

Working Name:

Crescent Social Guardian

Purpose:

The purpose of this software/ System is to analyze digital chat conversations for indicators of online grooming, sexual predation, coercion, manipulation, boundary testing, or other potentially harmful behavior involving minors.

It in itself will not declare a predator alone, but it will go through the following processes:

1. Receive chat messages from the extension to the adapter
2. Clean and package data for transport to the model.
3. Use machine learning to interpret messages and intent, identifying patterns.
4. When a warning needs to be issued, a message is pushed to the web application.
5. Web application receives JSON file from the backend and upon receiving, a case is created, and a notification is sent to the required user to review the assigned case.
6. The notification will include recommendations of actions to take based on the case summary.

Problem statement:

Gaming platforms, social applications, and other online services may process very large volumes of user-to-user messages. Manual review of all conversations is impractical, while keyword-based filters may miss grooming behaviors that develop gradually across multiple messages.

Organizations need a system that can identify potentially dangerous conversational patterns, prioritize the highest-risk conversations, and provide reviewers with sufficient evidence to investigate the situation.

The proposed system will help organizations and parents reduce the amount of manual review required to monitor online chat logs by automatically analyzing conversations, identifying potential safety concerns, and generating actionable risk reports.

System Vision:

The system will contain 4 major Systems : Chat platform → Extension → Adapter → Transformer-based classification model → Web Application → Reviewer

Extension:

The extension will (upon download and installation), use JS code to obtain chat logs and see chat mutations. When a mutation is noticed, a trigger occurs sending the whole window of the viewed chat to the back end where the adapter is located.

Adapter:

The adapter receives or collects chat data from the extension and validates it. The adapter can convert these using API endpoints with FastAPI. The chat data will be cleaned and formatted into the desired format using Pydantic. The adapter packages the data into standard format and sends it to the model through the model interface.

Transformer-based classification model (ML model):

The transformer-based classification model analyzes chat content and produces risk-classification scores. The transformer produces contextual feature representations from the input, and the classifier head converts those representations into class scores or predicted labels. A backend service then applies configurable risk thresholds, associates flagged results with relevant chat metadata and excerpts, and returns structured data for display in the website dashboard

Explainability Layer:

This will receive chat logs that contribute to the probability threshold. Captum takes in the messages and creates raw math gradient tensors, finding specific high-risk words. Captum sends Pytorch tensors to Transformers-Interpret. Transformers-Interpret will take these tensors and create a list of Tuples, which will pass through a middleware component to be formatted and sent to BART to summarize the formatted tuples and provide a User-focused summary.

Web application:

This will receive a package from the summarization model, with all required information to handle the alert. It will unpack the alert message, locate the intended reviewer who would be assigned to the account, inform them of an assigned review, and display the information hidden until login. Web Application will be multi-tenant, role-based authenticating/authorizing so only those who should see the alert will see the alert.

Initial Scope:

1. The extension packages the chat and sends it to the adapter.
2. The adapter validates and formats the conversation.

3. The adapter sends the sanitized data to the model through the model interface.
4. The data is preprocessed, then passed into the transformer model to get the initial tensor of the data.
5. The softmax-classifier head takes in the tensor and returns a prediction array that will be used to determine if a conversation is predatory or not.
6. When a conversation is given a weight from the classification model, it is then sent to the database for storage.
7. Upon a threshold being hit, a case is built. When the case is created, a notification is sent to the authorized reviewer.
8. The reviewer can sign in to the web app to review the alert.
9. The reviewer can address or close the alert.

Secondary Scope/future Scope:

1. The web application and phone app show real-time for analysis and demonstrations.
2. Direct import and interaction with game platforms and chatting software.

Stakeholders:

Platform organization

Safety reviewer

System admin

Anti-Predator software owners

Security personnel

Platform users

Parents/Guardians

Legal team

Primary user roles:

1. Reviewer
2. Admin
3. integration/system admin

Functional requirements:**Extension:**

FR-E1 – Browser Compatibility

The extension shall operate within supported Chromium-based web browsers and shall execute on supported web-based chat platforms.

FR-E2 – Dynamic Chat Monitoring

The extension shall detect newly rendered chat messages dynamically without requiring the user to manually refresh the webpage.

The extension may use the JavaScript MutationObserver interface to monitor changes to the rendered chat content.

FR-E3 – Client-Side DOM Inspection

The extension shall inspect the Document Object Model (DOM) of the active webpage to identify and extract rendered chat-message content.

The extension shall obtain message text from applicable HTML elements using rendered text properties such as innerText or textContent.

FR-E4 – Message Tracking and Duplicate Prevention

The extension shall assign or derive temporary identifiers for extracted message nodes and maintain a record of previously processed messages to prevent duplicate extraction during the active session.

Adapter:

FR-A1- Adapter validation

The adapter shall verify the chat log is in the format denoted by the extension.

FR-A2- FastAPI Normalization

The adapter shall cleanse and normalize the data to a standard format.

Transformer-based classification model:

FR-T1: Preprocessing

The model will accept chat logs from the adapter as input and then preprocess/tokenize text.

FR-T2: Transformer

The transformer will analyze the chat logs by reading entire conversation windows, allowing the model to capture the context of each message using chat text and de-identified speaker metadata, preserving user privacy while still capturing conversational context.

FR-T3: Classifier Head

The contextual representations from the transformer are fed into a XGBoost classifier head that learns to map examples to the correct class using a classification loss function, an optimizer, and a learning-rate scheduler. The classifier head allows the model to assign each conversation to a grooming-risk category. The XGBoost layer converts the model's output into probabilities from 0.0 to 1.0 and labels the window based on the classes we specify.

Explainability Layer:

FR-EL1: Chat Record Receiving

The Explainability Layer (Transformers-Interpret) will take in chat logs that reach a set threshold.

FR-EL2: Captum Conversion

Captum shall be used in Transformers-Interpret to generate floating-point tensors

FR-EL3: Tuple Conversion

Transformers-Interpret will take in floating-point tensors and convert them into a list of tuples.

FR-EL4: Tuple Translator

The tuples will be sent to the Tuple Translator to be properly formatted and sent in to BART.

FR-EL5: BART Summarizer

BART will create a concise summary of the formatted tuples for the user to easily understand.

Web Application:

FR-W1: File Receiving

The web application shall receive files from the backend and database with no loss.

FR-W2: Flag Priority

The web application shall organize/ triage all flags into a list by priority.

FR-W3: Notification alert

The web application shall send notifications via email/ text to the reviewer upon receiving an alert.

FR-W4: KPI

The web app shall utilize KPI in card based structure to inform the user.

FR-W5: User Authentication/Authorization

The User shall log in with a specific name and password, allowing specific privileges on the WebApp.

Non-Functional Requirements:**High-level Requirements:****NF-H1: Efficient Storage**

There needs to be an efficient way for the web application to send conversations to the database to be stored. Also, there should be an efficient way to pull conversations from the database.

NF-H2: Strict Security

All of the clients' personal data that could be mentioned in the chats, such as addresses or credit card information, must be protected and kept safe.

Adapter:**NF-A1: Universal Input**

The Adapter can universally call chats from all social networks

Transformer-based classification model:**NF-T2: Light-weight model architecture:**

The model needs to be lightweight to prevent overhead.

Web application:**NF-WA1: Settings**

The WebApp will have notification preference settings as well as other accessibility features.

Performance:

NF-P1: Concurrent Management

We will have multiple chats being monitored at the same time, so we will need to be able to get multiple results from the model in real time.

NF-P2: Notification TimeStamp/Priority

Emails and texts must be sent within a specific timeframe to address critical triage flags.

NF-P3: Customer Flexibility

The WebApp shall be flexible for the use of companies (security dept.) as well as individuals.

NF-P4: Normalization

The classifier must understand and normalize erroneous sentences or words. This allows for slang, typos, and other types of grammatical deviations.

NF-P5: DOM Scrape/Batching Delay

Delay should be less than 300ms

NF-P6: Model Accuracy

Accuracy should be over 90%

NF-P6: WebApp Notifications

The WebApp shall send notifications within 5 - 10 seconds of receiving a high-risk packet.

NF-P7: System Availability:

The System shall remain online 99.9% of the time, and shall have less than 24 hours of downtime a month

Reliability and Security:

NF-R1: Credentials

Using role-based authentication and authorization for secure use

NF-R2: Encryption

Encrypted login details for user security and roles

NF_R3: Cloud Security

Cloud-protected sensitive data for chat log security

Legal Compliance:

NF-L1: PII Stripping

Chat logs must be cleaned of personally identifiable information after a set amount of time if not flagged.

NF-L2: Minimal Bias

Bias in false-positive rates must not deviate by more than a set amount.

User accessibility:

NF-U1: WAI (Web accessibility Initiative)

The website shall be user-friendly and provide WAI solutions to the impaired.

Interface diagram

