SendWise: A Privacy-Preserving On-Device Machine Learning Framework for Cyberbullying Risk Detection and Parental Awareness

Namrata Gaikwad [1], Sharada Ohatkar [2]

[1]MKSSS's Cummins College of Engineering for Women, Pune, India. E-mail: namrata.m.gaikwad@cumminscollege.in
[2]MKSSS's Cummins College of Engineering for Women, Pune, India. E-mail: sharada.ohatkar@cumminscollege.in

Abstract
Parents are concerned about cyberbullying and harmful online behavior, but extensive monitoring can compromise children's privacy and autonomy. This paper presents SendWise, a privacy-preserving monitoring system designed to improve online safety without exposing personal conversations. SendWise performs message analysis locally on an Android device using rule-based screening and a Random Forest classifier. Only behavioral metadata, including risk category, severity, timestamp, and user action, are transmitted to a web-based parental dashboard; message content is neither stored nor transmitted. User identifiers are protected using SHA-256-based pseudonymization. The system follows Privacy by Design principles and provides users with an intervention warning while allowing them to decide whether to edit or send a message. Evaluation on a manually annotated dataset of 20,122 messages achieved 85.96% precision, 95.73% recall, and a 90.58% F1-score on the held-out test set. Functional testing across three Android devices confirmed successful operation of the prototype. The results demonstrate that behavioral awareness can be provided to parents while reducing exposure of private communications, offering a practical approach for privacy-preserving family safety applications.

Keywords: Cybersecurity; Abusive Language Detection; Natural Language Processing; Machine Learning; Privacy-Preserving Systems; Cyber Harassment; Content Moderation

1. Introduction

Cyberbullying is a growing concern among young people. Studies show that about 37% of adolescents experience cyber- bullying each year [1], [2]. The effects can be serious. Victims may face emotional stress, anxiety, and lower academic performance [1], [2]. Many parents use digital monitoring tools to help protect their children. These tools are designed to identify harmful online behavior. However, most of them also reveal private conversations. When a risk is detected, parents often receive parts of the original messages. This creates a challenge. Children need protection, but they also need privacy. Commercial parental monitoring tools can scan online activity and generate alerts when potentially harmful behavior is detected [3]. Commercial products such as Bark and Qustodio are examples of this approach. While these systems can help identify harmful content, they may also expose private conversations that are unrelated to safety concerns. As children grow older, privacy becomes increasingly important. Research suggests that intrusive monitoring can weaken trust and reduce open communication between parents and adolescents [4], [5]. Some parental control systems avoid message monitoring altogether. They rely on screen-time restrictions or application blocking features [6]. While these controls may reduce expo- sure to certain online risks, they can also limit healthy social interaction and opportunities to develop digital skills [6]. Research in child development highlights different approaches to parental involvement. One approach focuses on communication, guidance, and voluntary disclosure. Another relies more heavily on monitoring and control [7], [8]. Studies suggest that open communication often leads to healthier parent-child relationships and greater willingness to share concerns [7], [8]. Despite these findings, many existing digital safety tools continue to prioritize surveillance. This paper presents an alternative approach. Instead of exposing message content, the proposed system focuses on behavioral patterns. Following Privacy by Design principles [9], SendWise performs cyberbullying detection directly on the user's device. Only summarized behavioral metadata is shared with parents. Message content is processed locally on the device and is not included in transmitted metadata. The goal is not surveillance. The goal is to provide meaningful awareness while preserving trust and privacy.

1.1 System Objectives and Design Requirements:

We developed SendWise to explore a different approach to parental monitoring. The system is based on the idea that parents often need awareness of online risks but not access to private conversations. Instead of sharing message content, SendWise focuses on behavioral patterns and risk indicators. Several design goals guided the development of the system. First, user privacy must be always protected. Message content should remain on the device and should never be available to parents or external services. This helps maintain trust while reducing unnecessary exposure of personal conversations. Second, the system must be able to identify potentially harmful interactions with reasonable accuracy. These interactions may include cyberbullying, harassment, threats, or other forms of risky communication that can affect adolescent well- being. Third, the

information presented to parents should be useful and easy to understand. Parents need enough information to recognize concerning trends and decide whether intervention is necessary. At the same time, the system should avoid generating excessive notifications that may lead to alert fatigue. Another important goal is efficiency. The application should run smoothly on everyday mobile devices. Battery consumption, memory usage, and processor load should remain low so that users do not experience noticeable performance issues. Finally, the system should respect adolescent autonomy. Young users should retain control over their daily communication and decision-making. Rather than acting as a surveil- lance tool, SendWise is designed to encourage reflection and responsible online behavior through educational prompts and warnings. Together, these goals form the foundation of the proposed system and guide both the technical architecture and user experience design.

1.2 Contributions:

This work contributes to the design of privacy-aware parental monitoring systems in several ways. First, it introduces a three-layer architecture that separates content processing from parental reporting. Potentially harmful messages are analyzed on the device itself, while only limited behavioral metadata are shared. This design helps reduce privacy risks by ensuring that message content never leaves the user's device. Second, the paper describes a lightweight detection approach that combines rule-based analysis with machine learning techniques. The method is designed for mobile environments where memory, battery life, and processing power are limited. This allows cyberbullying detection to be performed efficiently without relying on cloud-based content analysis. Third, the study presents a dashboard that focuses on behavioral trends rather than individual messages. Parents receive summaries, risk indicators, and activity patterns that support informed decisions. At the same time, the privacy of personal conversations is preserved. Together, these contributions demonstrate that parental awareness and user privacy do not have to be competing goals. The proposed approach shows how meaningful safety insights can be provided without exposing private message content.

1.3 Comparison with existing work:

1) Parental Monitoring Technologies:

Many commercial parental monitoring tools rely on direct access to message content. These systems monitor online communications and alert parents when potentially harmful behavior is detected [5], [10]. Commercial products such as Bark and Qustodio are examples of this approach [3], [11]. While such systems can help identify online risks, they also require broad access to personal conversations. Research suggests that highly intrusive monitoring may reduce trust and discourage open communication between parents and adolescents [4], [5]. Other approaches focus on influencing behavior rather than monitoring conversations. For example, systems that provide real-time warnings when offensive language is detected encourage users to reconsider their actions before sending a message. Prior work has shown that technology-based interventions can support the detection and prevention of cyber-bullying and harmful online interactions [12], [16]. However, these approaches generally focus on user behavior and do not provide parents with awareness of emerging risks. As a result, many existing solutions fall into one of two categories. They either rely on extensive access to message content or provide no parental awareness at all. Relatively few systems attempt to balance parental awareness with strong privacy protections. This gap motivates the development of privacy-preserving approaches that provide meaningful safety insights without exposing personal communications.

2) Privacy-Preserving Computing:

Privacy protection has become an important consideration in modern digital systems. Privacy by Design proposes that privacy safeguards should be built into the system architecture from the beginning rather than added later through policies or user agreements [9]. Recent research has also explored techniques that reduce the need to collect or centralize sensitive in- formation. Examples include differential privacy, which limits the exposure of individual data records, and federated learning, which allows model training across distributed devices without transferring raw data to a central server [13], [14]. These approaches show that useful insights can often be obtained without direct access to personal information. Despite growing interest in privacy-preserving technologies, their use in parental monitoring applications remains limited. Many existing systems continue to rely on direct access to message content and communication histories.

3) Cyberbullying Detection Systems:

Research on cyberbullying detection has grown significantly over the last decade. Early approaches often relied on keyword matching and manually defined rules. However, later studies found that harmful content is frequently context- dependent and cannot always be identified through keywords alone [15], [16]. More recently, advanced machine learning approaches, including ensemble learning frameworks, have demonstrated improved detection performance across diverse social media platforms while addressing variations in cyberbullying language [17]. Despite these advances, most existing approaches focus primarily on

improving detection accuracy using centralized analysis of message content. In contrast, SendWise performs cyberbullying detection directly on the user's device and shares only behavioral metadata with parents, thereby preserving user privacy while still enabling parental awareness.Factors such as tone, intent, and conversational context often influence whether a message should be considered harmful. Most cyberbullying detection systems perform analysis on centralized servers where complete message content is available. This allows the use of computationally intensive machine learning models. In contrast, SendWise performs analysis directly on the user's device. As a result, the detection process must operate with limited memory, processing power, and battery consumption while still maintaining acceptable accuracy.

The comparison focuses on the main architectural difference between existing approaches and SendWise. Commercial monitoring systems generally provide parents with detailed alerts or content-related information. Restrictive systems focus mainly on limiting access to applications or services. SendWise takes a different approach by keeping message analysis on the device and reporting only behavioral metadata. The purpose of this comparison is therefore not to claim that SendWise provides more information than existing tools, but to show how it changes the privacy boundary between detection and parental reporting.

A comparison of the architectural and privacy characteristics of SendWise with existing approaches is presented in Table 1.

Table 1. Novelty comparison table

Comparison Parameter	**Content-Monitoring Systems (Bark/Qustodio)**	**Conventional Cyberbullying ML**	**SendWise**
On-device inference	Varies	Usually no on-device inference	Yes
Parent sees message content	Yes/Partial	Not applicable	No
Data transmitted	Content/risk information	Raw text to analysis system	Behavioural metadata
Privacy boundary	Low	Low	High
Parent awareness	High	Usually none	Yes
Intervention	Parent alert/control	Detection	User warning + parent trends
Language	Not reported	Not reported	English and Hinglish
Evaluation Dataset	Not reported	Not reported	20,122 manually annotated messages; 5,031 held-out test messages
Empirical Result	Not reported	Not reported	Precision: 85.96%; Recall: 95.73%; F1: 90.58%

The novelty of SendWise is primarily architectural rather than a claim of superior classification accuracy. The system separates local risk detection from parental reporting: the message is required for local classification but is not required for parental visualization. This creates a privacy boundary between the person generating the message and the parent receiving safety-related information. The complete end-to-end processing workflow of SendWise is illustrated in Figure 1.

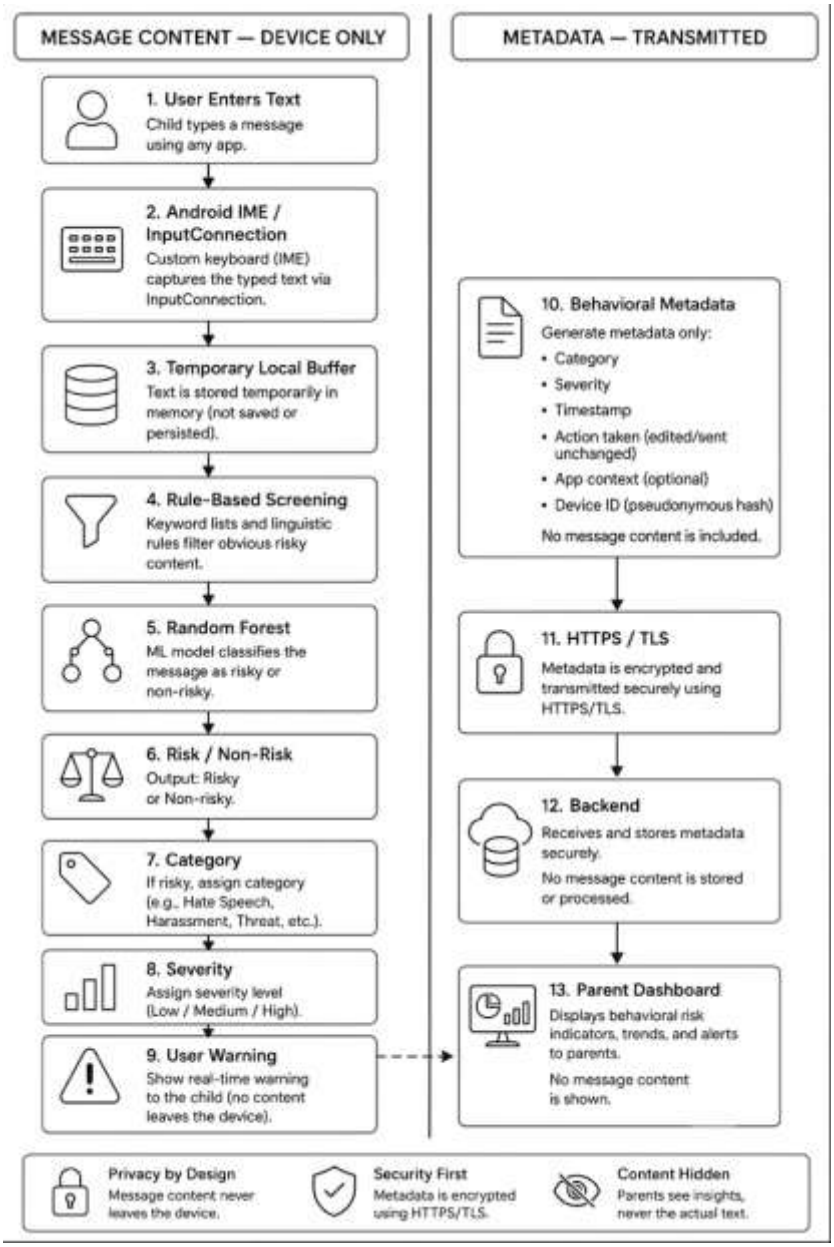

Figure 1. Original End-to-End SendWise Processing Workflow

2. Materials and Methods

SendWise is organized into three layers. Each layer performs a specific task: message analysis, metadata transmission, and parental reporting.

2.1 On-Device Analysis:

SendWise is implemented as an Android Input Method Editor (IME). This allows the system to analyze text entered in different applications without requiring access to large amounts of device data or additional intrusive permissions. Text is processed using a temporary memory buffer with a maximum size of 500 characters. The buffer exists only in RAM during analysis. Once processing is complete, the buffer is cleared and the content is discarded. When the active application exposes a compatible editor action or input submission event through the Android IME framework, SendWise evaluates the text held by the active input field before the action is completed. Android's IME framework provides access to the active input connection and supports editor actions such as IME_ACTION_SEND. The implementation is therefore dependent on the input controls and editor actions exposed by the target application. If the confidence score exceeds 0.5, the user receives a warning message indicating that the content may be harmful. The user can then choose to edit the

message or continue sending it. The system does not prevent message transmission. Instead, it provides an opportunity for users to reconsider potentially harmful content before sending it. The final action taken by the user, such as revising the message or sending it unchanged, is recorded as behavioral metadata. At no stage is message content stored in persistent storage, written to logs, or transmitted outside the device. The current implementation does not guarantee coverage for application-specific custom input controls, interfaces that bypass standard IME editor actions, voice-only input, hardware-keyboard input, or messages received from other users. Pasted text is processed when it enters a supported input field, whereas received messages are not independently analyzed by the IME. These limitations restrict the system's coverage to compatible text-entry workflows. This sender-side scope is intentional because analyzing received communications would require broader access to incoming message content and would be inconsistent with the system's privacy-minimization objective.

2.2 Metadata Transmission:

If potentially harmful content is identified, the text is evaluated by the classification module. The module assigns a content category and calculates a severity score. To protect privacy, SendWise separates message processing into two distinct streams: message content and behavioral metadata.

1) Content Path (Local Only):

The original message remains on the user's device and continues directly to the application where it was created. SendWise does not store a copy of the message, save it to local storage, or send it to external services. The text is analyzed temporarily during processing and is removed from memory once analysis is complete.

2) Metadata Path (Transmitted):

When the classification probability reaches the predefined threshold of 0.5, the system creates a metadata entry describing the detected event. The threshold was selected as a balanced operating point between detecting potentially harmful messages and limiting unnecessary intervention prompts. The selected value was retained for the deployed prototype to avoid increasing false-positive interventions despite the higher recall observed at a lower threshold. This record contains only summary information about the interaction and does not include any message content. The fields included in the transmitted metadata record are summarized in Table 2.

Table 2. Sendwise transmitted metadata

Field	Description
user_id_hash	Pseudonymous user/device identifier
timestamp	Time of detected event
category	Assigned risk category
severity	Low, Medium, or High
action	User response to intervention
session_id	Temporary session identifier

The metadata record does not contain message text, recipient information, conversation history, or the linguistic features used by the classifier

The user_id_hash value is generated using the device Android ID together with a randomly generated salt and a SHA-256 hashing function. The salt is generated for the device and stored using the Android Keystore. Periodic salt rotation is not currently implemented; therefore, records associated with the same installation may remain linkable over time. The identifier should consequently be regarded as pseudonymous rather than anonymous.

The prototype retains only the metadata required for dashboard aggregation. The current prototype does not enforce an automated metadata retention period. A production deployment should enforce a defined metadata retention period and automatic deletion policy. Access to stored metadata should be restricted to the authenticated parent account associated with the corresponding child-device pairing. The current prototype supports explicit account unlinking through the dashboard; automated revocation policies beyond account unlinking are left for future deployment.

3) Transmission Protocol:

Metadata is sent to the backend service using HTTPS POST requests. All communication is protected using TLS 1.3 encryption. Certificate pinning is prepared in the network layer for deployment, while the current

prototype relies on TLS-protected communication without active pin enforcement. To prevent misuse of the service, the system limits each device to a maximum of 100 requests per hour. This threshold is sufficient for normal operation while helping to reduce malicious or excessive traffic.

4) Backend Infrastructure:

The backend is implemented using Next.js API routes and deployed through Vercel, with Supabase Authentication used for parent accounts and Upstash Redis used for metadata and pairing-related storage. The backend receives metadata records and validates incoming requests before storing them. Only metadata are stored; message content is never received by the server.

5) Privacy and Security Threat Model:

SendWise follows a data-minimization approach in which message content is processed locally and is not included in transmitted metadata. The privacy model assumes that the Android operating system and the SendWise application are not fully compromised. The system does not claim to protect message content from a rooted device, malware with equivalent device privileges, or a compromised operating system.

The backend receives only behavioral information generated after local classification. This information includes the detected category, severity, timestamp, action taken, and a pseudonymous user identifier. The server does not receive the original message text. This limits the amount of sensitive information available if backend data are exposed.

The use of a hashed identifier also reduces direct exposure of the device identity. However, hashing does not provide complete anonymity because repeated identifiers may still allow records from the same device to be linked. The current design therefore provides pseudonymization rather than full anonymity.

The Android IME implementation also has platform limitations. Applications that use custom input controls, restricted input fields, or alternative input mechanisms may not provide the same level of keyboard-based coverage. These limitations are considered in the interpretation of the system's deployment scope. The user-facing intervention interface used by SendWise is shown in Figure 2.

Figure 2. Original SendWise intervention interface designed by the authors

2.3 Parental Dashboard

Parents access the system through a secure web-based dashboard. Authentication is handled using standard web security mechanisms. The dashboard is designed to provide awareness of behavioral trends without exposing private con- versations. As a result, parents see summaries, activity patterns, and risk indicators rather than the original message content.

1) Linking Parent and Child Accounts:
The pairing mechanism uses a cryptographically generated six-digit code to establish the association between the child device and parent account without transmitting message content. The pairing code is stored with a 15-minute expiration period and is invalidated after successful redemption. The system also limits failed redemption attempts to reduce repeated guessing. Parent accounts can explicitly unlink an associated child device through the dashboard. The pairing process does not require transmission of personally identifiable message information.
2) Dashboard Visualizations:
The parental dashboard presents behavioral information using aggregated statistics and visual summaries. The objective is to provide awareness of online behavior patterns while maintaining the privacy of individual conversations.
Intervention Timeline: A time-series chart displays the number of detected interventions over a rolling 30-day period. Events are grouped according to severity level. This view allows parents to observe whether intervention frequency is increasing, decreasing, or remaining stable over time.
Category Distribution: A category summary chart shows the relative proportion of detected events across five risk categories: harassment, threats, hate speech, sexual content, and self-harm risk. This overview helps identify the types of issues that occur most frequently.
Action Summary: The dashboard reports how users respond to intervention prompts. Specifically, it displays the proportion of messages that were edited after a warning compared with those that were sent without modification. These statistics provide insight into how often users reconsider potentially harmful content.
Risk Indicators: Several quantitative indicators are calculated from the collected metadata. These include weekly intervention frequency, recent severity trends, and an overall risk score derived from the occurrence and severity of recent events. The indicators help parents identify patterns that may require additional attention.
Privacy-Focused Reporting: The dashboard is designed to display behavioral trends rather than conversation details. No message content, recipient information, or communication history is presented. To reinforce this design principle, the interface explicitly informs parents that only behavioral summaries are available and that message content remains private.

2.4 Classification algorithm

The classification component was designed to balance detection accuracy with the resource limitations of mobile devices. To achieve this goal, SendWise combines rule-based analysis with machine learning. Clear and explicit cases are identified using predefined rules, while more complex or ambiguous messages are evaluated using a Random Forest classifier
1) Category Definitions
The system considers five categories of harmful online communication relevant to adolescent safety.
Harassment: This category includes insults, name-calling, repeated unwanted contact, and aggressive language directed toward another person. Examples include statements such as “You’re so stupid” and “Nobody likes you.”
Threats: Threat-related content includes language suggesting physical harm, intimidation, or violence. Examples include statements such as “Watch your back” and “You’ll regret this.”
Hate Speech: This category covers discriminatory language directed at protected groups, including race, ethnicity, religion, gender, disability, or sexual orientation. Examples include slurs, derogatory stereotypes, and exclusionary statements.
Sexual Content: This category includes unwanted sexual communication, requests for explicit material, sexual coercion, and other forms of inappropriate sexual interaction.
Self-Harm Risk: This category covers messages indicating possible self-harm, suicidal intent, or serious emotional distress. Such messages require careful interpretation because expressions of distress may not always indicate immediate risk.
The rule-based component identifies explicit and predefined high-risk linguistic patterns before the machine-learning stage. Messages that are not resolved by the predefined rules are evaluated by the Random Forest classifier. An example of the parental dashboard presenting aggregated behavioral risk indicators is shown in Figure 3.

Figure 3. Original SendWise parental dashboard showing aggregated behavioural risk indicators

2) Machine Learning Classifier

Messages that are not confidently classified by the rule- based module are forwarded to a machine learning classifier. This stage is intended to identify implicit harassment, indirect threats, and other forms of harmful communication that may not be captured through keyword matching alone.

Model Architecture: The deployed system uses a Random Forest classifier because it provided the strongest overall detection performance among the evaluated conventional classifiers during model development. The model was selected for its relatively simple inference process and suitability for the intended mobile implementation.

Feature Engineering: Text input was converted into numerical representations using TF-IDF. Both word unigrams and bigrams were included to capture individual harmful terms as well as short phrase-level patterns. The vocabulary was limited to a maximum of 5,000 features, with a minimum document frequency of two. Importantly, the TF-IDF vocabulary and weighting parameters were fitted only on the training/development data and then applied unchanged to the held-out test set. No test-set information was used during feature construction. The configuration of the deployed Random Forest classifier is summarized in Table 3.

Table 3. Random forest configuration

Parameter	**Setting**
Classifier	Random Forest
Number of trees	200
Maximum tree depth	None
Class weighting	Balanced
Feature representation	TF-IDF
N-gram range	Unigram and bigram
Maximum features	5,000
Minimum document frequency	2
Probability threshold	0.5

Algorithm 1. SendWise Local Risk Detection

Input: Text entered through the supported Android IME

Output: Risk classification, category, severity, and intervention metadata

- Receive the active text input through the Android input connection.
- Store the text temporarily in memory.
- Apply text preprocessing.
- Apply predefined rule-based screening.
- If the rule-based stage identifies a directly matched pattern, assign the corresponding risk indication.
- Otherwise, transform the text using the TF-IDF representation.
- Apply the trained Random Forest classifier.
- If the predicted risk probability is ≥ 0.5, classify the input as risk-related; otherwise, classify it as non-risk.
- If risk-related, determine the category and severity.
- Display the appropriate warning to the user.
- Generate behavioral metadata without including the original message.
- Transmit only the metadata record through the protected backend connection.
- Clear the temporary message buffer.

A threshold of 0.5 was selected for the deployed prototype as a practical operating point between detection sensitivity and unnecessary intervention prompts. Although a lower threshold increased recall, it also increased the number of non-risk messages classified as risk-related. Because each detected event can generate an intervention prompt, the selected threshold was retained to limit unnecessary interruptions while maintaining high recall.

The dataset contains substantially more non-risk than risk-related messages. Instead of artificially oversampling the minority class, the Random Forest classifier used balanced class weighting during training. This approach preserves the original data distribution while assigning greater training importance to risk-related examples.

The Random Forest model was selected as the deployed classifier after exploratory comparison with alternative conventional machine-learning approaches. The final deployed configuration was subsequently fixed for evaluation. Class weighting was used to account for the imbalanced risk/non-risk distribution. Feature extraction was fitted only on the development/training data.

The Random Forest configuration was kept fixed during final evaluation. Feature extraction was fitted using the training data only. The held-out test set was used only for final performance measurement. The comparative performance of the evaluated conventional machine-learning classifiers is presented in Table 4.

Table 4. Comparison of machine learning classifiers

Classifier	Precision (%)	Recall (%)	F1-score (%)
SVM	~75	~81	~78
KNN	~63	~67	~65
Logistic Regression	~69	~72	~70
Naive Bayes	~85	~54	~66
Gradient Boosting	~71	~49	~58
Random Forest	~86	~95	~90

Several conventional machine learning classifiers were evaluated during model development, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Naive Bayes, Gradient Boosting, and Random Forest. Random Forest was selected for deployment based on its suitability for the intended on-device implementation, including its compact model representation, straightforward inference process, and compatibility with the resource constraints of the Android environment. The final Random Forest configuration was subsequently fixed before evaluation on the held-out test set.

Training Data: The complete dataset contains 20,122 manually labelled messages. A predefined 75:25 partition was used, resulting in 15,091 training messages and 5,031 held-out test messages. The training set contains 13,191 non-risk and 1,900 risk-related messages. The test set contains 4,398 non-risk and 633 risk-related messages. The class distribution was preserved across the two partitions.

Performance: Several machine learning models were evaluated during development, including logistic regression, support vector machines, and Random Forest. Random Forest was selected for deployment based on its comparative classification performance during model development. Evaluation was performed using the predefined held-out test set described in the Experimental Setup section. The selected classifier achieved 85.96% precision, 95.73% recall, and a 90.58% F1-score at the selected probability threshold.

Severity Scoring

Severity is calculated from four factors: language intensity, threat explicitness, discriminatory content, and contextual structure. Each factor contributes to the final severity assessment. The resulting score is mapped to three levels: Low, Medium, and High.

Messages containing multiple offensive expressions receive higher severity ratings than isolated incidents. Explicit threats are weighted more heavily than vague statements, and hate speech directed at protected groups is assigned greater importance because of its potential impact. These severity levels are later used in dashboard analytics and risk calculations.

$$Severity\ Score = 0.30L + 0.30T + 0.25D + 0.15C. \quad (1)$$

where:

L = language intensity

T = threat explicitness

D = discriminatory content

C = contextual structure.

The factors used to determine severity and their effects on the final assessment are summarized in Table 5.

Table 5. Severity assessment factors

Factor	Description	Effect on severity
Language intensity	Strength and repetition of offensive language	Higher intensity increases severity
Threat explicitness	Difference between vague and direct threats	Direct threats receive higher severity
Discriminatory content	Presence of hate or discriminatory targeting	Targeted discriminatory content increases severity
Contextual structure	Coherence and apparent intent of the message	More explicit or structured harmful content increases severity

2.5 Experimental Setup

The SendWise evaluation was performed using the manually annotated corpus of 20,122 messages. The dataset was divided using a stratified 75:25 partition, producing 15,091 messages for model development and 5,031 messages for final held-out evaluation. The class distribution was preserved across the two partitions. The development portion was used for model development and classifier comparison, while the held-out test set of 5,031 messages was reserved for final performance evaluation. The held-out test set was used for final performance evaluation of the selected Random Forest configuration. Text preprocessing was applied before feature extraction. The message text was represented using TF-IDF features with word-level unigrams and bigrams. The maximum vocabulary size was limited to 5,000 features and the minimum document frequency was set to two. Feature extraction was fitted using the development/training data and subsequently applied to the held-out test data.

The final deployed classifier was a Random Forest model containing 200 trees with no fixed maximum depth and balanced class weighting. A probability threshold of 0.5 was used for the final risk/non-risk decision. Threshold sensitivity was additionally examined to evaluate the trade-off between false-positive and false-negative detection.

The primary evaluation metrics were precision, recall, F1-score, accuracy, specificity, ROC-AUC, and PR-AUC. Confidence intervals were estimated using bootstrap resampling of the held-out test predictions. The experimental software environment comprised Android Studio Ladybug 2024.2.1, Android SDK API level 35 (Android 15), Python 3.12.7, scikit-learn 1.5.2, and the backend runtime Node.js 26.0.0, running on macOS 26.5.2 with an Apple M4 Pro (14-core CPU) and 24 GB unified memory.

The SendWise prototype was functionally evaluated on three Android devices representing different hardware configurations: Redmi Note 7 Pro, Redmi 8A, and OnePlus Nord CE5. Testing focused on installation, keyboard operation, local classification, warning generation, and metadata transmission. The device evaluation was intended to verify functional operation across different Android configurations rather than provide a controlled hardware benchmark.

Functional testing was conducted using the SendWise custom IME across multiple social networking and messaging applications, including WhatsApp as the representative application. The overall composition of the dataset used for evaluation is presented in Table 6.

Table 6. Dataset composition

Class	Samples	Percentage
Non-risk	17,589	87.41%
Risk-related	2,533	12.59%
Total	20,122	100%

The distribution of messages between the development/training and held-out test partitions is shown in Table 7.

Table 7. Dataset split

Partition	Total	Non-risk	Risk-related
Development/Training	15,091	13,191	1,900
Held-out Test	5,031	4,398	633
Total	20,122	17,589	2,533

The dataset was partitioned using a stratified 75:25 split. The 15,091-message development portion was used for model development and classifier comparison, while the 5,031-message test portion was reserved for final evaluation. No separate fixed validation set was used. The risk-related class represents 12.59% of the complete dataset. Class proportions were retained across the predefined training and test partitions. The distribution of messages across the five risk categories is presented in Table 8.

Table 8. Risk category distribution

Risk Category	Number of Messages
Harassment	950
Threats	550
Hate speech	400
Sexual content	333
Self-harm risk	300
Total risk-related messages	2,533

The hardware and software configurations of the three Android devices used for functional testing are summarized in Table9.

Table 9. Evaluation device configuration

Parameter	Redmi Note 7 Pro	Redmi 8A	OnePlus Nord CE5
Android version	Android 10 / MIUI 12	Android 10 / MIUI 12	Android 15 / OxygenOS 15
RAM	6 GB	4 GB	8 GB
CPU / SoC	Snapdragon 675	Snapdragon 439	Dimensity 8350 Apex
Battery capacity	4,000 mAh	5,000 mAh	7,100 mAh
Network	Jio 4G	Jio 4G	Jio 4G / 5G
Keyboard	SendWise Custom IME	SendWise Custom IME	SendWise Custom IME
Classification runs	≥50	≥50	≥50
Measurement	Functional testing using ADB, logcat and timestamps	Functional testing using ADB, logcat and timestamps	Functional testing using ADB, logcat and timestamps

2.6 Dataset Description

The dataset used in this study was gradually collected during 2025 from publicly accessible content on YouTube and X. The corpus was independently collected by the authors rather than assembled from previously published benchmark datasets. Publicly accessible textual content relevant to harmful-language detection was collected from YouTube and X and subsequently annotated using the SendWise annotation scheme. The collection focused on publicly visible textual content relevant to online communication and harmful-language detection. No private accounts, private conversations, or restricted user content were accessed. The collected content was subsequently reviewed and manually annotated according to the SendWise annotation scheme. A team of five annotators participated in the annotation process. The annotators used common annotation guidelines defining non-risk content and five risk categories: harassment, threats, hate speech, sexual content, and self-harm risk. Annotations were independently reviewed and subsequently cross-verified within the five-member annotation team, and disagreements were discussed and resolved according to the predefined guidelines. The annotation process was intended to improve consistency in assigning labels across different linguistic expressions and contexts. The final corpus contains 20,122 messages, comprising 17,589 non-risk messages (87.41%) and 2,533 risk-related messages (12.59%). The corpus contains both English and Hinglish text. Message content was used only as input to the classification pipeline; user identifiers and unrelated metadata were not used as classifier features. Before model development, duplicate message texts were checked and removed where applicable. The final dataset contained no exact duplicate texts across the training and test partitions. Text preprocessing and feature extraction were performed using the training portion only to prevent information from the held-out test set from influencing model development. Because the source material was collected from public online platforms, redistribution of raw message text may be subject to the terms and rights associated with the original platforms. Therefore, the accompanying repository provides the experimental code, dataset schema, annotation definitions, preprocessing procedure, and derived dataset statistics, while raw text is redistributed only where permitted.

Public content was collected incrementally during 2025 using predefined searches related to harmful online communication, cyberbullying, harassment, threats, hate speech, and related safety concerns. Only publicly visible textual content was considered. Entries that were inaccessible, duplicated, non-textual, or unsuitable for annotation were excluded before labeling. The collected messages were then manually reviewed and assigned to the SendWise annotation categories using the predefined annotation guidelines.

The corpus was collected specifically for this study and was not assembled by combining previously published benchmark datasets. Therefore, no source-dataset label mapping was required.

The annotation process used independent review and subsequent cross-verification among the five annotators. Formal inter-annotator agreement statistics were not recorded during the initial annotation process, and the original annotation records do not contain sufficient per-annotator labels to reconstruct an agreement coefficient reliably. Therefore, an inter-annotator agreement statistic is not reported. The use of five annotators, independent review, cross-verification, and predefined annotation guidelines were used to improve consistency during dataset construction. A summary of the dataset provenance, annotation process, and associated attributes is provided in Table 10.

Table 10. Dataset provenance and annotation summary

Attribute	Description
Collection period	2025
Source platforms	YouTube and X
Collection type	Publicly accessible textual content
Total messages	20,122
Non-risk messages	17,589 (87.41%)
Risk-related messages	2,533 (12.59%)
Languages	English and Hinglish
Annotation	Manual annotation
Annotators	5
Verification	Cross-verification within annotation team

Risk categories	Harassment, Threats, Hate Speech, Sexual Content, Self-Harm Risk
Exact duplicate check	Performed before final partition
Classifier input	Message text
User identifiers used as features	No
Collection year	2025

The operational definitions used for the annotation categories are presented in Table 11.

Table 11. Sendwise annotation categories

Category	Annotation definition
Non-risk	Neutral, informational, conversational, or critical content without targeted harmful intent
Harassment	Insults, abusive language, repeated unwanted communication, or aggressive language directed toward another person
Threats	Explicit or implicit statements indicating physical harm, intimidation, or violence
Hate speech	Discriminatory or derogatory language targeting a protected or identifiable social group
Sexual content	Unwanted sexual communication, explicit sexual requests, sexual coercion, or inappropriate sexual interaction
Self-harm risk	Content indicating possible self-harm, suicidal intent, or serious emotional distress

The corpus was manually annotated directly using the SendWise annotation scheme; therefore, no source-dataset label conversion was required.

2.7 Computational Considerations

SendWise was designed to perform classification locally using a compact Random Forest model and a limited TF-IDF feature representation. Classification is triggered only when supported text-entry events require analysis rather than continuously during typing. This design reduces unnecessary processing and avoids transmitting message content for remote analysis.

Functional testing was performed on the three Android devices listed in Table 9. The tests covered keyboard activation, text entry, local classification, warning generation, and metadata transmission. The prototype operated successfully during the functional tests on all three devices.

A controlled quantitative benchmark of memory consumption, CPU utilization, battery consumption, and device-specific inference latency was not retained from the prototype evaluation. Therefore, these values are not reported as measured results in this study. A controlled hardware benchmark across different Android devices is identified as an area for future work. ADB, logcat, and timestamp-based observations were used for functional verification but were not treated as controlled quantitative measurements of CPU, RAM, battery consumption, or inference latency.

3. Results

1) Accuracy Metrics:

The Random Forest classifier was evaluated on the held-out test set containing 5,031 messages. The deployed Random Forest classifier produced 4,299 true negatives, 99 false positives, 27 false negatives, and 606 true positives. Precision was 85.96%, recall was 95.73%, and the F1-score was 90.58%. Overall accuracy was 97.50%, while specificity was 97.75%. These results indicate that the classifier identified most risk-related messages while limiting false alarms among non-risk messages. The corresponding confusion matrix for the held-out test set is presented in Table 12.

Table 12. Confusion matrix on the held-out test set

	Predicted Non-risk	Predicted Risk
Actual Non-risk	4,299	99
Actual Risk	27	606

The confusion matrix shows that 606 of the 633 risk-related test messages were correctly detected, while 27 were missed. Among the 4,398 non-risk messages, 99 were incorrectly flagged. Class-wise precision, recall, and F1-score results are summarized in Table 13.

Table 13. Class-wise classification performance

Class	Precision	Recall	F1-score	Support
Non-risk	99.38%	97.75%	98.56%	4,398
Risk	85.96%	95.73%	90.58%	633
Macro average	92.67%	96.74%	94.57%	5,031
Weighted average	97.69%	97.50%	97.55%	5,031

2) Category-Specific Performance:

Detection performance varied across categories because of differences in language patterns and contextual complexity. Harassment-related content generally produced the highest accuracy because it often contains direct linguistic indicators such as insults, abusive language, or repeated negative expressions. Threat detection was moderately challenging. While explicit threats were identified reliably, distinguishing genuine threats from jokes, sarcasm, or playful exchanges required additional contextual interpretation. Hate speech presented further challenges because some terms may be used offensively in one context and non- offensively in another. The interpretation of reclaimed language and community-specific expressions also affected classification performance. Sexual content was the most difficult category to detect consistently. Users frequently employ indirect language, abbreviations, slang, or euphemisms, making classification more complex than direct keyword-based detection.

3) Threshold Sensitivity:

The classification threshold was evaluated at multiple operating points to examine the trade-off between precision and recall. Although higher thresholds increased precision and produced higher F1-scores at some operating points, they also reduced recall. A threshold of 0.5 was retained for the deployed prototype as a balanced operating point between recall and false-positive intervention. Although a lower threshold increased recall, it also increased the number of non-risk messages classified as risk-related. Because each detected event can generate an intervention prompt, the selected threshold was intended to avoid unnecessary interruptions while maintaining high recall. The effect of different probability thresholds on precision, recall, and F1-score is shown in Table 14.

Table 14. Threshold sensitivity analysis

Probability Threshold	Precision	Recall	F1-score
0.4	86.42%	99.53%	92.51%
0.5	85.96%	95.73%	90.58%
0.6	87.00%	93.05%	89.92%
0.7	87.86%	92.58%	90.15%
0.8	90.80%	91.94%	91.37%
0.9	97.00%	91.63%	94.23%

4) ROC and Precision-Recall Analysis:

The classifier achieved a ROC-AUC of 99.82%, indicating strong separation between risk-related and non-risk messages across probability thresholds. Because the dataset is imbalanced, precision-recall performance was also examined. The resulting PR-AUC was 98.85%. These measures complement the reported precision, recall, and F1-score and provide a broader view of classifier performance. The estimated performance metrics together with their 95% confidence intervals are presented in Table 15.

Table 15. Test-set performance with 95% confidence intervals

Metric	Estimate	95% Confidence Interval
Precision	85.96%	83.33–88.53%

Recall	95.73%	94.05–97.21%
F1-score	90.58%	88.84–92.20%
Accuracy	97.50%	97.06–97.91%

For each of 5,000 bootstrap replicates, test-set observations were sampled with replacement and the evaluation metric was recalculated. The 2.5th and 97.5th percentiles of the resulting bootstrap distribution were used as the 95% confidence interval.

5) Device Compatibility:

The prototype was tested on Redmi Note 7 Pro, Redmi 8A, and OnePlus Nord CE5 devices. Testing focused on installation, keyboard activation, text entry, local classification, warning generation, and metadata transmission. The application operated successfully during the functional tests on all three devices. These tests demonstrate cross-device functional feasibility, but they should not be interpreted as a controlled quantitative benchmark of Android hardware performance.

The results demonstrate that privacy-preserving parental awareness can be achieved without transmitting or storing message content. By performing classification directly on the device and sharing only behavioral metadata, SendWise reduces exposure of sensitive information while still providing parents with indicators of potentially harmful online interactions.

The evaluation showed that the Random Forest classifier achieved strong recall while maintaining acceptable precision. On the held-out test set, the classifier achieved 85.96% precision, 95.73% recall, and a 90.58% F1-score. This balance is important in safety-related applications, where missed incidents may be more problematic than occasional false positives. The cross-device functional tests indicate that the prototype can operate across different Android hardware configurations. However, quantitative claims about memory, CPU, battery, and latency require controlled benchmarking and are therefore left for future evaluation.

These results should be interpreted together with the privacy trade-off of the system. SendWise intentionally provides parents with less information than content-monitoring systems. A parent cannot inspect the original message or reconstruct the conversation from the reported metadata. This limits contextual understanding but also reduces unnecessary exposure of adolescent communications. The proposed design therefore prioritizes data minimization rather than maximum parental visibility.

A key design decision in SendWise is the separation of awareness from surveillance. Rather than providing parents with direct access to conversations, the system reports aggregate behavioral patterns and risk indicators. This approach is intended to support parental involvement while reducing privacy concerns commonly associated with content-monitoring systems.

The system also emphasizes user autonomy. When potentially harmful content is detected, users receive a warning but remain free to decide how to proceed. This design differs from enforcement-based approaches that block communication or impose restrictions. Such an approach may encourage reflection and responsible online behavior while preserving the user's ability to make decisions independently.

Despite these advantages, several limitations should be acknowledged. First, the system does not provide parents with conversational context. While this protects privacy, it may also limit the ability to interpret certain events accurately. Second, cyberbullying detection remains a challenging task because harmful intent is often context-dependent. Sarcasm, humor, cultural expressions, and evolving slang may affect classification performance. Third, the current evaluation focused primarily on technical performance and system feasibility. Long- term studies involving parents and adolescents are needed to understand how the system influences trust, communication, and online behavior over time.

4. Discussion

4.1 Comparison with Existing Systems

Existing parental monitoring solutions generally follow one of two approaches. Some systems provide extensive access to communication content, while others rely primarily on restrictive controls such as application blocking or screen-time management.

1) Content-Monitoring Systems:

Commercial platforms such as Bark and Net Nanny provide alerts together with message excerpts when potentially harmful content is detected. This approach offers detailed contextual information but re- quires access to private communications. In contrast, SendWise limits reporting to behavioral metadata. As a result, parents receive less detailed information, but the privacy of individual conversations is preserved.

2) Restrictive Control Systems:

Screen-time controls and application-blocking tools focus on limiting access to digital platforms rather than analyzing communication behavior. Although such approaches may reduce exposure to online risks, they may also restrict legitimate educational and social interactions. SendWise instead allows communication to continue while providing information about potential behavioral concerns.

3) No-Monitoring Approaches:

Families that choose not to use monitoring technologies avoid privacy concerns associated with digital oversight. However, they may have limited visibility into emerging online risks. SendWise attempts to provide a compromise by offering awareness of behavioral patterns without exposing message content.

4.2 Ethical Considerations

The deployment of parental monitoring technologies raises important ethical questions regarding privacy, autonomy, and parental responsibility. Adolescents have legitimate interests in maintaining personal privacy, while parents have a respon- sibility to protect children from online harm. Balancing these interests remains a significant challenge for designers of digital safety systems.

The proposed architecture addresses this challenge by minimizing data collection and limiting the information shared with parents. Message content remains on the device and is not transmitted to external services. Only aggregated behavioral metadata are reported through the dashboard.

Transparency is equally important for ethical deployment. Adolescents should be informed about the purpose of the system, the information being collected, and how that information is used. Open discussion of monitoring practices may help maintain trust and encourage responsible digital behavior. Consequently, SendWise is intended to complement family communication and digital safety education rather than replace them.

From a data-governance perspective, the system follows a data-minimization approach. Message content is not required by the parental dashboard and is excluded from backend storage. The metadata retained by the system should be limited to the fields required for risk summaries and trend analysis. Access to the dashboard should be restricted to the authorized parent account. The current prototype does not claim to provide complete anonymity or protection against a compromised device. These limitations should be communicated clearly during deployment.

4.3 Preliminary Parent Feedback

During the early development of SendWise, the concept was discussed informally with a small group of parents. The discussions were intended to understand whether parents considered behavioral risk indicators useful when message content was not available to them. Participants generally responded positively to the privacy-preserving approach and expressed interest in receiving information about emerging risk patterns without direct access to private conversations.

These discussions were exploratory and were not conducted as a formal user study. No standardized questionnaire, statistical usability assessment, or controlled evaluation was performed. Therefore, the feedback is presented only as preliminary design input and not as evidence of user acceptance or system effectiveness.

4.4 Limitations and Future Research

1) Current Implementation Limitations:

Several limitations of the current implementation should be acknowledged. First, the prototype was developed for Android devices and has not yet been evaluated on other mobile platforms. Second, the classification model was trained on a corpus containing both English and Hinglish text. Performance may vary when applied to multilingual conversations, regional slang, or culturally specific expressions. The current system also relies on a fixed classification model that is shared across users. While this approach sim- plifies deployment and evaluation, it may not fully capture differences in communication styles among individuals, age groups, or communities. In addition, message classification is performed on individual messages rather than extended conversation histories. Consequently, some contextual information that could improve interpretation is not considered. Finally, devices operate independently and require explicit account linking for multi-device use. Although this design supports privacy and data separation, it may reduce convenience for families managing multiple devices.

The current evaluation also has several methodological limitations. The final test set represents the data distribution used during development and may not represent all forms of adolescent online communication. The reported results therefore should not be interpreted as evidence of generalization to every messaging platform or population. In addition, the current evaluation focuses on message-level classification and does not establish whether the detected patterns correspond to sustained cyberbullying relationships over time. The dataset was collected from publicly accessible YouTube and X content and therefore may not fully represent private adolescent conversations or the communication patterns of younger users. The Android IME

approach depends on application compatibility with standard input controls and editor actions. Applications using custom input mechanisms or alternative communication paths may not be fully covered. A formal inter-annotator agreement analysis was not performed during the initial dataset construction. Future work will include agreement statistics such as Cohen's or Fleiss' kappa to quantify annotation consistency. The current evaluation also has a device-performance limitation. Although SendWise was functionally tested on three Android devices, controlled measurements of memory consumption, CPU utilization, battery usage, and inference latency were not retained. Consequently, the present results demonstrate functional feasibility rather than a comprehensive hardware-performance benchmark. Future work will conduct repeated measurements under controlled conditions across a broader range of Android devices.

2) Future Research Directions:

Several opportunities exist for extending the proposed system. One promising direction is the use of federated learning, which would allow models to improve using distributed user data while maintaining privacy protections [15]. Such an approach could support continuous model improvement without requiring the collection of raw message content. Future work may also investigate methods for incorporating conversational context, communication history, and behavioral patterns into the detection process. These additions could improve the identification of subtle forms of harassment, manipulation, and emerging cyberbullying behavior. Another area for exploration is adaptive personalization. User-specific or family-specific sensitivity settings may allow the system to better accommodate different communication styles, age groups, and parental preferences while maintaining privacy guarantees. Most importantly, long-term user studies are needed to evaluate real-world effectiveness. Future research should examine not only detection accuracy but also broader outcomes such as parental awareness, adolescent perceptions of privacy, trust within families, and the system's influence on online behavior over extended periods.

5. Conclusions

This study presented SendWise, a parental monitoring system designed to improve awareness of online risks while protecting user privacy. Unlike conventional monitoring tools that provide access to message content, SendWise performs analysis directly on the device and shares only behavioral metadata with parents. The proposed architecture combines on-device detection, secure metadata transmission, and dashboard-based reporting. The finalized Random Forest classifier achieved 85.96% precision, 95.73% recall, and a 90.58% F1-score on the held-out test set. These results demonstrate the technical feasibility of implementing the proposed privacy-preserving monitoring architecture on the evaluated Android devices. A central contribution of this work is the separation of parental awareness from content surveillance. Parents receive information about behavioral trends and potential risks without access to private conversations. This design seeks to support safety while respecting the privacy and autonomy of adolescents. The current implementation represents an initial step to- ward privacy-aware family safety technologies. Future work should evaluate the system in long-term real-world settings and examine its effects on parental awareness, adolescent trust, and online behavior. Further research may also explore multilingual support, personalized models, and federated learning approaches to improve detection performance while maintaining privacy protections. Overall, the findings suggest that meaningful parental awareness can be achieved without collecting or exposing message content. Privacy-preserving approaches such as SendWise may offer a practical alternative to traditional surveillance-based monitoring systems.

- **Author Contributions**

Conceptualization, N.G. and S.O.; methodology, N.G.; software, N.G.; validation, N.G. and S.O.; formal analysis, N.G.; investigation, N.G.; data curation, N.G.; writing—original draft preparation, N.G.; writing—review and editing, N.G. and S.O.; visualization, N.G.; supervision, S.O. Both authors have read and agreed to the published version of the manuscript.

- **Funding**

This research received no external funding.

- **Data Availability Statement**

The source code, dataset, dataset schema, annotation guidelines, preprocessing procedure, and derived dataset statistics are available through the authors' public repository [18]. The corpus was collected from publicly accessible third-party platforms, and redistribution of individual source messages is subject to the terms and rights applicable to the original platforms. The repository includes the dataset used for the reported

experiments together with the associated annotation definitions, preprocessing procedure, and experimental configurations.

- **Ethics Statement**

The present work focused on the technical development and evaluation of a privacy-preserving prototype. No formal human-subject experiment involving children or adolescents was conducted as part of the reported evaluation. Informal discussions with parents were used only as preliminary design feedback and were not treated as a formal user study. Future studies involving parents or adolescents will require appropriate informed consent or assent procedures and institutional ethical review where applicable. Because the study used publicly accessible online content and did not involve recruitment or experimentation with children or adolescents, no formal human-subject ethics approval was required for the reported technical evaluation. The system is designed to minimize exposure of potentially sensitive information by excluding message content from transmitted and stored metadata. The reported prototype does not collect private conversations or directly recruit minors for evaluation. Any future deployment involving children or adolescents should follow applicable child-data protection, consent, and institutional requirements, including relevant requirements under the General Data Protection Regulation (GDPR), the Children's Online Privacy Protection Act (COPPA) where applicable, and India's Digital Personal Data Protection Act, 2023. Future studies involving minors should also incorporate appropriate parental or guardian consent, adolescent assent where applicable, and institutional ethical review.

- **Acknowledgments**

The authors would like to thank MKSSS's Cummins College of Engineering for Women, Pune, for providing the facilities and support required to carry out this research.

- **Conflicts of Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

- **Generative AI Statement**

During the preparation of this work, the authors used OpenAI ChatGPT to assist in improving the presentation, organization, and academic writing of the manuscript. The research concept, system design, methodology, implementation, experimental results, and conclusions are the authors' original work. After using this tool, the authors reviewed and edited all AI-assisted content as needed and take full responsibility for the content of this publication.

References

1. R. M. Kowalski, G. W. Giumetti, A. N. Schroeder, and M. R. Lattanner, “Bullying in the Digital Age: A Critical Review and Meta-Analysis of Cyberbullying Research among Youth,” Psychological Bulletin, vol. 140, no. 4, pp. 1073–1137, 2014. DOI 10.1037/a0035618
2. J. W. Patchin and S. Hinduja, Summary of Our Cyberbullying Research (2007–2019). Cyberbullying Research Center, 2019.
3. Bark Technologies, Bark for Families: Comprehensive Online Safety Monitoring, 2024. [Online]. Available: https://www.bark.us [Accessed: Aug. 24, 2026].
4. S. T. Hawk, W. W. Hale, Q. A. Raaijmakers, and W. Meeus, “Adoles- cents’ Contact with Friends: The Role of Maternal Monitoring,” Journal of Research on Adolescence, vol. 18, no. 3, pp. 401–419, 2008.
5. P. Wisniewski, H. Jia, H. Xu, M. B. Rosson, and J. M. Carroll, “’Preven- tative’ vs. ’Reactive’: How Parental Mediation Influences Teens’ Social Media Privacy Behaviors,” in Proc. 18th ACM Conf. Computer Supported Cooperative Work & Social Computing (CSCW), 2015, pp. 302–316. DOI: 10.1145/2675133.2675293
6. S. Livingstone and E. J. Helsper, “Parental Mediation of Children’s Internet Use,” Journal of Broadcasting & Electronic Media, vol. 52, no. 4, pp. 581–599, 2008. DOI: 10.1080/08838150802437396
7. M. Kerr and H. Stattin, “What Parents Know, How They Know It, and Several Forms of Adolescent Adjustment: Further Support for a Reinterpretation of Monitoring,” Developmental Psychology, vol. 36, no. 3, pp. 366–380, 2000. DOI: 10.1037/0012-1649.36.3.366
8. H. Stattin and M. Kerr, “Parental Monitoring: A Reinterpretation,” Child Development, vol. 71, no. 4, pp. 1072–1085, 2000. DOI: 10.1111/1467-8624.00210

9. A. Cavoukian, Privacy by Design: The 7 Foundational Principles. Information and Privacy Commissioner of Ontario, 2009.
10. S. Yardi and A. Bruckman, “Social and Technical Challenges in Parenting Teens’ Social Media Use,” in Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), 2011, pp. 3237–3246. DOI: 10.1145/1978942.1979422
11. Qustodio, Parental Control and Digital Wellbeing, 2024. [Online]. Available: https://www.qustodio.com [Accessed: Aug. 24, 2026].
12. T. Chakrabarty, K. Gupta, and S. Muresan, “Pay ’Attention’ to Your Context When Classifying Abusive Language,” in Proc. Third Workshop on Abusive Language Online, 2019. DOI: 10.18653/v1/W19-3508
13. C. Dwork, “Differential Privacy,” in Proc. 33rd Int. Colloquium on Au- tomata, Languages and Programming (ICALP), Part II. Berlin, Germany: Springer, 2006, pp. 1–12.
14. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273–1282.
15. T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” Proc. International AAAI Conference on Web and Social Media, vol. 11, no. 1, pp. 512–515, 2017. DOI: 10.1609/icwsm.v11i1.14955
16. K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard, “Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying,” ACM Transactions on Interactive Intelligent Systems, vol. 2, no. 3, Art. 18, 2012.
17. I. A. Raj and R. Vidya, "Adaptive Ensemble Learning with a Fine-Tuned Framework for Cyberbullying Detection in Cross-Platform Social Media Environments," Engineering, Technology & Applied Science Research, vol. 16, no. 1, pp. 31809–31813, Feb. 2026. DOI: 10.48084/etasr.15164.
18. N. Gaikwad and S. Ohatkar, “SendWise Dataset and Reproducibility Materials,” GitHub, 2026. [Online]. Available: https://github.com/NamrataG7/SendWise [Accessed: Aug. 27, 2026]