A Statistics About the Dataset

The PANC dataset (see Table 3) was split into a training set (60%) and a test set (40%). The training set consists of 1,753 positive segments (representing in total 298 full-length positive conversations and 9% of the training examples) and 17,598 negative segments, whereas the test set contains 11% examples of grooming.

B Illustrations of our framework

B.1 The Training Phase

Figure 2 illustrates the training phase of our framework. A global server selects clients to participate and distributes a model to them; the clients will then further train the model in a privacy-preserving manner on their mobile devices using their own personal data as well as a portion of warm-up data, as we can see in Alice's cellular device.

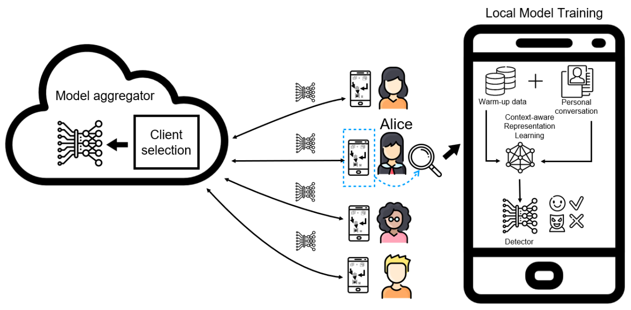


Figure 2: eSPD: Training Phase

B.2 The Inference Phase

In Figure 3, we show how different messages received by Alice are analyzed by first being turned into word embeddings and then passed to a classifier given a sliding window for classification. Note that the final prediction is determined based on the previous sequence of predictions and that a warning notification is triggered only when multiple messages are sequentially classified as being grooming messages.

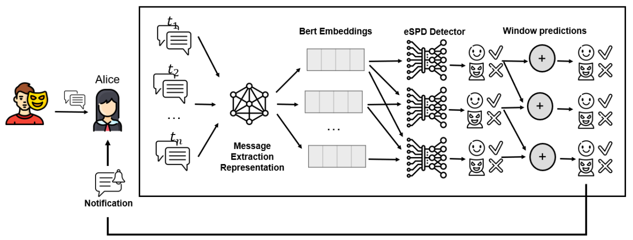


Figure 3: eSPD: Inference Phase

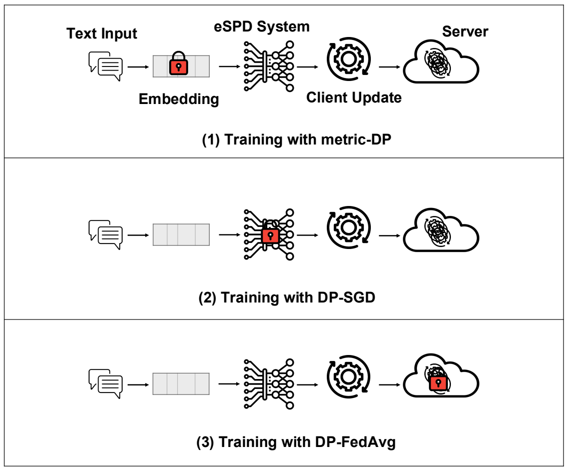


Figure 4: Illustration of the different privacy implementations: a lock representing DP is applied first to the embedding representation, then to the local training, and finally to the global training.

B.3 The Privacy Implementations

B.4 The Deployed System

In Figure 5, we present a visualization of a synthetic setup based on the proposed framework using a predatory conversation from the PANC dataset. It can take weeks or even months before a warning notification is triggered when a child is being lured by an abuser. Our goal is to minimize the harm by detecting the abuse early and sending a notification to the user. It is up to the user to decide whether to continue the conversation or report the predator. Note that in our framework, both training and inference phases are happening locally and users' personal conversations are never shared with a third party. Moreover, the global aggregated model from the server can further be tuned and personalized based on users' local data. In Figure 3, we show how different messages received by Alice are analyzed by first being turned into word embeddings and then passed to a classifier given a sliding window for classification. Note that the final prediction is determined based on the previous sequence of predictions and that a warning notification is triggered only when multiple messages are sequentially classified as being grooming messages.

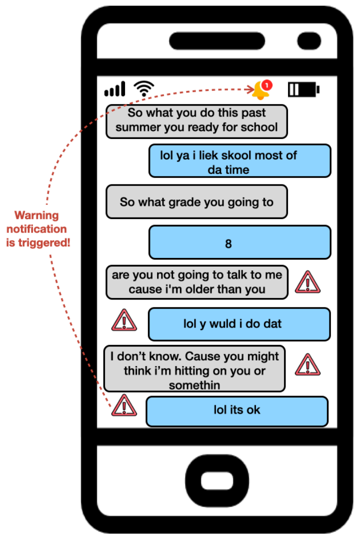
C Experimental Set-Up

In this section, we present the experimental details of our implementation.

FL Implementation. We use Flower (Beutel et al. 2020), an FL framework that facilitates large-scale experiments through its simulation tools, to implement our setup. For all the federated models except the FL models with DP-FedAvg, we collaboratively train an LR model with $10,000$ clients for 100 rounds. At each round of training, we select 10% of the clients randomly to participate in the training. At the end of each round, the parameters are aggre-

	Number of segments		# Words/segment		# Messages/segment	
Label	train	test	train	test	train	test
0	$17,598$ (91%)	$11,733$ (89%)	173 ($\pm1,385$)	184 (±1529)	36 (±25)	36 (±26)
1	$1,753$ (9%)	$1,426$ (11%)	289 (±218)	292 (±222)	64 (±43)	65 (±43)

Table 3: Statistics about the PANC dataset (Vogt et al. 2021).


So what you do this past summer you ready for school

lol ya i liek skool most of da time

So what grade you going to

8

are you not going to talk to me cause i'm older than you

lol y wuld i do dat

I don't know. Cause you might think i'm hitting on you or somethin

lol its ok

Figure 5: Visualization of eSPD in which the risk is detected, a warning is raised after passing a threshold, and the user is notified as early as possible.

.

gated with the FedAvg algorithm (McMahan et al. 2017a). The optimal number of rounds was determined by following the evolution of the validation loss of different models during training whereas the number of clients to sample for training was chosen with the help of hyperparameters tuning. For the FL model trained with DP-FedAvg, we experiment with $(50, 100, 200, 1000, 2000)$ clients sampled at each round and $(25, 50, 75, 100, 125, 150)$ rounds of training since these parameters have an impact on the overall privacy budget. The best hyperparameters for the federated models have been chosen using a random search.

Warm-up data creation. To implement the data-sharing strategy needed for training with non-IID data, we split the training set into three while making sure that each set contains different users: 10% of the dataset is randomly selected to create the warm-up data, and the rest is split between a training set (81%) and a validation set (9%). To ensure that no bias came from the warm-up split, we repeat the process three times and test our model with every split.

Client initialization. We create each client by randomly selecting one user from the training set. In our dataset, each user corresponds to a unique conversation, either predatory (OG user), and therefore constituted of multiple segments

	Number of segments	
Label	OG user	non-OG user
0	10	21
1	18 (±14)	10

Table 4: Statistics about the data distribution in a federated setting. The training set contains 15,125 non-predatory conversations and 86 predatory conversations.

of data, or non-predatory (non-OG user), and therefore constituted of a unique segment of data. If the selected user has a negative label (non-OG user), we select 10 additional users with negative labels and combine their data to compensate for the lack of non-grooming examples. Note that this was done to compensate for the shortcomings of the PANC dataset and will not be applied in a real-life scenario.

Finally, at initialization, each client receives a random, balanced portion of the warm-up data: 10 segments with a negative label and 10 segments with a positive one to complement their own data. Table 4 presents the average number of segments held by each type of client in a federated setting in our experiments.

While in our implementation we split the same training data into a "public" warm-up set and a "private" training set, in a real-life scenario, we expect the warm-up data to come from a publicly available data source (for example, a synthetic dataset or an existing publicly available dataset like the one we are using in this paper) since it will be shared among all participants in the training process.

Training with FL with metric DP Since the meaning of the privacy parameters η for metric DP depends on the chosen distance measure, the amount of noise added cannot be compared to the traditional ϵ in DP nor provide the same privacy guarantees. To choose the optimal amount of noise η for our application, we implement an adversarial attack – token embedding inversion (Qu et al. 2021b). We add noise η to each embeddings in our training set to obtain perturbed embeddings. For each perturbed embedding, we then compute the closest neighbor among all the non-perturbed embeddings in the training set using the Euclidean distance. We do this for different levels of η $(5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55)$. The goal of the attack is to recover the original non-perturbed embedding from the perturbed embedding.

Figure 6 shows how different levels of noise impact the accuracy of our attack: when $\eta = 5$, it is not possible to recover any of the original embeddings using the perturbed

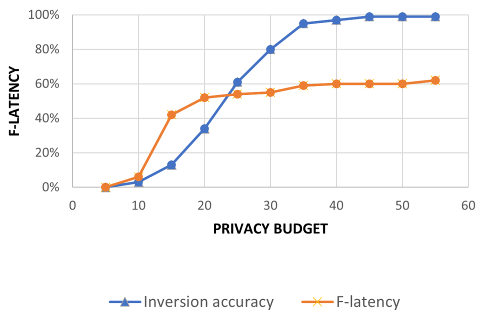


Figure 6: Impact of the privacy budget η on the F-latency score of the FL model with metric DP and on the accuracy of the embeddings inversion attack.

embeddings, whereas when $\eta = 55$ the inversion attack is successful at predicting the original embeddings 99% of the time. Hence, for our application, we estimate that $\eta = 20$ offers sufficient protection against a potential attacker since our inversion model is only able to predict the original embeddings with 34% accuracy.

Furthermore, to alleviate the high cost in utility that comes with adding noise directly to the data, we over-sample the minority class during the training of our models using Errecalde et al. (2017)'s oversampling technique. They considered that the minority class is formed not only by the complete conversation but also by portions of the full conversation at different time steps. Therefore, to account for the sequential nature of the eSPD problem and mitigate the imbalanced nature of the data, we enrich our dataset with chunks of conversations from the minority class, in our case, the conversations with a predator. By giving our system more training examples of the beginning of a conversation with a predator, we are able to manage the utility cost of metric-DP.

Training with FL with DP-SGD We use the Opacus library to compute the privacy budget, with a user-level computation for each round of FL training. Note that the total privacy budget upon training completion may vary if the same user is resampled multiple times. In such scenarios, privacy loss for users participating multiple times in FL training can be computed using composition theorems, such as (Kairouz, Oh, and Viswanath 2015) and can further amplified by randomized check-ins (Balle et al. 2020). Although we didn't implement a moment accountant on the server to distribute the privacy budget and calculate a tighter bound for composition, our methodology in this study involves using the moment accountant on the client side to manage the privacy budget of DP-SGD. It's important to emphasize that in real-life scenarios, the likelihood of the same user being resampled multiple times is low due to the larger pool of total users available for training and the possibility for each user to opt out of training. We conduct a random grid search to select the best hyperparameters: notably, the gradient clipping level $(0.5, 1, 2, 5, 7)$, the client's learning rates $(0.01, 0.05, 0.001, 0.0001)$, the batch size $(8, 16, 32, 100)$ and the number of local epochs of training $(1, 5, 10, 15, 20, 100)$.

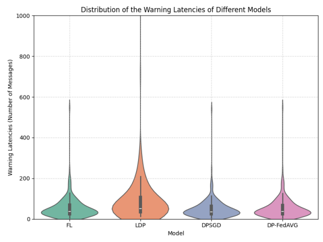


Figure 7: Distribution of the warning latencies for the full predatory conversations for each of our private models.

Training with FL with DP-FedAvg. We explore DP-FedAvg with both fixed and adaptive clipping (Hu et al. 2021) and test for different values for the update clipping level $(0.1, 0.5, 1, 2, 5, 10)$, noise level $(0.001, 0.01, 0.1, 1, 10, 100)$ and for the number of local epochs of training $(1, 5, 10, 15, 20, 100)$.

eSPD inference. All the models were evaluated using a 50-message sliding window and a skepticism level of 5, i.e. 5 of the last 10 predictions had to be positive before a warning was raised.

D Empirical Results

In this section, we explore additional quantitative results.

D.1 Warning Latency

Vogt et al. (2021) define the warning latency as the number of messages exchanged before a warning is raised. In Figure 7, we can see the distribution of the warning latencies for the four private models. We notice that while the FL, DP-SGD and DP-FedAvg models seem to have a similar distribution, for the LDP model, the number of messages it takes to get a final classification is on average higher. At the same time, the maximum of messages reach is also 3 times higher (1531) than the maximum needed for the other models (around 600).

D.2 Example of Conversation

Similarly, when looking at the difference in speed between models at the conversation level, we notice differences between the different models. For example, the following predatory conversation is flagged from the get-go by the centralized model after the following messages:

- Predator: "What's up little girl?"
- Girl: "Nuttin just chattin"
- Predator: "Kool"
- Predator: "So how old are you"
- Girl: "12/f/nva"
- Girl: "U?"

While the FL model and the FL model with DPSGD only flag it a dozen messages later, despite exchanges like "are you not going to talk to me cause i'm older than you" and only flags it after "so what you do this past summer". While

Model	**F1**	**Rec**	**Prec**	**Speed**	**F-lat**
Centralized Fine-Tuned $BERT_{base}$ (Vogt et al. 2021)	0.89	0.96	0.82	0.91	0.81
Centralized Fine-Tuned $BERT_{mobile}$ (Vogt et al. 2021)	0.80	0.95	0.69	0.72	0.58
Our FL Fine-Tuned $BERT_{mobile}$	0.79	0.96	0.68	0.73	0.58

Table 5: Evaluation results for the eSPD task - Baselines and Comparison with the literature

the DP-FedAvg model flags it on the first message and the FL model with metric DP only raises a warning much later. This is consistent with the distribution of the warning latencies observed in Figure 7.

D.3 False Positive Evaluation

In this paper, we stated that a 1% FPR strikes a good balance between ensuring system efficiency and avoiding unintended discrimination. In Figure 8, we can see how different false positive rates impact the F1-score and speed of our federated learning model.

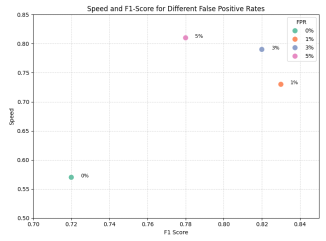


Figure 8: Change in Utility and Speed of Detection for different False Positive Rates

Models with higher FPR are faster but also lose in utility. Depending on the area where the model is deployed and the quality of the classifier, different thresholds might be more appropriate.

E Comparison with the literature

While Vogt et al. (2021) report their best results with a fine-tuned $BERT_{BASE}$ model, we do not use this model to be able to scale our experiments as fine-tuning large language models with a large number of clients in a federated setting is still an open-research question (Hilmkil et al. 2021).

However, for comparison purpose, we fine-tune a $BERT_{mobile}$ model in a federated setting and show that we can achieve similar performance with an F-latency score of 58% than the centralized setting presented by Vogt et al. (2021) as seen in Table 5.

For the rest of our experiments however, we leverage logistic regression to be able to scale to a large number of clients.