requests in the newer app directory structure.

Suggesting Messages Using AI

GPT-3 and GPT-Neo are both transformer-based language models designed for natural language processing tasks. They are built on the transformer architecture, which uses self-attention mechanisms to understand relationships within a sequence of words.

GPT-3: Developed by OpenAI, GPT-3 is a state-of-the-art model with 175 billion parameters. It excels in generating human-like text across a variety of tasks due to its extensive training on diverse datasets. GPT-3 employs autoregressive decoding, where each word or token is generated sequentially, considering the input and previously generated tokens. This model is pre-trained on large datasets and can be fine-tuned for specific applications, such as text summarization or creative writing.

GPT-Neo: An open-source alternative to GPT-3, GPT-Neo, developed by EleutherAI, is designed to make large-scale language modeling accessible to the community. While not as powerful as GPT-3 in terms of parameter size, GPT-Neo models like the 2.7 billion parameter variant perform well on tasks like text generation and question answering. They are trained on public datasets and can be integrated for custom applications.

Algorithms Used for Suggesting Messages

- **Transformer Architecture:** Both models rely on self-attention mechanisms, which allow them to capture long-range dependencies in text efficiently. This makes the models adept at understanding the context of a conversation or a user prompt.
- **Autoregressive Decoding:** In message suggestion tasks, the models generate tokens sequentially. The generation of each token depends on the previously generated tokens and the input prompt. This ensures the output is coherent and contextually appropriate.
- **Prompt Conditioning:** The models are conditioned with a well-crafted prompt (e.g., "Generate three open-ended questions"). This guides the generation process to align the output with the desired structure and intent, such as creating engaging social media messages.
- **Tokenization:** Before processing, input text is divided into smaller units called tokens. The

model processes these tokens to predict the next token iteratively, forming a complete response.

- **Beam Search or Sampling:** For diverse and creative outputs, methods like Top-k Sampling or Nucleus Sampling are often employed. These introduce randomness by selecting from the top-ranked tokens at each step, ensuring variability in generated messages while maintaining relevance.

VII. CONTENT MODERATION USING AI

The GPT-3.5 Turbo model, which is available through platforms like Hugging Face, is primarily based on a transformer architecture, specifically the GPT (Generative Pre-trained Transformer) model, which uses unsupervised learning techniques to understand and generate human-like text. While GPT-3.5 Turbo itself is not explicitly designed for offensive content detection, it can be adapted for that task using various AI techniques.[VI]

Transformer Architecture (GPT-based models)

Self-Attention Mechanism: The core component of transformer models like GPT is the self-attention mechanism, which allows the model to consider the entire context of the text while making predictions. This context-awareness is useful when detecting subtle offensive content that might not be obvious in isolated words but is evident in the overall tone or context.

Pre-training and Fine-tuning: GPT-3.5 is pre-trained on a massive amount of text data and fine-tuned on specific tasks like offensive language detection. During fine-tuning, the model is trained on datasets labeled with offensive and non-offensive content, allowing it to learn patterns and features specific to harmful language.

Binary Classification: This is the most common technique where the model is trained to classify text as either "offensive" or "non-offensive." The training data consists of labeled examples, where each text instance is tagged with one of these two labels. Popular algorithms for text classification include:

- Logistic Regression
- Naive Bayes
- Support Vector Machines (SVM)
- Neural Networks (especially deep learning models like CNNs and RNNs)

Multi-class Classification: Instead of just two categories, the model can classify text into multiple categories such as "hate speech," "abusive," "profane," or "non-offensive." This allows for more nuanced detection of different types of offensive content.

Text Classification

Supervised Learning: In the case of offensive language detection, supervised learning techniques are often employed. The model is trained on a labeled dataset of text examples, where each example is tagged as "offensive" or "non-offensive." The model learns to classify text based on these labels.

Multi-Class Classification: Sometimes, the offensive content detection model might categorize text into multiple classes, such as "abusive," "profane," "hate speech," or "non-offensive." These models use softmax or similar activation functions to output probabilities for each category.

Sentiment Analysis

While sentiment analysis typically categorizes text as positive, negative, or neutral, it can be adapted to detect offensive language by associating specific negative sentiments or aggressive tones with harmful content. The model might be fine-tuned on datasets where aggressive or hateful speech is labeled as "negative sentiment," triggering the detection of offensive language.

Lexicon-Based Approaches: In these approaches, a predefined dictionary (or lexicon) of words associated with specific sentiments (e.g., positive, negative, neutral) is used. Sentiment scores are calculated based on the presence of these words in the text. Examples include:

VADER (Valence Aware Dictionary and sEntiment Reasoner): It is a lexicon and rule-based sentiment

analysis tool specifically tailored for social media text. VADER detects not just positive or negative sentiment but also the intensity and polarity of emotions, which can be useful in detecting offensive or aggressive content.

SentiWordNet: A lexical resource for sentiment analysis that assigns sentiment scores to words. This can be used to identify negative emotions or offensive language.

Rule-Based Sentiment Analysis: Rules are created to identify sentiment based on patterns in text. For example, the presence of certain words (e.g., "hate," "violence," "abuse") might trigger a "negative" sentiment label. This approach can be useful in detecting offensive or hateful language.

VIII. CONCLUSION

In conclusion, the development of Anonify represents a significant step toward creating a secure and user-friendly anonymous messaging platform.[IX] While we have made substantial progress in implementing core functionalities, including user registration, message sending, and basic content moderation, there is still work to be done to enhance the application further.[X] Our focus on integrating AI-powered content moderation has laid the groundwork for a safer user experience, effectively filtering out inappropriate content and ensuring a positive environment for our users. And that is what we aim to work on for the next stage of development. Moreover, the project has provided valuable insights into the complexities of full-stack development, particularly in leveraging Next.js for seamless server-side rendering and efficient API routing. Although some advanced features, such as OTP verification, are yet to be fully integrated, our progress thus far underscore the project's potential. This project not only aims to foster anonymous communication but also emphasizes the importance of user safety and trust in digital interactions. In essence, Anonify is poised to evolve into a comprehensive platform that balances anonymity with accountability, creating a space where users can express themselves freely whilst feeling secure and spreading positivity. As we continue to develop and enhance Anonify, we are excited about the positive impact it will have on fostering open communication in a safe environment.

REFERENCES

1. M. Morales, A. Boyina, D. Kothari, M. Moniruzzaman, and A. Sultana, “WhisperLink: A Novel Anonymous Messaging Service for a Secured Data Communication,” in 2024 IEEE/ACIS, 2024.
2. Y. Nakayama and K. Sumi, "The Impact of Anonymity on Communication in the Metaverse," in Proceedings of the IEEE COMPSAC, 2024.
3. Lee, Y., & Kim, K. H. (2020). De-motivating employees’ negative communication behaviors on anonymous social media: The role of public relations. Public Relations Review, 46(4), 101902. https://doi.org/10.1016/j.pubrev.2020.101902
4. Milosevic, T., Verma, K., Carter, M., Vigil, S., Laffan, D., Davis, B., & O’Higgins Norman, J. (2023). Effectiveness of artificial intelligence–based cyberbullying interventions from youth perspective. Social Media + Society, [Article number]. https://doi.org/10.1177/20563051231156469
5. Wang, H., Hee, M. S., Awal, M. R., Choo, K. T. W., & Lee, R. K.-W. (2023). Evaluating GPT-3 generated explanations for hateful content moderation. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).
6. Liu, B. (2012). Sentiment analysis and opinion mining. In Synthesis Lectures on Human-Centered Informatics (Vol. 5, No. 1, pp. 1-167). Morgan & Claypool Publishers.
7. Gorwa, R. (2019). Content moderation, algorithmic governance, and the new politics of the internet. Internet Policy Review, 8(2). doi:10.14763/2019.2.1414
8. Auth.js. (n.d.). Documentation. Retrieved September 21, 2024, from https://authjs.dev

9. Next.js. (n.d.). API routes: Create API endpoints in Next.js. Retrieved August 11, 2024, from https://nextjs.org/docs/api-routes/introduction
10. Postman. (n.d.). Postman documentation. Retrieved September 01, 2024, from https://learning.postman.com/docs/getting-started/introduction/