Leveraging Cross-Silo Federated Learning in Process Mining [Short Paper]

Dipanwita Thakur[1], Antonella Guzzo[1], and Giancarlo Fortino

DIMES, University of Calabria, Italy
dipanwita.thakur@unical.it

Abstract. Process mining provides insights from event logs but sharing data across organizations is often restricted by privacy and legal concerns. We propose a cross-silo federated learning (FL) framework that combines ProM-based local preprocessing with RNN-based prediction, where model updates are aggregated via FedAvg without exposing raw logs. Experiments show accuracy close to centralized training while maintaining strict data isolation, establishing a scalable foundation for privacy-preserving, collaborative process intelligence.

Key words: Process Mining, Federated Learning, Cross-silo, Privacy Preservation

1 Introduction

Process Mining (PM) combines data mining and business process management to support tasks such as discovery, conformance checking, and augmentation [1]. Recent advances in machine learning (ML) have enhanced PM with capabilities like anomaly detection, log lifting, clustering, and real-time analysis [2], but raise challenges of privacy, scalability, and collaboration. Predictive process monitoring increasingly relies on sensitive, distributed data [3]. Federated Learning (FL) addresses these concerns by enabling cross-silo model training without sharing raw logs. Our framework integrates ProM-based preprocessing with deep sequence learning to support accurate, privacy-preserving, and collaborative process prediction.

This paper makes the following key contributions:

- We propose a cross-silo federated learning framework for predictive process monitoring that preserves event log privacy while enabling collaborative training.
- Our modular pipeline combines ProM-based local preprocessing with deep sequence encoders to ensure semantic consistency across heterogeneous silos.
- We evaluate the approach on synthetic and real-world logs, showing accuracy close to centralized baselines while preserving privacy and analyzing communication–performance trade-offs.

2 Related Work

PM links data science and business process management, supporting discovery, conformance, and enhancement [4]. Predictive monitoring extends this to forecasting outcomes such as next activity [5], remaining time [6], or compliance. While ML and deep learning improve accuracy [7,8], they typically rely on centralized data, limiting applicability in privacy-sensitive, multi-organizational settings. The PI Engine [9] addresses this gap by integrating ML and AI into PM for proactive analysis. FL [10] enables decentralized training by sharing model updates instead of raw data, with variants like FedAvg, FedProx, and FedOpt tackling heterogeneity. Though successful in healthcare and finance [11], its integration into PM remains limited. Privacy-preserving approaches such as log anonymization [12] and secure multiparty computation [13] offer partial solutions but lack scalability. Recent studies in federated PM [14, 15] highlight its potential but assume uniform logs and overlook preprocessing challenges. Our work advances this line by combining ProM-based preprocessing with GRU sequence modeling in a cross-silo FL framework. This ensures semantic alignment, privacy, and scalability, achieving predictive accuracy close to centralized baselines while addressing communication–performance trade-offs.

3 Methodology

This study proposes a cross-silo FL framework for predictive process monitoring, where organizations collaboratively train a shared deep learning model without exchanging raw event logs. Each silo uses ProM for local preprocessing, converting raw logs into structured, abstracted traces suitable for deep learning. A central server then aggregates locally trained models via FedAvg while ensuring data privacy.

As shown in Figure 1, ProM extracts case IDs, timestamps, and activities, applies noise filtering, and generates structured logs. Trace abstraction ensures semantic consistency, and abstracted traces are encoded into embeddings and processed by a GRU-based RNN locally. Only model weights are shared with the server, which aggregates them to build a global predictive model without exposing raw data.

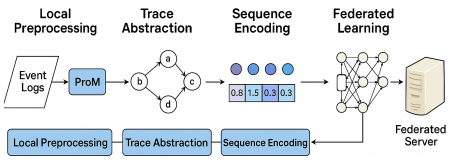


Fig. 1: Modular Pipeline with ProM-based Local Preprocessing

Each client $k \in \{1, \ldots, K\}$ holds an event log $\mathcal{L}_k = \{\sigma_i^{(k)}\}$, where each trace $\sigma_i^{(k)} = \langle e_1, e_2, \ldots, e_n \rangle$ consists of sequential events. Each event e_j includes attributes such as $e_j = \langle \texttt{activity}, \texttt{timestamp}, \texttt{resource}, \texttt{lifecycle}, \ldots \rangle$. Using ProM, each silo performs: **Filtering:** Removing infrequent activities or noise, **Abstraction:** Trans-

forming raw logs into sequence representations, and **Encoding:** Mapping events into feature vectors via one-hot encoding, positional embedding, or temporal features. Let $x_i^{(k)}$ denote the encoded input of trace $\sigma_i^{(k)}$, and $y_i^{(k)}$ the label (e.g., next activity or timestamp). Each silo implements a lightweight neural network model, specifically an RNN-based Event Predictor, to learn next-event prediction from its local sequences. The model architecture includes an embedding layer to represent activities in a dense vector space, a GRU (Gated Recurrent Unit) layer for capturing sequential patterns and a fully connected layer with a softmax output over the activity vocabulary. Each client trains its model using cross-entropy loss and Adam optimizer, without sharing raw data. After local training, each client sends the model's parameter weights to the central server. Each silo trains a local model $f(\cdot; \mathbf{w})$ using local data. The global federated learning objective is $\min_{\mathbf{w}} \quad F(\mathbf{w}) := \sum_{k=1}^{K} \frac{N_k}{N} F_k(\mathbf{w})$, where N_k is the number of samples at silo k, $N = \sum_{k=1}^{K} N_k$ is the total number of samples, $F_k(\mathbf{w}) = \frac{1}{N_k} \sum_{i=1}^{N_k} \ell(f(x_i^{(k)}; \mathbf{w}), y_i^{(k)})$ is the local empirical loss, and $\ell(\cdot, \cdot)$ is a supervised loss function (e.g., cross-entropy). Each global training round t includes the steps: (1) The server sends the current model $\mathbf{w}^{(t)}$ to a subset of clients $\mathcal{S}_t$. (2) Each client $k \in \mathcal{S}_t$ updates the model locally using $\mathbf{w}_k^{(t+1)} = \mathbf{w}^{(t)} - \eta \cdot \nabla F_k(\mathbf{w}^{(t)})$ and finally (3) the server aggregates the updated weights using $\mathbf{w}^{(t+1)} = \sum_{k \in \mathcal{S}_t} \frac{N_k}{\sum_{j \in \mathcal{S}_t} N_j} \cdot \mathbf{w}_k^{(t+1)}$. The communication cost per round is $\mathcal{C}^{(t)} = \sum_{k \in \mathcal{S}_t} \Big(\text{size}(\Delta \mathbf{w}_k^{(t)}) + \text{size}(\mathbf{w}^{(t)}) \Big)$. Assuming the model has d parameters stored as 32-bit floats, each client exchanges $\mathcal{C}_k = 2d \times 4$ bytes. Quantization or sparsification strategies can reduce $\mathcal{C}_k$.

The proposed framework supports various predictive process mining tasks such as **Next Activity Prediction:** $y_i^{(k)} = a_{j+1}$ given partial trace prefix, **Timestamp Estimation:** Predicting inter-arrival or completion times and **Remaining Time Prediction:** Estimating time-to-completion of a running case. The model $f(\cdot; \mathbf{w})$ can be instantiated as a GRU-based architecture, an RNN designed to capture temporal dependencies in sequential data. The GRU architecture efficiently manages memory and gating using the reset and update gates. Unlike LSTMs, GRUs do not require a separate memory cell, making them computationally simpler while still capable of modeling long-term dependencies in sequential data.

3.1 Convergence Behavior

We analyze the convergence behavior of our proposed FL framework under standard assumptions commonly adopted in distributed optimization literature. Let the global objective be defined as $F(\mathbf{w}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\mathbf{w})$, where F_k denotes the local objective at silo k, and $\mathbf{w}$ is the shared model parameter. **Assumptions: (A1)** L**-Smoothness:** Each F_k is L-smooth; i.e., $\|\nabla F_k(\mathbf{w}_1) - \nabla F_k(\mathbf{w}_2)\| \leq L \|\mathbf{w}_1 - \mathbf{w}_2\|$ for all $\mathbf{w}_1, \mathbf{w}_2$. **(A2) Unbiased Stochastic Gradients:** $\mathbb{E}[\nabla f_k(\mathbf{w})] = \nabla F_k(\mathbf{w})$. **(A3) Bounded Variance:** $\mathbb{E}\|\nabla f_k(\mathbf{w}) - \nabla F_k(\mathbf{w})\|^2 \leq \sigma^2$. Under the above assumptions, and using a constant learning

rate $\eta = \mathcal{O}(1/\sqrt{T})$, our method ensures that after T communication rounds across K silos, the expected gradient norm of the global loss function satisfies $\min_{t=1,\ldots,T} \mathbb{E}\|\nabla F(\mathbf{w}^{(t)})\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{KT}}\right)$. This bound implies sublinear convergence, where both increased communication rounds T and larger silo participation K enhance learning stability. Notably, this is achieved without requiring centralized access to sensitive event logs, preserving data sovereignty while ensuring robust global model convergence.

Proof We follow a standard stochastic federated optimization argument, similar to the analysis in [10, 16]. Let $\mathbf{w}^{(t)}$ be the global model at round t, updated via Federated Averaging as $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_t \sum_{k \in S_t} \frac{1}{|S_t|} \nabla F_k(\mathbf{w}^{(t)})$, where $S_t \subseteq \{1, \ldots, K\}$ is the subset of selected silos at round t, and η_t is the learning rate. Under the assumption that each F_k is L-smooth and the gradient noise is bounded, we can use the standard descent lemma to bound the expected decrease in global loss between rounds, $\mathbb{E}[F(\mathbf{w}^{(t+1)})] \leq \mathbb{E}[F(\mathbf{w}^{(t)})] - \frac{\eta_t}{2}\mathbb{E}\|\nabla F(\mathbf{w}^{(t)})\|^2 + \frac{L\eta_t^2\sigma^2}{2K}$, where σ^2 denotes the bounded variance of the stochastic gradients. Summing the inequality over $t = 1$ to T and choosing a constant learning rate $\eta_t = \eta = \mathcal{O}(1/\sqrt{T})$, we obtain $\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla F(\mathbf{w}^{(t)})\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{KT}}\right)$. This result implies that the expected gradient norm decreases over time, and the convergence improves with the number of participating silos K and communication rounds T. Importantly, this convergence is achieved without sharing raw event logs, thus preserving cross-silo data privacy.

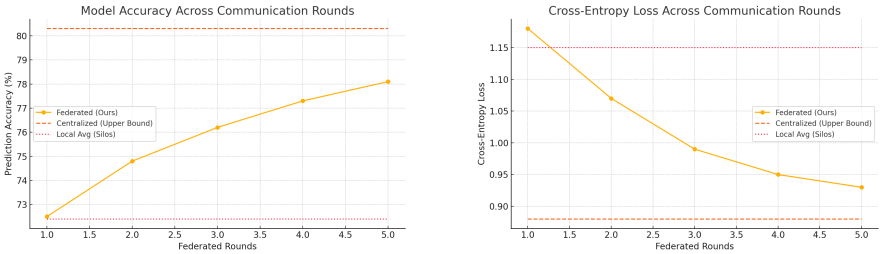
4 Performance Evaluation

We evaluate our cross-silo FL framework using three silos: one with the BPI Challenge 2019 dataset [17] and two with synthetic variants. Each silo trains a GRU-based RNN (embedding size 128, two GRU layers with 128 units and dropout 0.3) optimized with Adam and cross-entropy loss. Local models run for 5 epochs, and FedAvg aggregates updates over 5 rounds.

Table 1: Performance Comparison of Federated and Baseline Models

Model	**Accuracy (%)**	**Cross-Entropy Loss**	**Data Shared**
Local (Silo 1)	72.4	1.15	No
Local (Silo 2)	74.1	1.08	No
Local (Silo 3)	70.9	1.22	No
Centralized (Upper Bound)	80.3	0.88	Full Logs
Federated (Ours)	**78.1**	**0.93**	Model Weights Only

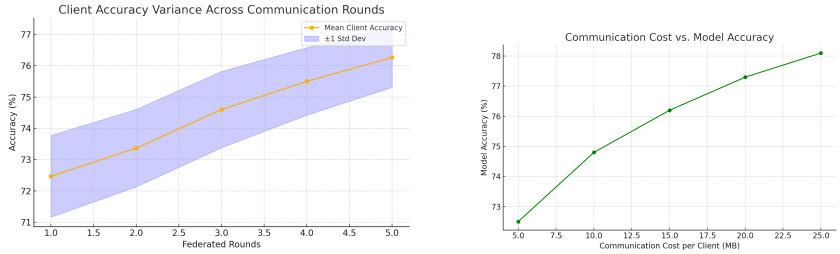
Our federated model achieves 78.1% accuracy, closely approaching the centralized baseline (80.%) while preserving privacy. Figures 2a and 2b show accuracy and loss convergence across rounds. Client variance (Figure 3a) narrows over



(a) Model accuracy over federated communication rounds.

(b) Cross-entropy loss across federated communication rounds.

Fig. 2: Accuracy and Loss



(a) Client accuracy variance across federated communication rounds

(b) Communication Cost vs. Model Accuracy in Cross-Silo Federated PM

Fig. 3: Accuracy vs. Communication rounds and cost

time, indicating fairer performance across silos. Finally, Figure 3b highlights the communication–accuracy trade-off, with diminishing returns beyond 15–20 MB per client, confirming the practicality of our framework in resource-constrained, privacy-sensitive settings.

5 Conclusion

We propose a cross-silo federated learning framework for predictive process monitoring that preserves privacy while enabling collaborative model training. Local preprocessing with trace abstraction ensures data remain within organizations, while deep encoders capture temporal and contextual dependencies. Our approach achieves accuracy close to centralized baselines and highlights communication–performance trade-offs, enabling privacy-preserving process intelligence in domains such as healthcare, finance, and supply chains. Future work will explore asynchronous FL, personalization, and differential privacy to enhance robustness and compliance.

Acknowledgment

This work contributes to the basic research activities of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

1. Deloitte: Global process mining survey. Technical report, Technical report, Deloitte (2025)
2. Tavares, G.M., Barbon Junior, S., Damiani, E., Ceravolo, P.: Selecting optimal trace clustering pipelines with meta-learning. In Xavier-Junior, J.C., Rios, R.A., eds.: Intelligent Systems, Springer International Publishing (2022) 150–164
3. Hanga, K.M., Kovalchuk, Y., Gaber, M.M.: A graph-based approach to interpreting recurrent neural networks in process mining. IEEE Access **8** (2020) 172923–172938
4. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer (2016)
5. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with lstm neural networks. CAiSE (2017)
6. Verenich, I.e.a.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. Information Systems (2019)
7. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2) (March 2019)
8. Evermann, J.e.a.: Predicting process behaviour using deep learning. In: BPM. (2017)
9. Veit, F., Geyer-Klingeberg, J., Madrzak, J., Haug, M., Thomson, J.: The proactive insights engine: Process mining meets machine learning and artificial intelligence. In: International Conference on Business Process Management. (2017)
10. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, PMLR (2017) 1273–1282
11. Rieke, N., Hancox, J., et al.: The future of digital health with federated learning. npj Digital Medicine **3**(1) (Sep 2020) 119
12. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining. Business & Information Systems Engineering **61**(5) (Oct 2019) 595–614
13. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Secure multi-party computation for inter-organizational process mining. In Nurcan, S., Reinhartz-Berger, I., Soffer, P., Zdravkovic, J., eds.: Enterprise, Business-Process and Information Systems Modeling. (2020) 166–181
14. van der Aalst, W.M.: Federated process mining: Exploiting event data across organizational boundaries. In: 2021 IEEE International Conference on Smart Data Services (SMDS). (2021) 1–7
15. Khan, A., Ghose, A., Dam, H.: Cross-silo process mining with federated learning. In Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H.y., eds.: Service-Oriented Computing, Cham, Springer International Publishing (2021) 612–626
16. Stich, S.U.: Local sgd converges fast and communicates little. In: International Conference on Learning Representations. (2019)
17. van Dongen, B.: Bpi challenge 2019 (version 1) [data set]. In: 4TU.Centre for Research Data. (2019)