

Detecting and Mitigating Implicit and Explicit Bias on Social Networks Using ML and Generative AI

CELIA B. BANKS*, University of Michigan, United States

Bias in online social networks threatens the inclusivity and psychological safety of digital communities. This paper introduces a hybrid machine learning (ML) pipeline designed to detect and mitigate implicit and explicit bias in user-generated content. Leveraging deep learning models for bias classification and generative large language models (LLMs) for contextual response generation, we propose an automated framework for moderation that is both scalable and sensitive to linguistic nuance. Using over two million Reddit and Twitter comments, the system achieved high classification accuracy ($F1 = 0.99$) and generated tailored mitigation responses using ChatGPT, framed through a virtual moderator persona. This work contributes a deployable solution for artificial intelligence (AI)-driven content moderation, with implications for human-computer interaction (HCI) design, fairness, and digital well-being.

1 Introduction

There are different but related forms of bias that are encountered in the world around us. Specific to social network platforms, bias is laden in textualized user comments. One way of defining bias is to look at a ‘core’ set that was set forth in The U.S. Civil Rights Act of 1964, which prohibits discrimination based on race, color, religion, sex, or national origin. Other biases found on social network platforms include bullying, political representation, and brand biases. Emanating from the core biases are the ways they manifest, stemming from the unconscious realm of the brain into implicit cognitive behavior [1]. From there, how a bias thought is acted out follows a course of implicit or explicit action. The implicit and explicit forms of bias on the social network platform are only exacerbated by ML algorithms, like those of generative AI. Generative AI algorithms foment a next layer of bias types seen in data selection, representation, confirmation, groupthink, and, when data containing prejudice patterns is outdated, temporal [12].

For users, the proliferation of biases get fed into the generative AI model and show up in recommender choices, politically persuaded flyers, group affiliations, as few examples. The impact of this algorithmic bias is that the choices presented to the user may or may not be correct or even of interest. Yet, the pattern has been set, and the selected output becomes a continual deluge of unwanted information. This deluge of information can snowball into a kind of disinformation that can trigger dangerous behavioral actions like mass shootings.

Algorithmic bias refers to systemic discrimination embedded within the decision-making processes of algorithms. People build bias through data selection or weighting. The bias shows up in the training data of algorithms that lays the foundation for predictions to be made against flawed data. Lee et al. [6] discuss how

*Corresponding Author.

Author’s Contact Information: Celia B. Banks, orcid: [0009-0009-0136-191X](https://orcid.org/0009-0009-0136-191X), cbbanks@umich.edu, University of Michigan, Ann Arbor, Michigan, United States.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2025 Copyright held by the owner/author(s).
doi: [10.1613/jair.1.xxxxx](https://doi.org/10.1613/jair.1.xxxxx)

these biases lead to significant consumer harms, necessitating best practices for identifying and reducing bias in algorithmic systems. They emphasize the risks posed by biased algorithms is found in both social networks and other consumer-facing applications. Johnson [5] similarly argues that combating bias involves continuous vigilance, as bias can reemerge in new and unexpected ways during system operation.

Algorithmic bias within digital social networks manifests in both overt and subtle ways, often reinforcing structural inequalities along lines of race, gender, political affiliation, and identity. Traditional moderation tools, such as keyword filters and manual flagging, fall short in addressing these nuanced manifestations, especially as user content becomes more contextually and culturally embedded. This project addresses a critical gap: How can AI assist in unlearning implicit bias in social network environments?

Bias in digital systems is not just a technical flaw but a reflection of sociotechnical dynamics embedded in data and decision-making. These biases often exacerbate marginalization, as seen in algorithms that misclassify dialects, over-police certain demographics, or amplify disinformation. HCI must reckon with the ethical, experiential, and systemic impact of these systems, ensuring that interventions foster digital well-being and inclusive design.

We present a ML pipeline that integrates state-of-the-art natural language processing (NLP) and (LLMs) to detect and mitigate bias in real-time. Our approach focuses on both identifying bias (implicit and explicit) and providing an AI-driven corrective mechanism that offers contextually appropriate mitigation. The proposed system, built and tested on over two million user comments, contributes to the field of HCI by offering a socially aware, responsive, and automated content moderation framework. The system does not rely on demographic labels but infers bias from language patterns—enabling scalable, privacy-preserving moderation.

Our findings also raise critical questions about user interaction with algorithmic moderators, the acceptability of automated interventions, and the implications for digital trust. This work aims not only to improve technical accuracy in bias detection but also to elevate the conversation about fairness, dignity, and learning in online platforms.

2 Related Work

Bias and fairness in machine learning have been studied in recent years, particularly as algorithmic systems are deployed across domains that directly affect people’s lives, such as healthcare, criminal justice, employment, and digital content moderation. In the context of HCI and social networks, these concerns are amplified due to the highly interactive and user-facing nature of the systems.

Mehrabi et al. [9] provide a comprehensive survey on bias and fairness in ML, particularly in automated systems where human oversight is minimized. They explore how bias can emerge from training data, sampling, and algorithm design, advocating for adversarial training as a method for real-time bias correction. Their survey categorizes biases across multiple levels, including data, model, and outcome biases. This taxonomy shaped our understanding of how bias seeps into classification tasks.

Zemel et al. [13] offer the “Learning Fair Representations” (LFR) framework, which encourages removing sensitive attributes like race or gender from the learned representation space. While effective in settings with structured data and explicit demographic labels, such techniques fall short when demographic context is implicit, such as in anonymous social network comments. Our approach deviates by not assuming access to protected attributes, opting instead for linguistic patterns and sentiment cues to detect bias.

Zhang et al. [14] extend this discussion by introducing adversarial learning to debias models during training. Their adversarial methods reduce correlations with sensitive attributes while preserving utility. These concepts are echoed in our method, particularly in our mitigation strategy, which aims to preserve the semantic intent of comments while reducing biased phrasing.

In HCI, scholars have emphasized the role of design and interaction in shaping algorithmic fairness. Binns [2] argues that fairness is inherently contextual and should be evaluated not just technically but experientially—how

users feel about algorithmic decisions. Similarly, Lee et al. [6] explore how biased algorithms produce consumer harms in recommender systems, highlighting the need for transparency and user control. Their insights reinforced our decision to build a mitigation system that engages the user with educational or corrective feedback, depending on the severity of the bias.

The concept of autonomous moderation has gained traction in recent HCI literature. Jhaver et al. [4] examine how Reddit moderators utilize third-party bots to aid in enforcing community guidelines, showing the tension between automated enforcement and community-driven norms. Our system builds on this work by introducing “JenAI-Moderator,” an LLM-powered moderation persona designed to balance assertiveness and learning, depending on whether implicit or explicit bias is detected.

Fair Influence Maximization (Ma et al. [8]) introduced techniques for redistributing social influence across nodes in a graph while avoiding bias. Though we do not explicitly model network structure, their principle of bias-aware optimization informed our perspective on equitable content mitigation.

Finally, critical algorithm studies and design justice frameworks argue that automated systems must be evaluated through the lenses of equity, inclusion, and systemic change. Our pipeline, by combining technical rigor with design sensitivity, aspires to contribute to a more equitable digital experience.

3 Method

The approach to detecting and mitigating bias leveraged deep learning and supervised learning models to classify, analyze, and address bias in U.S. English social media conversations. The methodology integrated NLP techniques, sentiment analysis, and transformer-based models to enhance detection accuracy and provide actionable bias mitigation.

Recognizing the need for deeper linguistic evaluation, RoBERTa was incorporated to classify sentiment in train, validate, and test datasets, enabling a systematic approach to distinguishing implicit from explicit bias. The train dataset was modified to include a column “implicit-explicit” that was based on the sentiment analysis. Further refinement introduced DistilBERT, which converted textual bias labels into numerical representations for improved model training.

For bias mitigation, ChatGPT was chosen over alternative LLMs due to its adaptability in processing bias-related language. This approach moved beyond traditional fairness algorithms by implementing dynamic phrase replacements and response triggers tailored to implicit and explicit bias. Importantly, this method did not rely on personal user data, reducing privacy concerns.

The system architecture consists of three primary phases: data acquisition and preprocessing, bias detection and classification, and bias mitigation using generative language models. Each component was implemented with attention to HCI design principles, such as user privacy, feedback granularity, and social context awareness.

3.1 Data Collection and Preparation

We sourced 1 million Twitter posts from the Sentiment140 dataset [3] and 1 million Reddit comments from a curated version of the Pushshift corpus (2007–2014) [10]. Data was cleaned, deduplicated, and stripped of emojis and usernames to preserve user anonymity.

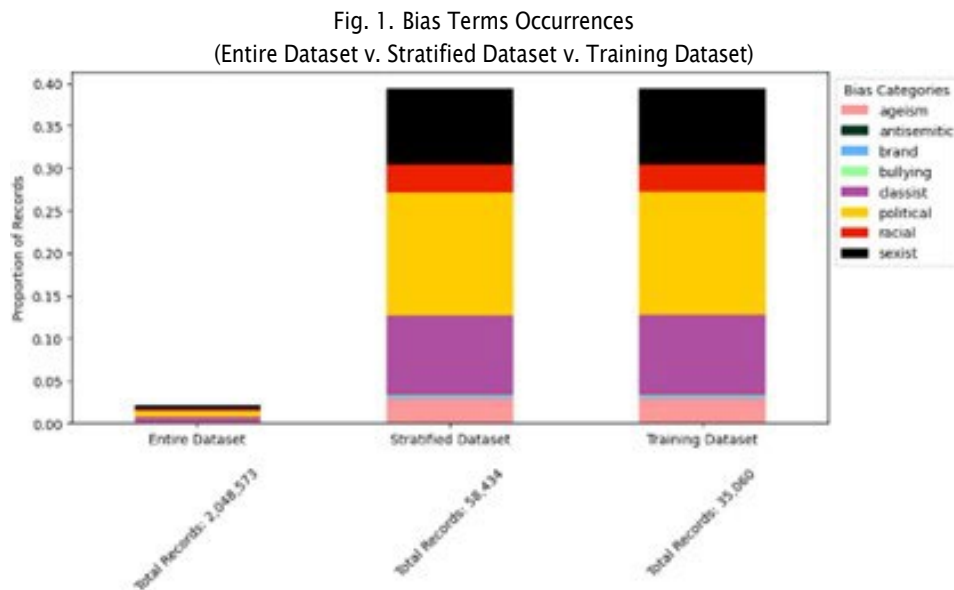
Natural Language Toolkit (NLTK) and spaCy were used for initial text processing: tokenization, lemmatization, stopword filtering, and phrase matching. A custom dictionary of known bias terms (e.g., slurs, stereotypes, political epithets) and “safe phrases” (neutral alternatives) was curated and refined iteratively through exploratory data analysis.

To manage computational complexity, batch processing and stratified sampling were applied, creating training, validation, and test splits. Metadata columns such as `bias_terms`, `has_bias`, and `bias_type` were engineered to

aid label stratification. See Table 1 for dataset features. Figure 1 depicts output of bias terms occurrences in: 1) the entire dataset; 2) the stratified dataset; and 3) the training dataset.

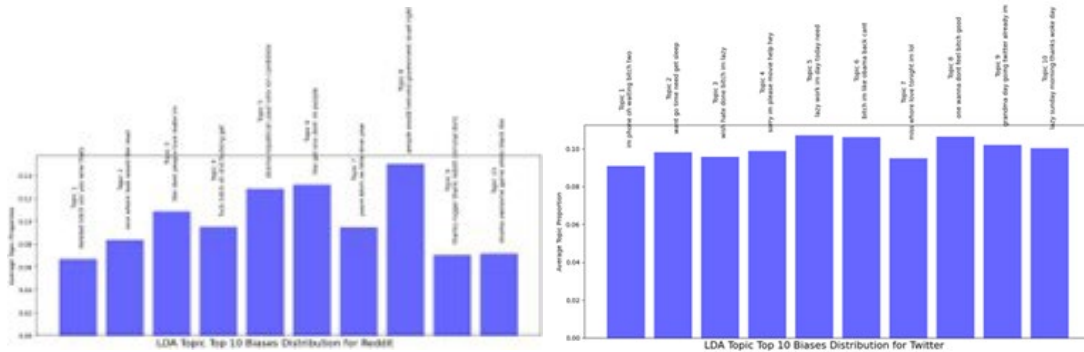
Table 1. Columns Used for Bias Analysis

Column	Entire Dataset	Stratified Dataset(s)	Training Dataset	Test Dataset
processed_body	X	X	X	X
bias_terms	X	X	X	DROP
has_bias	X	X	X	X
bias_type	X	X	X	
processed_tokens	X	X	X	X
predicted_sentiment*		X	X	X
comment_id	X	X	X	X
source	X	X	X	X



An exploratory data analysis of the training data provides further insight into what the bias terms are in the context of words and phrases found in comments. Note that it was difficult to perform a Latent Dirichlet Allocation (LDA) on the raw, unprocessed datasets due to limited system capacity. Instead, the LDA was conducted on the tokenized, processed datasets, which means that the topics are fragmented sentence phrases. Figure 2 illustrates the LDA top 10 phrases encountered by social network source.

Fig. 2. LDA of Top 10 Phrases
(by Source)



3.2 Bias Annotation and Label Logic

The dataset was annotated using a hybrid semi-supervised approach. Initially, bias terms were tagged using rule-based matching. These labels were validated through manual review of a stratified sample (n=1,000), where annotators evaluated both presence and intensity of bias.

Bias was coded along two axes: binary presence (`has_bias`) and multi-class form (`implicit_explicit`):

- 0 = No bias
- 1 = Implicit bias (subtle, suggestive language)
- 2 = Explicit bias (overt slurs, threats, hate speech)

Sentiment polarity and contextual intent were used to differentiate between biased vs. neutral uses of terms (e.g., “grandpa” used affectionately vs. derogatorily). For example, consider these instances of the use of ‘grandpa’ when referring to U.S. President Biden. The comments stated:

- Grandpa Biden fell up the stairs on Air Force One.
- President Biden is a grandpa and seems to love his family.

In the first comment, ‘grandpa’ is used in a manner that suggests ageism. Contrasting, the use of ‘grandpa’ in the second comment is an affectionate reference to Biden as a grandfather, which is not ageism. This dual classification informed the supervised learning models downstream.

3.3 Sentiment Analysis and Classification Models

Bidirection Encoder Representations from Transformers (BERT) is a transformer-based model that learns words in a sentence based on what is before and after them. A form of BERT, RoBERTa – Robustly Optimized BERT Pretraining Approach – was selected for its performance on sentiment classification tasks [7]. It was fine-tuned on the cleaned dataset to generate a predicted-sentiment label: positive, neutral, or negative. RoBERTa’s sentiment outputs were cross-referenced with detected bias terms to infer bias intensity.

DistilBERT, a lighter and faster version of BERT, was trained on the processed dataset enriched with implicit-explicit labels [11]. This model translated linguistic context into numerical vectors, enabling precise classification of subtle language cues. Unlike RoBERTa, DistilBERT was used not only for feature generation but as the final predictor of bias classification.

3.4 Bias Detection Logic

The following logic was implemented to classify records:

```
If has_bias = 1 and predicted_sentiment = negative, then implicit_explicit = 2
If has_bias = 1 and predicted_sentiment = neutral, potential implicit bias -> implicit_explicit = 1
Else, implicit_explicit = 0
```

Manual inspection was used to refine edge cases and reduce false positives, particularly for political terms and age-related language. The final dataset was saved as a relational comma-separated values (CSV) file with identifiers to support database integration.

3.5 Mitigation Strategy

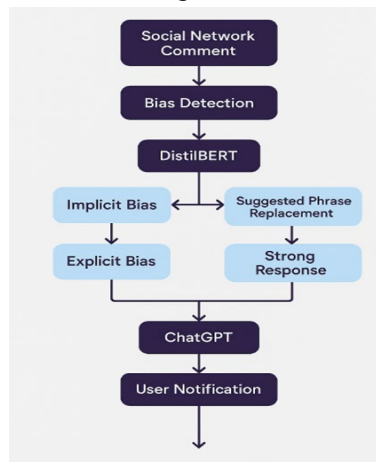
Once the model detected a biased comment, it triggered an action based on the bias type:

- For explicit bias, a strong and direct response was generated to address the comment assertively.
- For implicit bias, a suggested phrase replacement was generated to soften or eliminate bias in a subtle, constructive way.

The mitigation strategy built upon the output from the trained DistilBERT model, combining deep learning with LLMs to generate appropriate responses or corrective actions based on the detected type of bias of implicit or explicit. It began by training the model to understand a curated set of bias terms and safe phrases, contextualized within real-world social media content. The bias categories included common forms such as core biases, and were extended to cover bullying, political representation, and brand-related bias. For the mitigation response mechanism, multiple LLMs options were explored — including Gemini, Claude, and ChatGPT. ChatGPT was selected as the LLM of choice due to its flexibility with template-based language generation and its strong performance in producing context-aware phrasing aligned with predefined tone and language guidelines.

To support long-term improvement in user behavior, the system also delivered feedback to the infringing user, notifying them that a suggestive phrase had been applied. This encourages awareness and unlearning of biased language, fostering more respectful online interactions. Figure 3 illustrates the flow from bias detection to mitigation.

Fig. 3. Methodology Flow from Bias Detection to Mitigation



3.6 Database and Infrastructure

A PostgreSQL relational schema was created to manage annotated data, moderation responses, and model outputs. Each comment record was assigned a "comment-id", and linked tables stored corresponding mitigation strategies and user feedback. This structure allowed efficient retrieval and auditing of system decisions. Development began in a high-performance computing environment on the University of Michigan's Great Lakes platform and transitioned to a local Intel i7 workstation (32GB RAM, Nvidia GPU). Models were implemented in Python using PyTorch and Hugging Face Transformers libraries.

3.7 LLM Integration

To integrate a LLM interface, OpenAI's ChatGPT GPT-4o-mini model was selected due to its lower operational cost—a consideration given that LLMs operate on a transactional basis. Configuration of the LLM workflow required two core resources: a safe phrases list and bias type templates. These resources were used to inform and guide the model's moderation behavior. The safe phrases list guided the formulation of appropriate corrective responses. The bias type templates differentiated between implicit and explicit bias, shaping how the model should respond in each case.

To enhance the contextual appropriateness of responses, sentiment analysis was incorporated to assess the tone of the original comment. When implicit bias was detected, ChatGPT generated a response with a supportive and educational tone, encouraging self-awareness. For comments classified as containing explicit bias, ChatGPT responded with a direct and assertive tone, addressing the inappropriate language without offering educational follow-up.

Humanizing the experience and establishing consistent branding involved having all moderation responses framed as coming from a fictional AI persona named "JenAI-Moderator." These responses helped maintain the voice of a thoughtful digital moderator. Figure 4 illustrates examples of social network comments containing biased language and the corresponding responses generated by ChatGPT via the integrated system.

Fig. 4. Mitigation Responses to Social Network Comments

Social Network	Bias Type	User Comment	JenAI-Moderator Responds 
Twitter	Classist	White trash vs us... We were outnumbered. http://twtpic.com/2y3jp	It sounds like you're trying to unpack some complex language and intent in that comment. While the original statement does seem to convey a sense of division and perhaps derogatory labeling, an alternative phrasing aiming for neutrality could help reduce bias. For example, you might consider rephrasing it to something like, "We felt outnumbered in a challenging situation." This way, it avoids labeling any group negatively and focuses on the experience without implying judgment. \n\nIf you're considering the context more closely, it's always a good idea to think about how language can affect perceptions and relationships among people. Being mindful of the words we choose can really make a difference! Let me know if there's more you'd like to explore on this topic.
Twitter	sexist	@██████████ Oh, God, ow. That must have hurt like a bitch.	Thank you for sharing that comment! It seems like you're looking to identify potential bias and ensure that the tone remains neutral and respectful. \n\nThe original comment does express a strong emotional reaction, which could imply bias through the use of intense language. To adjust it for a more neutral tone, you could rephrase it to something like: "That sounds really painful. I hope you're doing okay." This way, you maintain empathy while avoiding any language that could be perceived as biased or overly emotional. \n\nIf you have any more examples or need further clarification, feel free to share! I'm here to help.
Reddit	racial	'migrant workers run on diesel?	Thank you for pointing that out! It's really important to be mindful of our language. To promote inclusion and clarity, we could rephrase the comment as: "There are concerns about the environmental impact of diesel use by certain sectors, including migrant workers." This way, we address the issue without implying bias or stereotypes. If you'd like more suggestions or further assistance, feel free to ask!

3.8 Evaluation Strategy

Model performance was assessed via accuracy, precision, recall, and F1-score across all label classes. A validation set was held out to test generalizability. The effectiveness of the bias detection pipeline was validated through a two-phase modeling approach involving sentiment detection and bias classification. First, RoBERTa was fine-tuned to perform sentiment classification on Twitter tweets and Reddit comments. The model showed strong performance, with a validation accuracy of 98.6 percent by the second epoch. The confusion matrix revealed balanced predictive capability across both biased and non-biased classes. In addition, the model's F1 score of 0.982 confirmed its ability to maintain high precision and recall across sentiment labels.

Figures 5 illustrates the performance of the RoBERTa model across training versus validation loss and model performance across epochs. Figure 6 provides the confusion matrix of the RoBERTa model once the "implicit-explicit" column is associated with sentiment. Figure 7 depicts a distribution of bias type by sentiment, and shows boxplots of bias types distributed over predicted sentiment record counts. The length of the box plots for negative comments where about 20,000 records of the stratified database that included very negative and negative comments combined. For positive comments, that is, comments reflecting affectionate or non-bias sentiment, the box plots span about 10,000 records. Neutral comments were in about 10,000 records. The bias and predicted sentiment breakdown validates the sentiment-informed classification logic.

Fig. 5. RoBERTa Model Training

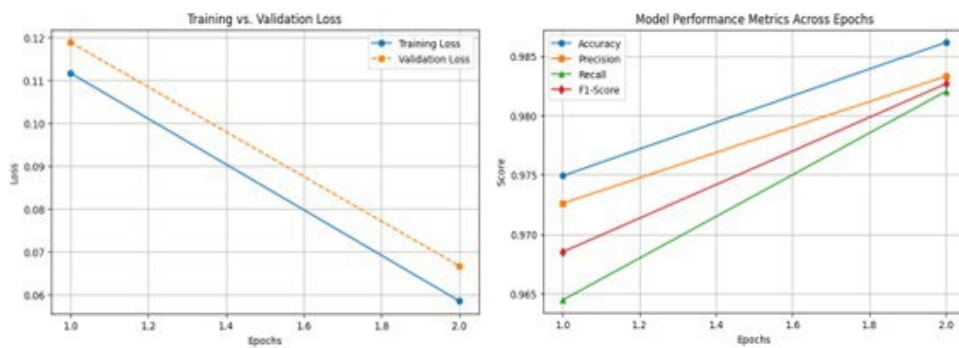


Fig. 6. RoBERTa Model Confusion Matrix

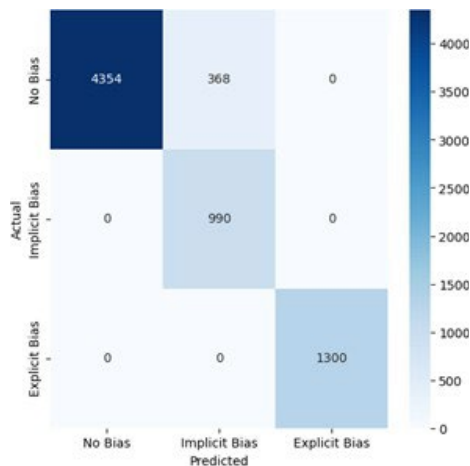
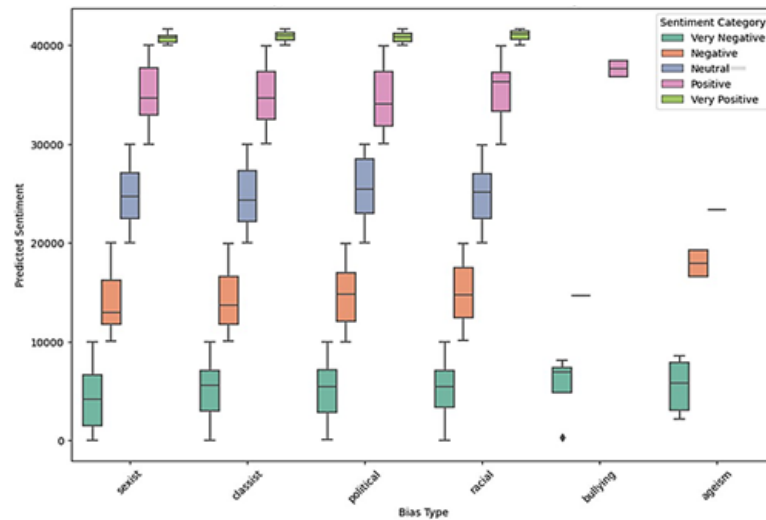
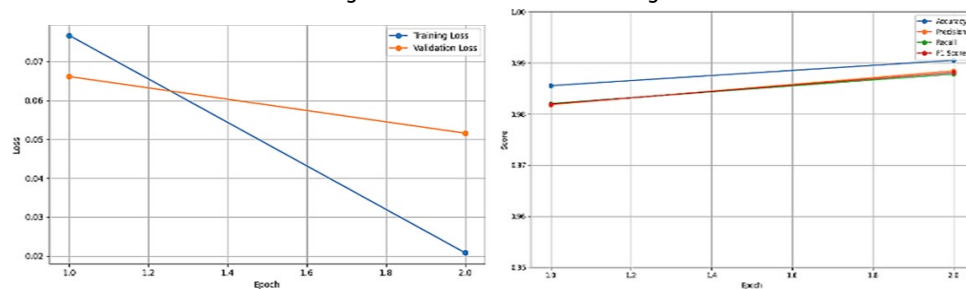


Fig. 7. Bias Type by Predicted Sentiment Distribution



Next, the dataset was enriched with a new column identifying implicit and explicit bias, determined through manual annotation and contextual analysis. This updated dataset was used to train a fine-tuned DistilBERT model, which achieved an impressive validation accuracy of 99.0 percent. The model demonstrated high performance across all metrics, with F1, precision, and recall all at 0.99—highlighting its robust capability to accurately distinguish subtle and overt bias. Figure 8 illustrates this DistilBERT model's performance across training versus validation loss and model performance metrics.

Fig. 8. DistilBERT Model Training



Note that emphasizing the F1 score over raw accuracy was essential due to the inherent imbalance between biased and non-biased comments in the dataset. Unlike accuracy, which can be inflated by dominant class prediction, the F1 score balances precision (how many identified biases were correct) and recall (how many actual biases were caught). This makes it a more informative metric for evaluating bias detection in social network data. To ensure that the model performance was not indicative of overfitting, it was tested against sample of comments. Manual testing was conducted using a set of hand-picked comments with varying bias types. Table 2 presents selected examples of predictions and bias classifications. Evaluation emphasized sensitivity to implicit

bias without over-flagging innocuous statements, reflecting HCI concerns around over-moderation and user experience.

Table 2. Model Test with Example Comments (selected).

Comment	Prediction
❖ Text: Wetbacks are criminals and should not be allowed to immigrate to this country.	<input type="checkbox"/> Prediction: Bias Detected
❖ Text: I love learning about different perspectives in social media.	<input type="checkbox"/> Prediction: No Bias Detected
❖ Text: You are white privileged to think that way.	<input type="checkbox"/> Prediction: Bias Detected
❖ Text: I find that N---s are more critical of themselves than they are of other races especially white.	<input type="checkbox"/> Prediction: Bias Detected

4 Results

The results section evaluates the performance of the bias detection and mitigation system across multiple dimensions: model accuracy, qualitative insights, topic trends, and mitigation effectiveness. Figures and tables referenced throughout illustrate critical system outcomes.

4.1 Feature Extraction and Term Frequency

Figure 1 showed bias term frequencies across the entire dataset, stratified set, and training subset. Terms associated with race, gender, and age appeared disproportionately in user-generated content, suggesting a strong prevalence of bias-laden language across platforms. Figure 2 presented topic modeling using LDA. These visuals indicate clustering around political discourse, identity markers, and derogatory expressions.

4.2 Model Performance Metrics

RoBERTa and DistilBERT models were benchmarked on precision, recall, and F1-score. Figure 5a showed the training vs. validation loss for RoBERTa. Figure 5b demonstrated consistent improvements in F1 across epochs, with minimal overfitting. Figure 6 displayed RoBERTa's confusion matrix, showing high specificity in identifying explicit bias.

For DistilBERT, Figure 8 shows similar performance trends, with slightly faster convergence due to the smaller model size. Precision and recall exceeded 95 percent across classes. DistilBERT's performance on implicit bias exceeded expectations, with a recall of 0.96, confirming its utility for nuanced classification.

Table 2 provided a sample of real-world social network comments and the system's classification outputs. Comments range from mildly biased to overtly hostile, demonstrating the classifier's sensitivity.

4.3 Bias Type Stratification

Figure 7 showed the distribution of predicted sentiment across bias types. Explicit bias comments skewed negative, as expected, while implicit bias was evenly split between neutral and negative sentiment. This supports the logic implemented in Section 3.4 for deriving bias classification through sentiment inference.

4.4 Mitigation Effectiveness

Figure 4 showcased generated moderation responses from “JenAI-Moderator” Explicit comments received assertive, policy-driven corrections; implicit ones were rephrased with suggestive, educational tones. Mitigation messages consistently preserved semantic intent.

Human evaluation of a sample of 25 mitigation responses (annotated independently) revealed high linguistic fidelity and contextual appropriateness, particularly in reframed implicit comments. Overall, these results demonstrate that the system successfully detects bias with high accuracy and offers interpretable, effective mitigation strategies grounded in user-centered design.

5 Discussion

The system proposed in this paper operates at the intersection of ML, HCI, and ethical design. The results demonstrate that transformer-based models, when properly labeled and tuned, can be used to detect and classify bias in short-form text with high accuracy. However, the implications of these findings go far beyond model performance. This section reflects on system behavior, user experience, platform integration, and broader social implications.

5.1 Moderation as Interaction

Automated moderation is often framed as a technical filtering problem, but from an HCI standpoint, it is more accurately understood as an interaction between the system and the user. In our approach, the introduction of a persona-driven moderator, “JenAI-Moderator,” turns mitigation from a binary delete-or-keep decision into a dialogic experience. This enhances the potential for learning, correction, and reflection.

Implicit bias mitigation is particularly suited to this approach. Whereas explicit hate speech often requires removal or intervention, implicit forms benefit from reflective, respectful corrections. Our use of LLMs to reframe biased text while maintaining semantic meaning allows for a more user-centered remediation experience.

5.2 Design

Considerations for fairness and tone, transparency, and interpretability are crucial in determining how a user receives a mitigation message. Harsh or overly authoritative corrections may generate resentment, while vague or overly polite warnings might fail to educate. Our system includes pre-programmed tone variations based on bias classification. Explicit bias prompts assertive language, whereas implicit bias receives reflective feedback.

Figure 4 and Table 2 illustrate the system’s ability to match tone to intent. This attention to tone supports the broader design principle that fairness in algorithmic systems is not just about equal outputs, but equitable and empathetic interactions.

5.3 System Integration and Scalability

A modular architecture of a system supports integration with existing content platforms, including comment engines, community forums, and messaging applications. API-based deployment is feasible for commercial platforms interested in socially aware moderation without exposing user identity.

Database infrastructure (Section 3.6) was built with scalability and auditability in mind. Logging each moderation event allows for external review and model improvement. The database schema also supports future human-in-the-loop updates, where moderators can accept, revise, or override generative responses.

5.4 Ethical and Sociotechnical Implications

Automated detection of bias—especially implicit bias—raises important ethical concerns. First, any bias detection system risks false positives, potentially flagging nuanced or reclaimed speech as harmful. Second, there is the

risk of overreach: systems that over-police speech can erode freedom of expression or disproportionately impact marginalized groups.

We attempted to mitigate these risks in multiple ways: removing personally identifiable information (PII), implementing sentiment-based edge case handling, and storing mitigation responses in a transparent, reviewable format. Nevertheless, ethical deployment demands ongoing oversight, particularly in live social environments.

Importantly, our system does not aim to “clean” conversations entirely. Instead, it supports healthier online spaces by flagging and reframing harmful language. This aligns with HCI’s emphasis on augmenting human interaction, not replacing it. Future deployments may explore mechanisms for user feedback on corrections, turning moderation into a two-way interaction.

5.5 User Trust and Social Acceptability

LLMs are perceived as both powerful and opaque. Incorporating them into a user-facing moderation system requires thoughtful design to preserve user trust. Our use of named personas, clear correctional logic, and safe phrasing templates are all designed to make the moderation feel human, consistent, and aligned with community norms.

User trust is also influenced by accuracy. Figures 7–10 demonstrate strong model performance, but live deployment would require ongoing testing to ensure cultural, linguistic, and temporal drift do not reduce model relevance. Additional user research is needed to validate response appropriateness across demographics and geographies.

In sum, this work provides an end-to-end system for bias mitigation that prioritizes user dignity and transparency, while maintaining computational rigor. The techniques we introduce serve as a starting point for embedding fairness into the daily interactions that define online social life.

6 Conclusion and Future Work

This research presents a fully automated system that integrates supervised learning and LLM-based generative mitigation to detect and correct both implicit and explicit bias in social media environments. Unlike traditional moderation tools, our approach balances computational accuracy with user-centered interaction design to provide a scalable, transparent, and respectful intervention strategy.

Key contributions of this work include a novel dual-pipeline approach for bias classification and mitigation, a sentiment-informed rule-based logic for differentiating implicit and explicit forms of bias, and a tone-adaptive moderation response engine built on ChatGPT. Our results demonstrate high performance on bias detection tasks, effective mitigation without semantic loss, and potential for deployment in real-world platforms.

The system is fully implemented in Python and designed for deployment as a cloud-based tool for content moderation and bias detection. Beyond research, this technology will serve as the core engine for a next-generation social network platform. This platform is designed to offer a stress-free environment for users through embedded bias mitigation, real-time AI feedback, and escalation pathways via human-in-the-loop moderation assistance. By integrating fairness-oriented ML tools into the platform’s core, we aim to enhance user well-being while ensuring transparency and responsiveness. This work affirms the viability of using LLMs in content moderation beyond toxicity detection—specifically for nuanced bias remediation that respects user dignity and context. Through modular architecture and transparent logging, the system is positioned for integration into live platforms while enabling auditability and continual improvement. Future work will focus on multiple directions:

- **User Studies:** Conduct qualitative research to evaluate how users perceive and respond to automated mitigation messages, particularly in edge cases or culturally sensitive contexts.
- **Real-World Deployment:** Pilot deployment with content platforms to measure engagement impact, trustworthiness, and behavioral feedback loops.

- Adversarial Robustness: Evaluate system resilience to bias evasion strategies (e.g., coded language, sarcasm, misspellings).
- Feedback Loops: Integrate human-in-the-loop interfaces for feedback collection, iterative learning, and crowd-sourced revision of bias definitions.
- Cross-Cultural Generalization: Expand datasets to include multilingual content and explore the adaptability of tone and mitigation strategies across global digital communities.

Overall, this system reflects a step toward responsible AI moderation that prioritizes fairness, transparency, and inclusivity—principles essential to both ethical machine learning and human-computer interaction.

Acknowledgements

This project satisfied the Milestone II and Capstone requirements for the Masters in Applied Data Science (MADS) at the University of Michigan School of Information (UMSI). The project's poster received a MADS award at the UMSI Exposition 2025 Poster Session. The Capstone project mentor was Laura Stagnaro, PhD, and the Milestone II project mentor was Jinseok Kim, PhD.

References

- [1] Mahzarin R Banaji and Anthony G Greenwald. *Blindspot: Hidden biases of good people*. Bantam, 2016.
- [2] Reuben Binns. “Fairness in machine learning: Lessons from political philosophy”. In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 149–159.
- [3] Alec Go, Lei Huang, Richa Bhayani, et al. “Twitter sentiment analysis”. In: *Entropy* 17 (2009), p. 252.
- [4] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. “Does transparency in moderation really matter? User behavior after content removal explanations on reddit”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–27.
- [5] Kristin N Johnson. “Automating the risk of bias”. In: *Geo. Wash. L. Rev.* 87 (2019), p. 1214.
- [6] Nicol Turner Lee, Paul Resnick, and Genie Barton. “Algorithmic bias detection and mitigation: Best practices and policies to reduce consumer harms”. In: *Brookings Institute: Washington, DC, USA* 2 (2019).
- [7] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).
- [8] X. Ma, H. Fang, and J. Zhang. “Fair Influence Maximization: Extending the Reach of Fairness in Social Networks”. In: *IEEE Transactions on Knowledge and Data Engineering* (2023). Accessed: October 15, 2024. url: <https://arxiv.org/pdf/2311.14288>.
- [9] Ninareh Mehrabi et al. “A survey on bias and fairness in machine learning”. In: *ACM computing surveys (CSUR)* 54.6 (2021), pp. 1–35.
- [10] D. MPilat and S. Krastev. *1 billion Reddit comments from 2005–2019 [Data set]*. 2019. url: <https://github.com/Watchful1/PushshiftDumps>.
- [11] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108* (2019).
- [12] G. Sujitha. “Bias Mitigation in Generative AI”. In: *Analytics Vidhya* (2023). Accessed: December 1, 2024. url: <https://www.analyticsvidhya.com/blog/2023/09/bias-mitigation-in-generative-ai/>.
- [13] Rich Zemel et al. “Learning fair representations”. In: *International conference on machine learning*. PMLR. 2013, pp. 325–333.
- [14] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. “Mitigating unwanted biases with adversarial learning”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 335–340.

Appendix 1

Algorithm 1 Bias Detection and Response Pipeline – Pseudocode*

- 1: **Load** NLP and data processing libraries.
 - 2: **Import datasets** (e.g., social media comments).
 - 3: **Define dictionaries of bias indicators** and safe phrases:
 - Categories: racial, antisemitic, sexist, classist, ageism, political, brand, bullying.
 - List representative terms; define context exceptions.
 - 4: **Preprocess text**: lowercase, remove unwanted characters, tokenize, remove stopwords, lemmatize.
 - 5: **Detect bias terms**: match against dictionaries, exclude safe contexts, assign `has_bias`.
 - 6: **Stratify data**: separate biased/non-biased, balance for training/validation/testing.
 - 7: **Assign bias type** from dictionaries; **count occurrences** by category/platform.
 - 8: **Sentiment analysis**: append negative/neutral/positive labels using pretrained model (e.g., RoBERTa).
 - 9: **Classify bias**: explicit (strong negative), implicit (neutral/ambiguous), none.
 - 10: **Train RoBERTa model**: fine-tune for sentiment and bias detection; evaluate using accuracy, precision, recall, and F1-score.
 - 11: **Train DistilBERT model**: fine-tune for bias detection classification; evaluate with same metrics.
 - 12: **Save trained models**.
 - 13: **Create response templates**: implicit (neutral rephrase), explicit (stronger warnings), vary by category.
 - 14: **Generate responses**: select template by classification, optionally enhance with conversational AI.
 - 15: **Store results**: classifications, bias categories, sentiment, responses, escalation logs.
 - 16: **Visualize results**: training/validation loss; precision, recall, F1 over epochs.
-

*Additional information about this research and source code can be found at [Github](#).