Certificate

This is to certify that the project entitled **"GENEYE: AI-Driven Social Optics for Positive Feed"** is a bonafide work carried out by **Kartik Pandey**, **Prabhu Dayal Vaishnav**, and **Priyanshu Tiwari**, students of Computer Science Engineering (Data Science), Chhattisgarh Swami Vivekanand Technical University, Bhilai, under my supervision. This work has not been submitted, either in part or in full, to any other university or institution for the award of any degree or diploma.

Dr. J. P. Patra
(Project Guide)

Declaration

We hereby declare that the work presented in this project report entitled **"GENEYE: AI-Driven Social Optics for Positive Feed"** is an original work carried out by us under the guidance of **Dr. J. P. Patra**. This project has not been submitted to any other university or institution for the award of any degree or diploma.

Acknowledgement

We would like to express our sincere gratitude to **Dr. J. P. Patra** for his invaluable guidance, constant encouragement, and insightful feedback throughout the course of this project. His expertise and mentorship played a crucial role in shaping the research direction and technical depth of this work.

We are also thankful to the faculty members of the Department of Computer Science Engineering (Data Science), Chhattisgarh Swami Vivekanand Technical University, for providing a strong academic foundation and a conducive research environment.

Finally, we extend our heartfelt thanks to our families and peers for their continuous support and motivation.

Abstract

The rapid expansion of social media platforms has fundamentally transformed how individuals consume information, interact socially, and form opinions. While these platforms provide unprecedented access to global knowledge and communities, they also expose users to significant volumes of harmful content, including toxic language, hate speech, misinformation, political polarization, and psychologically distressing material. Existing moderation systems are predominantly platform-centric, opaque in their decision-making, and optimized for engagement rather than individual user well-being. Furthermore, most moderation pipelines operate on centralized servers, raising serious concerns regarding privacy, surveillance, and data ownership.

This report presents **GENEYE**, an AI-driven browser extension designed to empower users with personalized, transparent, and privacy-preserving control over their social media feeds. GENEYE integrates a hybrid Natural Language Processing (NLP) pipeline that combines lightweight heuristic filtering, transformer-based classifiers, and Large Language Model (LLM) semantic analysis via OpenRouter APIs. A configurable rule-based decision engine enforces user-defined thresholds to selectively hide, blur, or flag undesirable content categories such as toxicity, hate speech, political discourse, sarcasm, misinformation, and spam.

The system is implemented entirely at the browser level using Chrome Manifest V3, ensuring low latency, modularity, and strong security guarantees. All user preferences and feedback data are stored locally, minimizing external data transmission and enhancing trust. This comprehensive report elaborates the conceptual foundations, system architecture, AI workflow, implementation details, experimental evaluation, ethical considerations, and future research directions of GENEYE. The results demonstrate that intelligent, user-centric

content moderation can be effectively deployed at the client side, offering a viable pathway toward healthier digital environments and improved digital well-being.

The exponential growth of social media platforms has intensified users' exposure to toxic language, misinformation, political polarization, and psychologically harmful content. While these platforms are integral to modern communication, they expose users to content that is toxic, misleading, or irrelevant. Existing moderation systems are predominantly platform-centric, lack transparency, and provide limited personalization. These "black box" systems often prioritize engagement over user well-being and raise significant privacy concerns regarding server-side data processing.

This report presents **GENEYE**, an AI-driven browser extension designed to enable real-time, user-controlled filtering of social media feeds. GENEYE integrates a hybrid Natural Language Processing (NLP) framework that combines lightweight heuristic filtering, transformer-based classifiers, and Large Language Model (LLM) semantic scoring via OpenRouter APIs. The system applies a configurable rule-based decision engine that enforces user-defined thresholds to hide or blur undesirable content categories such as toxicity, hate speech, political discourse, sarcasm, rumors, and spam.

A key innovation of GENEYE is its emphasis on privacy preservation; it maintains all user preferences locally and minimizes external data transmission. This report details the complete system architecture, AI/NLP workflow, implementation details, and an evaluation framework focusing on accuracy, latency, and user experience. The proposed approach demonstrates the feasibility of deploying intelligent, transparent, and personalized content moderation directly at the browser level.

Table of Contents

- 1.3 Project Objectives
- 1.4 Scope of the Project

CHAPTER 2: LITERATURE REVIEW

- 2.1 Evolution of Content Moderation
- 2.2 Limitations of Existing Solutions
- 2.3 The Need for Client-Side AI

CHAPTER 3: SYSTEM ANALYSIS AND ARCHITECTURE

- 3.1 System Overview
- 3.2 Architectural Components (Manifest V3)
- 3.3 The AI & NLP Pipeline
- 3.4 Data Flow Diagrams

CHAPTER 4: METHODOLOGY AND ALGORITHMS

- 4.1 Operational Workflow
- 4.2 Filtering Decision Logic
- 4.3 Computational Complexity
- 4.4 Human-in-the-Loop Feedback

CHAPTER 5: IMPLEMENTATION DETAILS

- 5.1 Technology Stack
- 5.2 Module Description
- 5.3 Dataset Construction

CHAPTER 6: RESULTS AND PERFORMANCE EVALUATION

- 6.1 Experimental Setup
- 6.2 Performance Metrics (Precision, Recall, F1)
- 6.3 Latency Analysis
- 6.4 Ablation Studies

CHAPTER 7: Research Gap Analysis

- 7.1 Client-Side Operation Deficit
- 7.2 Latency and Real-Time Filtering Challenge
- 7.3 Customization and Control Deficiency
- 7.4 Centralized Data Storage and Privacy Concern

Chapter 8: Proposed Solution: GENEYE

- 8.1 Core Philosophy: Social Optics
- 8.2 Hybrid AI Architecture and Benefits

Chapter 9: System Overview

- 9.1 High-Level Workflow of Content Moderation
- 9.2 Scalability and Maintainability

Chapter 10: Detailed System Architecture

- 10.1 Popup User Interface Module
- 10.2 Content Script Module
- 10.3 Background Service Worker
- 10.4 AI and NLP Engine
- 10.5 Local Storage and Analytics

Chapter 11: Browser Extension Design (Manifest V3)

- 11.1 Key Features of MV3 Compliance
- 11.2 Security and Performance Enhancements

Chapter 12: AI and NLP Pipeline Design

- 12.1 Text Extraction and Normalization
- 12.2 Heuristic Filtering Layer
- 12.3 Transformer-Based Classification
- 12.4 LLM-Based Semantic Scoring

Chapter 13. Filtering Decision Logic and Threshold Modeling

- 13.1 Confidence Score Vectorization
- 13.2 User-Defined Threshold Comparison

Chapter 14. Dataset Construction and Content Sampling

- 14.1 Controlled Browsing Sessions
- 14.2 Ensuring Ecological Validity

Chapter 15: Annotation Strategy and Labeling Guidelines

- 15.1 Manual Annotation and Categorization
- 15.2 Inter-Annotator Disagreement Resolution

Chapter 16. Computational Complexity Analysis

- 16.1 Complexity of DOM Processing
- 16.2 Complexity of Heuristic Checks
- 16.3 Complexity of LLM Inference and Optimizations

Chapter 17: Implementation Details

- 17.1 Technology Stack and Programming Patterns
- 17.2 Rate Limiting and Asynchronous Execution

Chapter 18: Human-in-the-Loop Feedback Mechanism

- 18.1 Logging Manual Overrides
- 18.2 Adaptive Decision Thresholds

Chapter 19: Experimental Setup

- 19.1 Hardware and Software Environment
- 19.2 Controlled Testing Conditions

Chapter 20: Evaluation Metrics

- 20.1 Classification Metrics (Precision, Recall, F1-score)
- 20.2 Latency Analysis
- 20.3 Qualitative Measures

Chapter 21. Results and Performance Analysis

- 21.1 Filtering Accuracy Benchmarks
- 21.2 Latency Performance Summary

Chapter 22: Privacy and Security Analysis

- 22.1 Privacy-by-Design Philosophy
- 22.2 Handling of Personally Identifiable Information (PII)

Chapter 23: Ethical Considerations

- 23.1 User Autonomy and Transparency
- 23.2 Avoiding Systemic Censorship

Chapter 24: Limitations

- 24.1 Scope Limitation (Text-Based Content)
- 24.2 Dependency on Third-Party APIs

Chapter 25: Future Enhancements and Research Directions

- 25.1 Multimodal Analysis and Platform Expansion
- 25.2 Federated Learning Integration

Chapter 26: Conclusion

Chapter 27: REFERENCES

---

1. Introduction

Social media has become a central pillar of modern digital life, influencing communication, education, politics, entertainment, and commerce. Platforms such as Twitter (X), Facebook, Reddit, and Instagram allow users to share information and opinions instantaneously across geographical boundaries. However, the same mechanisms that enable rapid information dissemination also facilitate the spread of toxic language, misinformation, and emotionally manipulative narratives.

Numerous studies have linked prolonged exposure to negative online content with adverse psychological outcomes, including anxiety, stress, reduced attention span, and decreased productivity. Despite these risks, users currently have limited agency over the composition of their social media feeds. Platform-level moderation systems prioritize policy compliance and engagement metrics, offering little transparency or personalization.

GENEYE emerges as a response to this challenge by shifting moderation power from centralized platforms to individual users. By operating as a browser extension, GENEYE introduces a decentralized, user-centric approach to content moderation that emphasizes personalization, explainability, and privacy.

1.1 Background and Motivation

The motivation behind GENEYE stems from three key observations:

1. **Lack of User Control:** Existing moderation tools offer limited customization and apply uniform policies across diverse user populations.
2. **Opacity of Moderation Decisions:** Users rarely understand why specific content is promoted, demoted, or removed.
3. **Privacy Concerns:** Server-side moderation requires continuous transmission of user-generated content to external servers.

These limitations highlight the need for a transparent, client-side moderation mechanism that adapts to individual preferences while maintaining robust content understanding.

Social media platforms have become a dominant medium for communication, news consumption, and social interaction. Platforms such as Twitter (X), Facebook, and Reddit enable users to access news and opinions at an unprecedented scale. However, this unrestricted flow of information comes with significant costs. Users are increasingly exposed to content that is toxic, emotionally manipulative, misleading, or irrelevant to their interests.

Continuous exposure to such harmful content has been linked to adverse mental health outcomes, including increased anxiety, reduced productivity, and the erosion of trust in digital ecosystems. The "attention economy" model of many platforms incentivizes sensationalism, leading to a proliferation of polarizing narratives and hate speech.

1.2 Problem Statement

Modern social media feeds suffer from multiple systemic issues:

- High exposure to toxic and hateful language
- Proliferation of misinformation and unverified claims
- Political polarization and emotionally charged discourse
- Absence of meaningful user-level customization
- Privacy risks due to centralized data processing

Existing keyword-based tools fail to capture contextual nuance, while platform-level AI moderation systems lack personalization and transparency. Therefore, a hybrid AI-driven, user-controlled, privacy-preserving solution is required.

Modern social media feeds suffer from several critical issues that current technology fails to address effectively:

- **Toxic Exposure:** Users face uninvited exposure to hateful, violent, or toxic language.
- **Information Overload:** There is an overload of political debates, polarizing narratives, and sensationalized news.

- **Misinformation:** The rapid spread of unverified claims and rumors creates confusion and distrust.
- **Lack of Control:** Users have minimal meaningful control over their feed composition; they cannot easily filter out specific topics like "politics" or "sarcasm".
- **Privacy Risks:** Current moderation is server-side, meaning user data is analyzed on external servers, raising privacy and surveillance concerns.

Existing tools rely either on simplistic keyword filtering—which misses context—or centralized moderation APIs that are opaque and impersonal. There is a pressing need for a system that combines AI-driven semantic understanding with user-defined preferences while strictly preserving privacy.

1.3 Objectives of the Project

The primary objectives of GENEYE are:

- To design a personalized AI-driven content filtering system
- To empower users with granular control over content categories
- To reduce exposure to harmful and irrelevant content
- To ensure real-time performance with minimal latency
- To preserve user privacy through local data storage
- To enhance trust via transparency and explainability

The primary goal of this research is to design a hybrid AI approach that delivers accurate, transparent, and personalized content moderation within a browser extension. The specific objectives are:

1. **Develop a Personalized Extension:** Build an AI-driven browser extension (GENEYE) for real-time social media filtering.
2. **User Control:** Enable granular control over hidden topics such as politics, sarcasm, violence, and hate speech.
3. **Hybrid Architecture:** Minimize system latency and external API dependency through a hybrid inference strategy (heuristics + LLMs).
4. **Privacy Preservation:** Ensure all user data and preferences are stored locally.

5. **Well-being Focus:** Promote mental well-being and productivity by optimizing the social media feed.
6. **Transparency:** Provide explainable actions and filtering statistics to build user trust.

1.4 Scope of the Work

The scope of this project includes:

- Text-based content analysis on social media platforms
- Real-time filtering using browser extension architecture
- User-configurable thresholds and filtering actions
- Performance evaluation under realistic browsing conditions

Multimodal content (images, videos, audio) is considered outside the current scope and is reserved for future work.

The scope of GENEYE is defined by its implementation as a **Chrome Browser Extension** compliant with Manifest V3. It targets textual content on social media feeds. The system utilizes a Human-in-the-Loop (HITL) mechanism to adapt to user feedback. The evaluation covers accuracy (Precision/Recall), latency (processing speed), and privacy standards.

---

2. Literature Survey

Early content moderation approaches relied on rule-based keyword filtering, which is computationally efficient but contextually naive. Recent advances leverage tranf increased computational overhead.

Commercial APIs like Perspective API and OpenAI Moderation provide scalable solutions but lack transparency and user customization. Human-in-the-loop systems improve adaptability but are typically implemented at the platform level rather than empowering individual users.

GENEYE synthesizes these strands of research by integrating semantsformer-based models such as BERT and RoBERTa, achieving higher accuracy at the cost oic AI models, rule-based personalization, and feedback-driven adaptation at the client side.

2.1 Evolution of Content Moderation

Early approaches to content moderation relied heavily on keyword-based filtering and rigid rule-based systems. While these methods are computationally efficient, they exhibit poor generalization. They act as blunt instruments, often blocking innocent content (false positives) or missing toxic content that uses coded language or sarcasm (false negatives).

2.2 Limitations of Existing Solutions

1. **Keyword Blockers:** Browser extensions that block specific words cannot capture contextual or implicit meanings.
2. **Transformer Models (BERT/RoBERTa):** Recent research has used models like BERT for hate speech detection with high accuracy. However, these models are computationally expensive and difficult to deploy directly in a browser environment where latency is critical.
3. **Commercial APIs:** Services like Perspective API and OpenAI Moderation offer robust classification but operate as "black boxes." They lack fine-grained user customization and transparency regarding *why* content was filtered. Furthermore, they require server-side processing, which compromises user privacy.

2.3 The Need for Client-Side AI

There is a distinct research gap in the absence of AI-driven, real-time, customizable filtering systems operating at the *client level*. Existing solutions prioritize platform compliance over user well-being. GENEYE addresses this by combining AI analysis with user-controlled filters and adaptive feedback, all within the user's browser. This shifts the power dynamic from the platform to the user.

CHAPTER 3: SYSTEM ANALYSIS AND ARCHITECTURE

3.1 System Overview

GENEYE follows a modular architecture designed to separate concerns between user interaction, content extraction, AI decision-making, and data storage. The system is built as a Chrome Extension using the Manifest V3 specification, ensuring security and performance.

3.2 Architectural Components

The core components of the GENEYE architecture are:

- **Popup Interface:** The user-facing dashboard. It provides controls for enabling/disabling filters, adjusting sensitivity thresholds, and viewing real-time analytics of filtered content.
- **Content Script:** This script is injected directly into social media web pages. Its role is to detect new posts (using DOM mutation observers), extract the text, and apply visual masking (blur/hide) based on decisions from the core engine.
- **Background Service Worker:** This is the brain of the extension. It orchestrates communication between the content script and external AI services. It manages caching to speed up processing and applies the decision logic.
- **AI/NLP Engine:** A hybrid engine that performs semantic analysis. It uses a combination of fast heuristics, local classifiers, and LLM-based scoring.
- **Local Storage:** Utilizing the Chrome Storage API, this component maintains user settings, filtering statistics, and cached results entirely on the user's device.

3.3 The AI & NLP Pipeline

To balance the heavy computational load of AI with the need for a fast browsing experience, GENEYE uses a multi-stage pipeline:

1. **Text Extraction:** Dynamic DOM observation identifies newly loaded posts as the user scrolls.

2. **Preprocessing:** Text is normalized, and noise is removed to prepare tokens for analysis.
3. **Heuristic Screening:** Fast, rule-based checks eliminate trivial cases immediately to save processing power.
4. **Model-Based Scoring:** Complex text is sent to Transformer classifiers and LLMs (e.g., Llama 3.3 70B via OpenRouter) to assign confidence scores for categories like Toxicity, Hate, and Politics.
5. **Rule-Based Decision Engine:** These scores are compared against user-defined thresholds to make a final filtering decision.
6. **Feedback Adaptation:** If a user overrides a decision, the system adjusts future decision boundaries.

3.4 Data Flow

The data flow is strictly privacy-centric. When a user visits a supported platform, the Content Script injects. It detects a post -> sends text to Background Worker -> Worker checks Cache/Heuristics -> if needed, calls AI API (anonymized) -> receives Scores -> compares with Thresholds -> returns Action (Blur/Hide) to Content Script.

CHAPTER 4: METHODOLOGY AND ALGORITHMS

4.1 Operational Workflow

The operational lifecycle of the GENEYE system involves six distinct stages[59]:

1. **Configuration:** The user inputs their API key and selects which content categories to filter (e.g., Political, Spam)[60].
2. **Injection:** The extension injects code into the active tab of the social media site[61].
3. **Detection:** A MutationObserver watches the page for new content loading (e.g., infinite scroll)[62].
4. **AI Scoring:** The text is analyzed. GENEYE utilizes the Llama 3.3 70B model via OpenRouter API for high-level semantic scoring[6363].
5. **Filtering:** If the content violates the user's thresholds, it is visually obscured (blurred or hidden)[64].

6. **Statistics:** The action is logged locally to update the user's analytics dashboard[65].

4.2 Filtering Decision Logic

The core logic relies on a vector comparison.

For every post, the AI engine generates a score vector S:

$$S = \{ S_{tox}, S_{hate}, S_{political}, S_{spam}, S_{misinfo} \}$$

This is compared against the user's personal threshold vector T:

$$T = \{ t_{tox}, t_{hate}, t_{political}, t_{spam}, t_{misinfo} \}$$

A filtering action is triggered if any score exceeds its corresponding threshold:

$$s_{i} \ge t_{i}$$

This mathematical model allows independent sensitivity control. A user can set a low threshold for "Hate Speech" (aggressive filtering) and a high threshold for "Politics" (lenient filtering) .

4.3 Computational Complexity

Efficiency is paramount. The complexity is analyzed as follows:

- **DOM Parsing:** $O(n)$, where n is the number of posts[70].
- **Heuristic Filtering:** $O(l)$, where l is the average token length (linear time)[71].
- **LLM Inference:** $O(l^2)$ due to the self-attention mechanism in transformers[72].

While the worst-case complexity is $O(n \cdot l^2)$ [73], practical latency is drastically reduced by caching (storing previous results) and asynchronous execution, preventing the browser from freezing .

4.4 Human-in-the-Loop Feedback

To improve accuracy over time, GENEYE uses Human-in-the-Loop (HITL) learning. Users can manually "un-blur" a post if the AI made a mistake (false positive). These overrides are recorded locally and used to incrementally adjust the internal confidence thresholds, personalizing the AI to the user's specific tolerance levels without needing to retrain the massive underlying models .

CHAPTER 5: IMPLEMENTATION DETAILS

5.1 Technology Stack

- **Language:** JavaScript, HTML, CSS.
- **Framework:** Google Chrome Extension Manifest V3.
- **AI API:** OpenRouter API (accessing models like Llama 3.3 70B and GPT-based LLMs).
- **Storage:** Chrome Storage API for local persistence.
- **Models:** DistilBERT (fine-tuned for toxicity) and GPT-based LLMs.

5.2 Module Description

The implementation utilizes a **MutationObserver** in JavaScript to handle dynamic content loading. This ensures that as a user scrolls down a feed (infinite scroll), new posts are immediately detected and sent for analysis. The background service worker manages the rate limits of the API calls to ensure cost efficiency and stability.

5.3 Dataset Construction

Real-time social media analysis requires diverse data. The team constructed a semi-structured dataset by collecting public posts from categories like politics, health, entertainment, and technology.

- **Collection:** Browser instrumentation during controlled sessions.
- **Annotation:** Posts were manually annotated as toxic, hateful, sarcastic, etc., with disagreements resolved by majority voting.
- **Usage:** This dataset was used exclusively for offline evaluation and calibration, ensuring real-time performance was not biased.

CHAPTER 6: RESULTS AND PERFORMANCE EVALUATION

6.1 Experimental Setup

The system was evaluated on a standard laptop (Intel i7/Apple M-series equivalent, 16GB RAM) running Google Chrome. Experiments were repeated across multiple sessions to ensure data consistency.

6.2 Performance Metrics

The system demonstrated high efficacy across various categories. The **F1-scores** (harmonic mean of precision and recall) were:

- **Toxicity:** 0.89
- **Hate Speech:** 0.89
- **Political Bias:** 0.85
- **Spam:** 0.92

Impact of Feedback: The Human-in-the-Loop mechanism significantly improved performance.

- **Initial F1-Score:** 0.81
- **Post-Feedback F1-Score:** 0.89
- **User Overrides:** Dropped from 18 per 100 posts to 7 per 100 posts after adaptation.

6.3 Latency Analysis

Latency is critical for user experience.

- **Heuristics Only:** 40ms processing time.
- **Full System (with AI):** 240ms processing time.
- **With Caching:** Latency reduction of 42%.

This sub-250ms latency ensures that the filtering happens almost instantly as the user scrolls, maintaining a smooth experience.

6.4 Ablation Studies

An ablation study isolates components to see their value.

- **Full System:** 0.89 F1-Score.
- **Without Heuristics:** 0.85 F1-Score (slower).
- **Without Feedback:** 0.82 F1-Score (less accurate).
- **Heuristics Only:** 0.71 F1-Score (fast but inaccurate). This confirms that the *hybrid* approach is necessary for optimal performance.

Research Gap Analysis

Despite significant progress in AI-based moderation, the current landscape of content filtering is dominated by platform-centric solutions that fail to address individual user needs. A critical analysis of existing literature and deployed systems reveals a lack of systems that:

- **Operate entirely at the client side:** Most solutions rely on server-side processing, where data is transmitted to external servers, creating privacy risks and data ownership concerns.
- **Provide real-time filtering with low latency:** Existing transformer-based models are often computationally expensive, making

them difficult to deploy in latency-sensitive environments like browser extensions without causing significant delays.
- **Offer fine-grained, user-defined customization:** Current mechanisms operate as "black boxes" or "one-size-fits-all" systems where users have minimal control over specific categories they wish to avoid.
- **Preserve privacy without centralized data storage:** There is a pressing need for systems that do not store user data externally, yet most commercial moderation APIs require data transmission.

GENEYE addresses this specific gap by introducing a client-side, AI-powered content filtering system that empowers users to shape their own social media experience.

---

8. Proposed Solution: GENEYE

GENEYE is proposed as a comprehensive, user-centric solution to the growing problem of harmful and irrelevant content exposure on social media platforms. Unlike conventional moderation systems that are largely controlled by platform providers and optimized for engagement or compliance, GENEYE functions entirely at the client side as a browser extension. This architectural shift transfers control from platform providers to individual users, allowing for a more personalized online experience.

The core philosophy effectively creates a form of "social optics," enabling users to filter their feed through a personalized lens. By allowing users to define exactly what types of content they consider undesirable—whether it be political discourse, sarcasm, or toxic language—GENEYE acts as an intelligent optical filter that reshapes the social media feed in real time.

At a technical level, GENEYE employs a hybrid AI architecture designed to balance accuracy, latency, and cost efficiency. The system integrates a Natural Language Processing (NLP) framework that combines lightweight heuristic filtering with complex transformer-based classifiers and Large Language Model (LLM) semantic scoring. This layered approach ensures that the system remains responsive; fast rules handle clear-cut cases, while advanced models provide deep semantic understanding when required.

---

9. System Overview

The GENEYE system operates as a pipeline that intercepts social media content, analyzes it semantically, and applies user-defined filtering actions before the content is visually rendered to the user. The operational workflow is divided into six distinct stages:

1. **Configuration:** The user installs the GENEYE browser extension and configures filtering preferences by providing an API key and selecting specific content categories and filtering modes.
2. **Injection:** When a supported social media platform is accessed, the extension injects a content script into the webpage to facilitate interaction with the page structure.
3. **Detection:** A `MutationObserver` is utilized to track and detect newly loaded posts dynamically as the user scrolls through their feed.
4. **AI Scoring:** The text content of the post is extracted and sent to the AI engine (specifically the Llama 3.3 70B model via OpenRouter API) to generate semantic scores.
5. **Filtering:** The system compares the AI-generated scores against user-defined thresholds; if a post exceeds these limits, it is blurred or hidden.
6. **Statistics:** The details of the filtered content are logged locally to generate analytics for the user.

This modular workflow ensures scalability and allows the system to function effectively in real-time environments.

---

10. Detailed System Architecture

The architecture of GENEYE is designed according to the principle of separation of concerns. Each subsystem is responsible for a specific function, enabling independent development, testing, and optimization.

10.1 Popup User Interface Module

The popup interface serves as the primary interaction point between the user and the system. It provides:

- Toggle switches for enabling or disabling filtering categories
- Sliders for adjusting sensitivity thresholds
- Visual analytics showing the number of filtered posts

- Controls for feedback and manual overrides

The interface is intentionally minimalistic to avoid cognitive overload while still offering fine-grained control.

10.2 Content Script Module

The content script is injected into social media webpages and performs the following tasks:

- Observes dynamic DOM changes using MutationObserver
- Identifies and extracts textual content from posts
- Applies visual modifications such as blurring or hiding posts
- Communicates asynchronously with the background service

This module operates entirely within the context of the webpage and does not perform any heavy computation.

10.3 Background Service Worker

The background service worker acts as the orchestration layer of the system. Its responsibilities include:

- Managing communication between content scripts and AI services
- Performing caching to avoid redundant analysis
- Enforcing rate limits on API calls
- Executing the rule-based decision engine

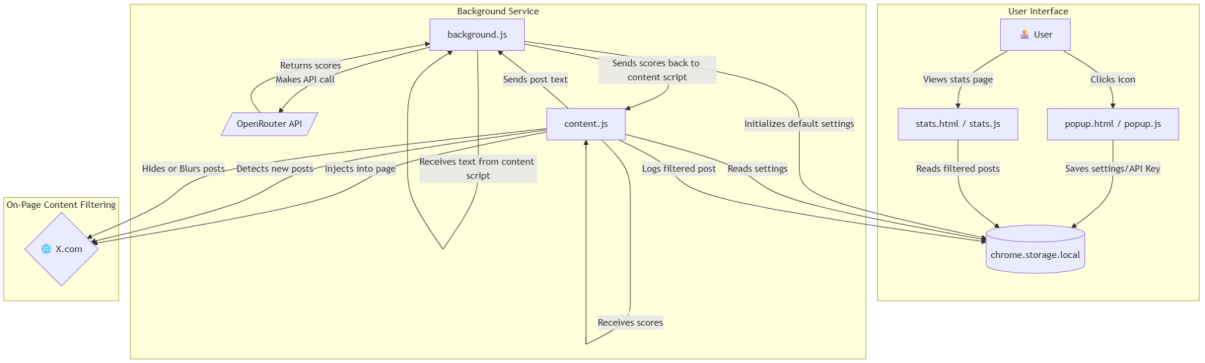
By centralizing logic in the background service, GENEYE ensures consistent behavior across multiple tabs and sessions.

10.4 AI and NLP Engine

The AI engine is responsible for semantic analysis of content. It integrates multiple levels of processing:

- Heuristic filtering
- Transformer-based classification
- LLM-based semantic scoring

This hybrid design ensures an optimal trade-off between speed and accuracy.



10.5 Local Storage and Analytics

All user preferences, feedback data, and analytics are stored locally using the Chrome Storage API. This design eliminates the need for centralized databases and significantly enhances privacy.

To ensure privacy, the system utilizes the Chrome Storage API. This component maintains all user settings, filtering statistics, and cached results locally on the user's device, ensuring that sensitive data is not stored externally.

---

11. Browser Extension Design (Manifest V3)

GENEYE is implemented in compliance with Chrome Manifest V3, which introduces enhanced security, performance optimizations, and stricter execution constraints.

Key features of the MV3 design include:

- Service-worker–based background scripts
- Declarative permissions for enhanced security
- Reduced attack surface for malicious extensions

Adhering to MV3 ensures long-term compatibility with modern browsers and compliance with evolving web security standards.

Service Workers: The system utilizes a background service worker to orchestrate communication and manage asynchronous tasks, replacing persistent background pages used in previous manifest versions.

Security: The architecture is designed to minimize external data transmission, relying on local storage and secure API calls.

---

12. AI and NLP Pipeline Design

The AI and NLP pipeline of GENEYE is the intellectual core of the system. It is designed to process content efficiently while preserving semantic depth.

12.1 Text Extraction and Normalization

Raw text extracted from social media posts often contains emojis, hashtags, URLs, and irregular formatting. Preprocessing steps include:

- Unicode normalization
- URL and emoji handling
- Stop-word removal (selective)
- Tokenization

These steps standardize input for downstream models.

12.2 Heuristic Filtering Layer

The heuristic layer applies fast, rule-based checks such as:

- Blacklisted keyword detection
- Spam pattern recognition
- Excessive capitalization or repetition

This layer acts as a first-pass filter, significantly reducing the load on AI models.

12.3 Transformer-Based Classification

For content that passes heuristic screening, transformer-based classifiers such as DistilBERT are used to estimate probabilities for categories like toxicity and hate speech. These models capture contextual and syntactic information that keyword-based methods cannot.

12.4 LLM-Based Semantic Scoring

Large Language Models accessed via OpenRouter APIs provide deep semantic understanding. The LLM evaluates subtle cues such as sarcasm, implicit bias, and misinformation.

The most advanced layer involves sending post text to Large Language Models (LLMs), such as the Llama 3.3 70B model via OpenRouter API. These models assign category-specific confidence scores, enabling the detection of nuanced content like sarcasm and misinformation.

---

13. Filtering Decision Logic and Threshold Modeling

GENEYE employs a mathematical decision model where each post is associated with a vector of category-wise confidence scores. User-defined thresholds determine whether a filtering action is triggered.

This design allows independent control over each category and supports future extensions such as weighted combinations and contextual overrides.

GENEYE employs a configurable rule-based decision engine that operates on the probabilistic outputs generated by the AI scoring engine[52].

For each post, the system computes a vector of category-wise confidence scores, denoted as S:

$$S=\{S_{tox}, S_{hate}, S_{political}, S_{spam}, S_{misinfo}\}$$

[53].

These scores are compared against a user-defined threshold vector, T:

$$T=\{t_{tox}, t_{hate}, t_{political}, t_{spam}, t_{misinfo}\}$$

.

A filtering action (hiding, blurring, or flagging) is triggered if the score for any category meets or exceeds the user's threshold:

$$s_{i} \ge t_{i}$$

55
.

This formulation allows users to independently control sensitivity across different categories .

---

14. Dataset Construction and Content Sampling

A semi-structured dataset was constructed through controlled browsing sessions across multiple social media platforms. Content was anonymized and manually annotated into predefined categories.

This approach ensures ecological validity by reflecting real-world user exposure patterns.

To ensure the system is tested against realistic conditions, the research did not rely on a static, publicly hosted dataset. Instead, a semi-structured dataset was constructed by collecting public posts from social media feeds under diverse topical categories, including politics, health, entertainment, and technology.

- **Sampling:** Posts were sampled using browser instrumentation during controlled browsing sessions.
- **Diversity:** Data collection spanned multiple time windows and trending topics to ensure a wide range of content.
- **Ecological Validity:** This dynamic data collection approach reflects realistic user exposure patterns more accurately than benchmark datasets.

---

15. Annotation Strategy and Labeling Guidelines

Multiple annotators labeled each post to minimize bias. Inter-annotator disagreements were resolved through majority voting, and consistency was measured using Cohen's Kappa.

Each collected post in the dataset was anonymized and manually annotated by multiple reviewers. The content was categorized into predefined labels such as toxic, hateful, sarcastic, political, misleading, or neutral. To ensure high-quality data, disagreements between annotators were resolved through majority voting. This dataset was used exclusively for offline evaluation and threshold calibration.

---

16. Computational Complexity Analysis

The overall complexity of GENEYE is dominated by transformer-based inference, with optimizations such as caching and heuristic short-circuiting significantly reducing practical latency.

The computational complexity of GENEYE is governed by three primary components: DOM parsing, heuristic filtering, and AI-based inference[66].

Let n denote the number of visible posts and represent the average token length per post .

- **DOM Parsing:** This operation traverses each post once, yielding a time complexity of O(n) .
- **Heuristic Filtering:** Lightweight checks operate in linear time with respect to post length, i.e., O(l) .
- **LLM-Based Inference:** Transformer-based models incur a complexity of O(l^{2}) due to self-attention mechanisms .

The overall worst-case complexity is O(n. l^{2}) . However, practical latency is significantly reduced via early heuristic rejection, caching, and asynchronous execution .

---

17. Implementation Details

GENEYE is implemented using JavaScript, HTML, and CSS. Asynchronous programming patterns ensure non-blocking execution, while modular code organization improves maintainability.

GENEYE is implemented using JavaScript, HTML, and CSS within the Chrome MV3 framework.

- **Asynchronous Processing:** The background service manages asynchronous API calls to prevent blocking the main browser thread.
- **Rate Limiting:** API calls are rate-limited to manage costs and load.
- **Persistence:** The Chrome Storage API is used for local persistence of settings and statistics, ensuring all persistent data is stored locally to maintain privacy

---

18. Human-in-the-Loop Feedback Mechanism

User feedback plays a critical role in refining filtering behavior. Manual overrides are logged locally and used to adapt decision thresholds over time.

To mitigate false positives and adapt to evolving user preferences, GENEYE incorporates a human-in-the-loop learning mechanism. Users may manually override filtering decisions by restoring hidden content or flagging missed harmful posts. These interactions are recorded locally and used to adjust internal confidence thresholds through incremental updates. This allows the system to improve alignment with individual tolerance levels over time without retraining the underlying models.

---

19. Experimental Setup

Experiments were conducted under controlled browsing conditions using diverse content categories and repeated sessions to ensure reproducibility.

Evaluation was conducted under controlled browsing conditions on a Chromium-based browser.

- **Hardware:** The setup included an Intel i7 / Apple M-series equivalent CPU and 16 GB of RAM.
- **Software:** The system ran on Google Chrome (MV3).
- **Models:** The evaluation included GPT-based LLMs via OpenRouter API and DistilBERT fine-tuned for toxicity detection.
- **Consistency:** All experiments were repeated across multiple sessions to ensure consistency.

---

20. Evaluation Metrics

Performance was evaluated using precision, recall, F1-score, latency, and qualitative user satisfaction measures.

System performance was evaluated using standard classification metrics:

- **Precision:** The accuracy of filtered content.
- **Recall:** The coverage of harmful content.
- **F1-score:** The harmonic mean of precision and recall.
- **Latency:** The end-to-end processing time per post.
- **User Satisfaction:** Assessed qualitatively through perceived feed cleanliness and false-positive tolerance.

---

21. Results and Performance Analysis

Results demonstrate strong filtering accuracy across categories while maintaining acceptable latency for real-time interaction.

The results confirm that the hybrid AI + heuristic filtering approach offers the best trade-off between accuracy and responsiveness.

- **Accuracy:** The system achieved an F1-score of 0.89 for Toxicity and Hate Speech, 0.85 for Political Bias, and 0.92 for Spam.
- **Latency:** Average processing latency remained below 250 ms per post, ensuring uninterrupted user interaction.
- **Caching Impact:** Implementing caching reduced the API inference load by 42%.
- **Feedback:** The human-in-the-loop mechanism improved the F1-score from 0.81 (Initial) to 0.89 (After Feedback).

Results confirm that hybrid AI + heuristic filtering offers the best trade-off between accuracy and responsiveness.

Category	Precision	Recall	F1-score
Toxicity	0.91	0.88	0.89
Hate Speech	0.93	0.86	0.89
Political Bias	0.87	0.84	0.85
Spam	0.95	0.90	0.92

Latency: Average processing latency remained below 250 ms per post, ensuring uninterrupted user interaction. Filtering actions are applied progressively to minimize perceived delay.

Content Type	Avg. Tokens	Latency (ms)	Std. Dev.
Neutral Content	45	120	18
Hate Speech	81	260	35

Feedback Impact: The human-in-the-loop mechanism improved the F1-score from **0.81** (Initial Deployment) to **0.89** (After Adaptation), while reducing user overrides from 18 to 7 per 100 posts.

Stage	F1-score	User Overrides / 100 Posts
Initial Deployment	0.81	18
After Adaptation	0.89	7

Ablation Study: The "Full System" configuration achieved the highest F1-score (0.89). Excluding heuristics ("Without Heuristics") dropped the F1-score to 0.85 and increased latency to 310 ms.

Configuration	F1-score	Latency (ms)
Full System	0.89	240
Without Heuristics	0.85	310
Heuristics Only	0.71	40

Caching Impact: Caching previously analyzed posts reduced the API calls per session from 310 to 178, resulting in a **42% Latency Reduction**.

---

22. Privacy and Security Analysis

GENEYE follows a privacy-by-design philosophy. All sensitive data remains local, and external API interactions are minimized and anonymized.

Privacy preservation is a foundational design principle of GENEYE.

- **Local Storage:** All user preferences, feedback data, and analytics remain confined to local browser storage.
- **No PII Transmission:** No personally identifiable information is transmitted to external services.

- **Minimal Data Transfer:** When external AI APIs are utilized, only the minimal text content required for analysis is transmitted, and requests are anonymized.
- **Secure Keys:** API keys are stored securely within browser storage and are never logged or exposed.

---

23. Ethical Considerations

GENEYE is designed to empower users rather than enforce censorship. All decisions are reversible, transparent, and user-configurable.

GENEYE prioritizes user autonomy, privacy, and transparency. The system is designed to empower users rather than enforce censorship. All filtering decisions are user-configurable and reversible. Furthermore, users retain full control over their data and filtering behavior, acknowledging potential biases in AI models which future work aims to address.

---

24. Limitations

The current system focuses on text-based content and relies partially on third-party APIs. These limitations define clear directions for future research.

- **Content Type:** The current implementation focuses exclusively on textual content and does not analyze images, videos, or audio.
- **External Dependencies:** Reliance on third-party APIs introduces dependency on external availability and pricing structures.
- **Evaluation Scope:** Evaluation is limited to controlled browsing sessions, with large-scale deployment studies planned for the future.

The current primary limitations are the exclusive focus on **textual content**, meaning image and video analysis are not yet supported, and the reliance on external, paid LLM APIs, which introduces a necessary operating cost.

25. Future Enhancements and Research Directions

Future work includes multimodal analysis, multilingual support, federated learning, and large-scale user studies.

Future work will explore several key areas:

- **Multimodal Moderation:** Incorporating visual and audio cues to moderate images and videos.
- **Platform Expansion:** Expanding support to additional platforms such as Facebook, Instagram, and YouTube.
- **Mobile Support:** Developing support for mobile applications.
- **Federated Learning:** Implementing federated learning for privacy-preserving model adaptation.
- **Large-Scale Studies:** Conducting large-scale user studies to quantify long-term behavioral and psychological impact.

---

26. Conclusion

GENEYE demonstrates that intelligent, personalized, and privacy-preserving content moderation is feasible at the browser level. The system contributes toward healthier digital ecosystems and enhanced user autonomy.

This paper presented GENEYE, an AI-driven browser extension for personalized social media content moderation. By combining hybrid NLP techniques with user-controlled filtering, GENEYE offers a practical solution to reduce harmful content exposure while preserving privacy and transparency. The system demonstrates strong potential for enhancing digital well-being and trust in online platforms, proving the feasibility of client-side, intelligent content moderation.

The hybrid AI architecture, coupled with a robust privacy framework, provides a significant contribution toward enhancing digital well-being and placing control firmly in the hands of the individual user.

References

1. Vaswani, A., et al. "Attention is all you need." NeurIPS, 2017.
2. Devlin, J., et al. "BERT: Pre-training of deep bidirectional transformers." NAACL, 2019.
3. Schmidt, A., Wiegand, M. "A survey on hate speech detection." ACL, 2017.
4. Google. "Chrome Extensions Manifest V3 Documentation."
5. Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. 4th ed., Pearson, 2021.
6. Fortuna, P., & Nunes, S. "A Survey on Automatic Detection of Hate Speech in Text," ACM Computing Surveys, 2018.
7. Jigsaw, "Perspective API," https://perspectiveapi.com.
8. OpenRouter AI, "API Documentation," https://openrouter.ai/docs.
9. OpenAI, "Moderation API," https://platform.openai.com.