Beyond Dashboards: *PMAgent* for RAG-Driven Natural Language Process Mining

Francesco Scala[1 [0009−0007−5224−0910]] and Luigi Pontieri[1 [0000−0003−4513−0362]]

Institute of High Performance Computing and Networking (ICAR-CNR), Via P. Bucci, 87036 Rende (CS), Italy
{francesco.scala,luigi.pontieri}@icar.cnr.it

Abstract. Modern organizations increasingly rely on Process Mining (PM) to extract actionable insights and monitor system behaviors from event logs. However, executing advanced analytical tasks, such as predictive monitoring, performance evaluation, and what-if analysis, typically demands profound technical expertise and complex, domain-specific query languages (e.g., PQL, LTL, Declare). This complexity severely limits the accessibility of PM tools for non-technical stakeholders. While Large Language Models (LLMs) offer a promising avenue for intuitive data interaction, applying them directly to raw process logs often leads to hallucinations and struggles with complex temporal dependencies. In this work, we propose *PMAgent*, a novel framework that integrates LLMs with Retrieval-Augmented Generation (RAG) to perform comprehensive and autonomous process mining analysis. *PMAgent* enables users to interrogate complex event logs entirely through natural language, smoothly translating conversational queries into advanced PM tasks like prediction and scenario evaluation. By grounding the LLM's reasoning in retrieved, fact-based process contexts via RAG, our approach ensures high accuracy and reliability. Preliminary evaluations demonstrate the feasibility of our approach, indicating that *PMAgent* has the potential to make complex process analytics more intuitive and accessible.

Keywords: Process Mining · Predictive Process Monitoring · Business Process Intelligence · Large Language Models · Retrieval-Augmented Generation.

1 Introduction

Process Mining and Accessibility. Modern organizations increasingly rely on Process Mining (PM) [1] to extract actionable insights and monitor system behaviors from event logs. PM focuses on discovering, monitoring, and improving real-world processes by analyzing sequences of events, providing valuable visibility into workflows, bottlenecks, and compliance deviations. However, as the field evolves from descriptive analysis to more advanced tasks, such as predictive monitoring, performance evaluation, and what-if scenario analysis, a significant accessibility barrier emerges. Executing these advanced analytical tasks typically demands profound technical expertise and proficiency in complex, domain-specific query languages (e.g., PQL, LTL, Declare). This technical complexity

restricts the use of sophisticated PM tools to highly specialized data scientists, leaving business users and domain stakeholders unable to interact directly with the process data.

The Role of Large Language Models. The recent advancements in Natural Language Processing (NLP), driven by Large Language Models (LLMs) [27], offer a paradigm shift in how users can interact with complex systems. Built on transformer architectures and pre-trained on vast amounts of data, LLMs have demonstrated remarkable capabilities in understanding intent and generating human-like text, opening the door to conversational interfaces for data analytics. In the context of PM, translating natural language queries into executable process analysis holds the potential to democratize process intelligence. By allowing users to interrogate systems using everyday language, organizations could drastically reduce the time-to-insight and foster a more data-driven decision-making culture across all hierarchical levels.

Existing Solutions and Limitations. Currently, the integration of LLMs into PM is still in its nascent stages. Most existing enterprise solutions rely on static dashboards or visual interfaces that, while intuitive, lack the flexibility for dynamic, ad-hoc exploratory queries. On the other hand, early attempts to use LLMs as direct query engines for event data struggle with the strict factual constraints required by process analytics [4]. Directly applying LLMs to raw event logs or relying solely on their parametric memory for complex analytical reasoning often leads to hallucinations, inaccuracies, and an inability to grasp intricate temporal dependencies. To mitigate these issues in other domains, Retrieval-Augmented Generation (RAG) [16] has emerged as a robust technique. RAG grounds the LLM's responses in retrieved, fact-based external knowledge. However, adapting RAG architectures to handle structured process logs for advanced tasks like predictive monitoring and scenario evaluation remains an open and challenging research area.

Contribution. To address these challenges and bridge the gap between sophisticated analytics and user accessibility, we introduce *PMAgent*, a novel framework that integrates LLMs with a RAG-driven architecture tailored for process mining. *PMAgent* is designed to operate as an autonomous conversational agent capable of translating natural language queries into advanced PM tasks, including next-step prediction, log analysis, and scenario evaluation (what-if analysis) but not only limited to. By grounding the LLM's reasoning in retrieved process context rather than raw, unguided log parsing, our framework minimizes hallucinations and ensures reliable, explainable insights. Preliminary evaluations demonstrate the viability of our approach, indicating that *PMAgent* has the potential to significantly lower the technical barrier in event log analysis, paving the way for fully conversational process intelligence.

Organization. The remainder of this paper is organized as follows: Section 2 provides an overview of related work on conversational analytics and LLM applications in PM. Section 3 details the architecture and the RAG integration of

PMAgent. Section 4 presents the experimental setup and discusses the preliminary results. Finally, Section 5 concludes the paper and outlines future research directions.

2 Related Work

Deep Learning and Event Abstraction in Process Mining. The integration of machine learning, especially deep learning, into process mining has significantly advanced predictive capabilities. Techniques like LSTMs [11] have been widely adopted for next-step prediction [23, 5, 10], demonstrating their effectiveness in capturing sequential dependencies in event logs. Beyond traditional recurrent architectures, the paradigm-shifting introduction of the Transformer model [24] has paved the way for attention-based mechanisms and LLMs to be successfully adapted to the process mining domain. A prominent challenge within this landscape involves log abstraction and low-level event tagging. For instance in[7] was proposed a data-efficient neuro-symbolic approach that combines abstract argumentation frameworks (AAFs) with machine learning sequence taggers to resolve mapping uncertainties and interpret low-level process event streams. Crucially, the flexible nature of our proposed *PMAgent* framework allows it to be seamlessly extended to support similar event abstraction and tagging tasks.

Suffix Prediction and Semantic Trace Histories. Traditional predictive process monitoring (PPM) approaches have extensively used Convolutional Neural Networks (CNNs) and attention-based models to process event logs and enhance interpretability [14, 26, 22, 8]. However, step-by-step prediction models often suffer from the propagation of prediction errors. To mitigate this limitation, recent literature has explored single-step multi-output forecasting for entire activity sequences. A notable example is LUPIN [19], which addresses activity suffix prediction by translating running process instances into rich semantic text stories according to narrative templates and processing them through a multi-output network. In our framework, we build upon this perspective by leveraging a trace-to-text conversion mechanism heavily inspired by their semantic trace histories, allowing *PMAgent* to gain deep contextual awareness of the ongoing process trace.

Large Language Models and Conversational Process Mining. Recently, the focus has shifted toward making process analytics more accessible through natural language interfaces. Early explorations of LLMs in process mining have highlighted their potential in code generation and basic log querying. However, direct application of LLMs to structured event data has proven problematic. Works such as [4, 3] systematically demonstrated that relying solely on the parametric memory of LLMs for process reasoning leads to severe hallucinations and failures in respecting strict temporal constraints. To overcome these limitations without relying on rigid, domain-specific query languages, *PMAgent* grounds the LLM's reasoning process using retrieved facts, effectively bridging the gap between conversational accessibility and factual accuracy.

Green AI and Retrieval-Augmented Generation. Deep learning models form the backbone of modern AI but are notoriously energy-intensive, especially during large-scale training phases. The resulting computational and environmental costs have spurred significant research into more sustainable approaches, establishing the field of Green AI. For example in [20] was introduced data-efficient model training frameworks that utilize adaptive subset sampling and lightweight active learning to minimize the carbon footprint of deep neural networks. While parameter-efficient fine-tuning (PEFT) strategies [12, 6, 15, 17] offer various trade-offs to reduce updated weights, they still introduce computational overhead. To achieve maximum energy efficiency, our framework entirely bypasses the training and fine-tuning loops. Instead, *PMAgent* couples a completely frozen, pre-trained LLM with RAG [16], dynamically fetching relevant process knowledge from explicit non-parametric memory at zero training-energy cost.

Agentic AI. This shift toward zero-shot contextual reasoning and runtime orchestration fits within the broader evolution toward autonomous intelligent architectures. As highlighted in [2], Agentic AI represents a transition from passive, task-specific predictive models to proactive, prompt-driven orchestration systems capable of persistent state management and execution planning. By designing our solution as a process mining agent (*PMAgent*), we position it directly within this emerging paradigm, exploiting hybrid reasoning capabilities to solve complex monitoring and predictive tasks without heavy training prerequisites.

3 Proposed Framework

In this section, we present the architecture of *PMAgent*, our conversational process mining framework. As illustrated in the conceptual architecture (Figure 1), the system is designed to seamlessly translate raw event logs into actionable insights through natural language interactions. The framework comprises three main interacting modules: Data Preprocessing (Log-to-Text Translation), the Retrieval-Augmented Generation (RAG) module, and the Large Language Model (LLM) orchestrator.

Data Preprocessing and Trace-to-Text Translation. The first fundamental step in our pipeline handles the ingestion and transformation of standard event logs. Traditional process logs, typically stored in tabular formats (e.g., CSV), contain sequences of events characterized by at least a *case ID*, a *timestamp*, and an *activity* name. However, LLMs are inherently optimized for natural language rather than raw tabular reasoning. To bridge this gap, our data preprocessing module applies a Trace-to-Text translation mechanism. The system groups the tabular records by case and reconstructs the execution history of each process instance as a continuous narrative directly into the text if available. To preserve the multidimensionality of the underlying process data, our translation module dynamically extracts and serializes additional attributes associated with

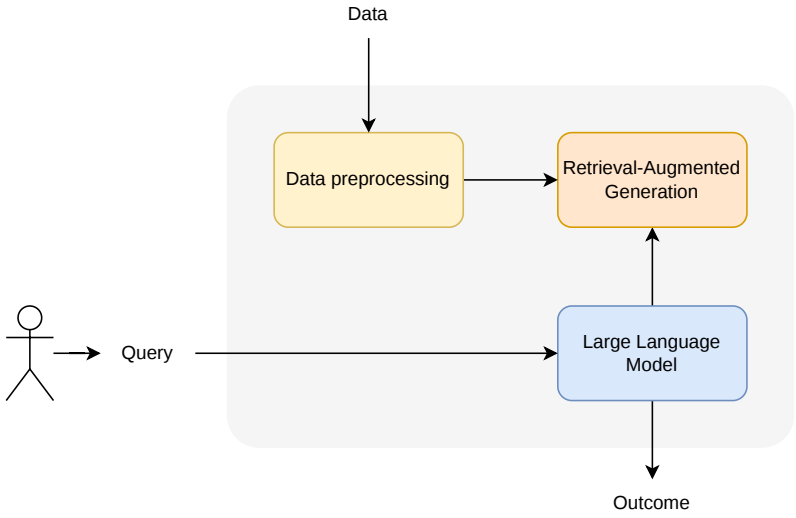

Fig. 1. Overview of the proposed *PMAgent*.

each event. As illustrated in Listing 1.1, every activity is enriched with a structured `Additional info` block, which captures domain-specific dynamics, these can vary depending on the set of the data. This serialization ensures that even heterogeneous datasets with varying schema structures are normalized into a coherent, context-rich narrative that the LLM can interpret without loss of detail, enabling more granular analytical reasoning.

This crucial transformation allows the framework to represent complex process behaviors, sequential dependencies, and contextual information in a format natively understandable by language models.

RAG Module. Once the event logs are translated into textual narratives, they are processed by the RAG module to build the framework's non-parametric memory. The generated stories are chunked into smaller, overlapping segments using a character-based text splitter to maintain context continuity. To map these textual chunks into a high-dimensional vector space, we utilize state-of-the-art text embedding models. The resulting embeddings are stored within a high-performance vector database (FAISS). This architecture enables rapid similarity-based retrieval: when a user submits a query, the system dynamically fetches only the most relevant historical traces, bottleneck descriptions, or execution variants that match the semantic intent of the question.

Listing 1.1. Example of Trace-to-Text translation with dynamic metadata serialization.

```
The case 'Case 1793' started its journey, at 2011-07-26
    08:52:23,
the activity 'Assign seriousness' was performed
  (Additional info ->
   supportsection: Value 1,     seriousness2: Value 1,
   product:        Value 4,     customer:     Value 306,
   workgroup:      Value 1,     responsiblesection: Value
        1,
   servicetype:    Value 1,     servicelevel: Value 2,
   resource:       Value 9).

then at 2011-07-26 13:32:51,
the activity 'Resolve ticket' was performed
  (Additional info ->
   supportsection: Value 1,     seriousness2: Value 1,
   product:        Value 4,     customer:     Value 306,
   workgroup:      Value 1,     responsiblesection: Value
        1,
   servicetype:    Value 1,     servicelevel: Value 2,
   resource:       Value 13).

then at 2011-09-10 09:00:40,
the activity 'Closed' was performed.
```

LLM Orchestrator and Query Execution. The core reasoning capability of *PMAgent* is powered by a pre-trained LLM, orchestrated via the LangChain[1] framework. To guarantee factual alignment and enforce strict analytical behavior, the reasoning process of *PMAgent* is governed by a carefully crafted prompt template. This system prompt instructs the underlying LLM to operate under rigorous logical boundaries, minimizing the risk of speculative responses. The exact prompt architecture implemented within the framework is provided in Listing 1.2.

The retrieved context from the FAISS database is strictly provided to the model alongside the query. To prevent hallucinations and ensure high factual accuracy, the prompt enforces strict operational rules: the LLM must base probabilities and estimations solely on the retrieved historical frequencies, perform what-if scenario evaluations by comparing alternatives to baseline metrics, and explicitly declare if the provided data is insufficient. Finally, the framework imposes a structured output generation (e.g., JSON format), ensuring that the resulting analytical insights—such as case status, bottleneck identification, and expected

[1] https://www.langchain.com/

Listing 1.2. System prompt template utilized by *PMAgent* to enforce structured analysis and prevent hallucinations.

```
<start_of_turn>user
You are a Senior Process Mining Analyst and Data
    Scientist. Your goal is to analyze event logs,
    process models, transition probabilities, and
    performance metrics provided in the context to answer
     complex questions about process execution.

Use strictly and exclusively the following pieces of
    context to answer the question at the end. The
    context may include historical execution data,
    bottleneck analysis, variant frequencies, and
    transition matrices.

Please adhere strictly to the following rules:
1. No Hallucinations: If the provided context does not
    contain sufficient data to calculate a probability,
    make a prediction, or answer the question, do not
    invent data. Simply state: "I cannot find the exact
    answer in the provided data."
2. Predictions & Probabilities: When asked about future
    outcomes, base your estimations strictly on the
    historical frequencies.
3. What-If Analysis: When performing "what-if" scenarios,
     compare the hypothetical scenario against the
    baseline metrics.
4. Case Status & Estimation: Identify its current state
    based on the logs, trace its most likely next steps.
5. Structure & Clarity: Keep your answer highly
    analytical, objective, and well-structured.
6. Format the code in json format were there are keys
    like "answer", "probability", "what_if_analysis", "
    case_status", "estimation", etc. depending on the
    question asked.

Context:
{context}

Question:
{question}
<end_of_turn>
<start_of_turn>model
```

completion times are consistently formatted and ready for downstream integration.

4 Experimental Evaluation

Settings and assessment criteria. To assess the feasibility of the proposed framework, we conducted a preliminary evaluation using a standard real-world event log, specifically the Helpdesk dataset. The log was preprocessed using our Trace-to-Text translation module to generate semantic trace histories. For the RAG architecture, text chunks were mapped into a vector space using the `BAAI/bge-small-en-v1.5` embedding model. The core LLM orchestrator was powered by `Google Gemma-3`. To evaluate the system's factual adherence, we set the model's temperature to 0.0, minimizing stochastic variability during text generation.

Qualitative Case Study: Question Answering & Prediction. In our first experiment, we evaluated the conversational and analytical capabilities of *PMAgent* on ad-hoc process queries. Traditional PPM models require specialized architectures—often distinct models for different tasks—to predict the next step, estimate remaining time, or evaluate process variants. Instead, we tested the flexibility of *PMAgent* by providing it with three distinct classes of natural language prompts, aiming to assess its zero-shot reasoning capabilities directly against the semantic trace histories retrieved from the FAISS database. The results of these interactions are reported in Table 1.

Specifically, we designed the queries to trigger three core behaviors enforced by our system prompt:

- **Next-Step Prediction:** We evaluated the model's ability to infer the subsequent activity based on historical sequences. As shown in the first query, *PMAgent* correctly identified 'Resolve ticket' as the logical next step. Crucially, when asked for a specific numeric probability that could not be confidently calculated from the retrieved chunks, the model strictly adhered to the "No Hallucinations" directive, refusing to fabricate a percentage;
- **Scenario Evaluation (What-if):** We tested the framework's capacity to perform dynamic hypothetical reasoning. By injecting a synthetic delay into the 'Take in charge ticket' activity, the LLM successfully identified the downstream propagation of the delay, mathematically formulating the expected impact ($X + Y + 6$ days) without attempting to invent absolute baseline durations that were absent from the prompt;
- **Adversarial Factual Checking:** We submitted a query deliberately requesting out-of-domain operational data (e.g., employee salaries and software vendors). The system successfully rejected the query, falling back to a raw text explanation indicating the lack of relevant context.

These preliminary interactions demonstrate that grounding a frozen LLM with retrieval-augmented process contexts allows for accurate, multi-task process

Table 1. Qualitative evaluation of *PMAgent* responses to natural language queries on the Helpdesk dataset.

User Query	***PMAgent* Output**

Next-Step Prediction: "Consider a case that has just completed the activity 'Assign seriousness'. Based on the historical traces provided in the context, what is the most likely next activity, and what is the probability of this case eventually reaching the 'Closed' status?"	**JSON Format:**
	`{`
	`"answer": "The most likely next activity after 'Assign seriousness' is 'Resolve ticket'.",`
	`"probability": "I cannot find the exact answer in the provided data.",`
	`"what_if_analysis": "Based on the provided data, all cases after 'Assign seriousness' immediately proceed to 'Resolve ticket'...",`
	`"case_status": "Currently in the 'Assign seriousness' state.",`
	`"estimation": "The case will likely transition to 'Resolve ticket' and then to 'Closed' based on the observed sequence..."`
	`}`

Scenario Evaluation (What-if): "What if the activity 'Take in charge ticket' is systematically delayed by 3 days compared to the baseline metrics present in the logs? Clearly explain the expected impact on the overall throughput time..."	**JSON Format:**
	`{`
	`"what_if_analysis": "If 'Take in charge ticket' is consistently delayed by 3 days, the overall throughput time would likely increase. The delay would propagate through the process...",`
	`"estimation": "Estimating the new average completion time requires understanding the typical duration of each activity. Let's assume 'Take in charge ticket' is X days and 'Wait' is Y days. With a 3-day delay, the new average completion time would be approximately X + 3 + Y + 3 = X + Y + 6 days."`
	`}`

Adversarial Check: "What is the average hourly salary of the support agents resolving the tickets, and which external software vendor is used to manage the Helpdesk system?"	**Raw Text Fallback:** "I cannot find the exact answer in the provided data. The context does not include information about hourly salaries or the external software vendor used for the Helpdesk system."

analysis while successfully mitigating the hallucination risks typically associated with parametric memory alone.

5 Conclusions

In this work, we introduced *PMAgent*, a novel conversational framework that leverages RAG to democratize access to advanced Process Mining analytics. By implementing a Trace-to-Text translation mechanism, *PMAgent* bridges the gap between structured tabular event data and the natural language processing capabilities of Large Language Models. In alignment with Green AI principles, the architecture completely bypasses energy-intensive training and fine-tuning loops, relying instead on zero-shot contextual reasoning over an external, non-parametric vector memory (FAISS) powered by `Google Gemma-3`.

Our preliminary evaluation on the Helpdesk dataset confirms the feasibility and robustness of the approach. *PMAgent* successfully executed complex analytical tasks, including next-step prediction and what-if scenario evaluations, while generating structured, ready-to-integrate JSON outputs. Crucially, the strict operational boundaries enforced by our system prompt effectively mitigated hallucination risks, forcing the model to gracefully decline out-of-domain adversarial queries and declare data insufficiency whenever historical frequencies were missing.

Several promising directions outline our future research. First, we plan to expand the empirical evaluation by conducting a more rigorous and formal validation across multiple diverse real-world datasets to thoroughly assess the framework's generalization capabilities and scalability. Second, we aim to refine the underlying reasoning engine to enable *PMAgent* to formally compute and extract mathematically consistent event probabilities, thereby providing reliable quantitative metrics even when historical frequencies are sparsely distributed in the retrieved context. Furthermore, in alignment with Green AI principles, we plan to systematically monitor the computational costs and energy footprint of the proposed architecture using FLOPpy [21], a hardware-agnostic tracking library. Moreover, we plan to extend the framework to smart building and environmental monitoring contexts, enabling conversational analytics for digital twins and indoor climate forecasting [13]. Finally, we aim to validate the robustness of *PMAgent* within highly sensitive medical and healthcare domains [25], and explore its application to large-scale social processes to assist in unveiling complex user dynamics in evolving social debates [18, 9].

Data and Code Availability. The implementation of *PMAgent*, including the Trace-to-Text translation module and the evaluation scripts used in this study, is publicly available to ensure reproducibility. The source code, documentation, and instructions for running the framework on custom event logs can be accessed at: **AVAILABLE UPON ACCEPTANCE**

Acknowledgments. This work has been partially supported by the SINTESI research project, funded under the National Programme for Research, Innovation and Competitiveness (PN RIC 2021–2027), within the FAIR (Future Artificial Intelligence Research)

Extended Partnership, funded by the European Union and the Italian Ministry of University and Research (MUR). Project code: PE00000013, CUP: H97G22000210007.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. van der Aalst, W.: Data Science in Action, pp. 3–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4, https://doi.org/10.1007/978-3-662-49851-4
2. Abou Ali, M., Dornaika, F., Charafeddine, J.: Agentic ai: a comprehensive survey of architectures, applications, and future directions. Artificial Intelligence Review **59**(1), 11 (2025). https://doi.org/10.1007/s10462-025-11422-4, https://doi.org/10.1007/s10462-025-11422-4
3. Berti, A.: Pm4py.llm: a comprehensive module for implementing pm on llms (2024), https://arxiv.org/abs/2404.06035
4. Berti, A., Kourani, H.: Diagnosing llm hallucinations in process mining tasks: a taxonomy and a benchmark. In: Mendling, J., Leemans, S., van Dongen, B.F., Reijers, H. (eds.) Mining a Scientist's Process: Essays Dedicated to Wil van der Aalst on the Occasion of His 60th Birthday, pp. 631–647. Springer Nature Switzerland, Cham (2026), https://doi.org/10.1007/978-3-032-17618-9_41
5. Dentamaro, V., Impedovo, D., Pirlo, G., Semeraro, G.: Next activity prediction and elapsed time prediction on process dataset. In: Falchi, F., Giannotti, F., Monreale, A., Boldrini, C., Rinzivillo, S., Colantonio, S. (eds.) Proceedings of the Italia Intelligenza Artificiale - Thematic Workshops co-located with the 3rd CINI National Lab AIIS Conference on Artificial Intelligence (Ital IA 2023), Pisa, Italy, May 29-30, 2023. CEUR Workshop Proceedings, vol. 3486, pp. 605–609. CEUR-WS.org (2023), https://ceur-ws.org/Vol-3486/19.pdf
6. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: efficient finetuning of quantized llms. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
7. Fazzinga, B., Flesca, S., Furfaro, F., Pontieri, L., Scala, F.: Combining abstract argumentation and machine learning for efficiently analyzing low-level process event streams. Complex & Intelligent Systems (2026). https://doi.org/10.1007/s40747-026-02340-1, https://doi.org/10.1007/s40747-026-02340-1
8. Folino, F., Guarascio, M., Liguori, A., Manco, G., Pontieri, L., Ritacco, E.: Exploiting temporal convolution for activity prediction in process analytics. In: Koprinska, I., Kamp, M., Appice, A., Loglisci, C., Antonie, L., Zimmermann, A., Guidotti, R., Özgöbek, Ö., Ribeiro, R.P., Gavaldà, R., Gama, J., Adilova, L., Krishnamurthy, Y., Ferreira, P.M., Malerba, D., Medeiros, I., Ceci, M., Manco, G., Masciari, E., Ras, Z.W., Christen, P., Ntoutsi, E., Schubert, E., Zimek, A., Monreale, A., Biecek, P., Rinzivillo, S., Kille, B., Lommatzsch, A., Gulla, J.A. (eds.) ECML PKDD 2020 Workshops - Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020): SoGood 2020, PDFL 2020, MLCS 2020, NFMCP 2020, DINA 2020, EDML 2020, XKDD 2020 and INRA 2020, Ghent, Belgium, September 14-18, 2020, Proceedings. Communications in Computer and Information Science,

vol. 1323, pp. 263–275. Springer (2020). https://doi.org/10.1007/978-3-030-65965-3_17, https://doi.org/10.1007/978-3-030-65965-3_17
9. Gullo, F., Mandaglio, D., Tagarelli, A.: Neural discovery of balance-aware polarized communities. Machine Learning **113**(9), 6611–6644 (sep 2024). https://doi.org/10.1007/s10994-024-06581-4, https://doi.org/10.1007/s10994-024-06581-4
10. Gunnarsson, Bjorn Rafn and vanden Broucke, Seppe and Weerdt, Jochen De: A direct data aware LSTM neural network architecture for complete remaining trace and runtime prediction. IEEE TRANSACTIONS ON SERVICES COMPUTING **16**(4), 2330–2342 (2023), {http://doi.org/10.1109/tsc.2023.3245726}
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735
12. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9
13. Islam, M.B., Guerrieri, A., Gravina, R., Pontieri, L., Rizzo, L., Scala, F., Vinci, A., Fortino, G.: itempdt: Ai-powered digital twin for forecasting indoor temperature in smart buildings. In: 2025 IEEE Conference on Pervasive and Intelligent Computing (PICom). pp. 121–128 (2025). https://doi.org/10.1109/PICom68402.2025.00022
14. Kamala, B., Latha, B.: Process mining and deep neural network approach for the prediction of business process outcome. In: 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT). pp. 1–4 (2022). https://doi.org/10.1109/IC3IOT53935.2022.9767941
15. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3045–3059. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.243, https://aclanthology.org/2021.emnlp-main.243/
16. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
17. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.353, https://aclanthology.org/2021.acl-long.353/
18. Martirano, L., La Cava, L., Tagarelli, A.: Unveiling user dynamics in the evolving social debate on climate crisis during the conferences of the parties. Pervasive and Mobile Computing **112**, 102077 (2025)
19. Pasquadibisceglie, V., Appice, A., Malerba, D.: Lupin: A llm approach for activity suffix prediction in business process event logs. In: 2024 6th International Conference on Process Mining (ICPM). pp. 1–8 (2024). https://doi.org/10.1109/ICPM63005.2024.10680620

20. Scala, F., Flesca, S., Pontieri, L.: An efficient model training framework for green ai. Machine Learning **114**(12), 275 (11 2025). https://doi.org/10.1007/s10994-025-06907-w, https://doi.org/10.1007/s10994-025-06907-w
21. Scala, F., Mandarino, F., Martirano, L., Pontieri, L.: Floppy: A hardware-agnostic python library to monitor the computational cost of machine and deep learning algorithms (2026), https://github.com/Franco7Scala/FLOPpy
22. Sindhgatta, R., Moreira, C., Ouyang, C., Barros, A.: Exploring interpretable predictive models for business processes. In: Business Process Management: 18th International Conference, BPM 2020, Seville, Spain, September 13–18, 2020, Proceedings. p. 257–272. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-58666-9_15, https://doi.org/10.1007/978-3-030-58666-9_15
23. Tax, N., Verenich, I., {La Rosa}, M., Dumas, M.: Predictive business process monitoring with lstm neural networks. In: Pohl, K., Dubois, E. (eds.) Advanced Information Systems Engineering : 29th International Conference, CAiSE 2017, Essen Germany, June 12-16, 2017. Proceedings. pp. 477–492. LNCS, Springer, Germany (2017). https://doi.org/10.1007/978-3-319-59536-8_30, http://caise2017.paluno.de/welcome/, 29th International Conference on Advanced Information Systems Engineering, CAiSE 2017, CAiSE 2017 ; Conference date: 12-06-2017 Through 16-06-2017
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
25. Vizza, P., Tradigo, G., Giancotti, R., Scala, F., Guzzi, P.H., Irace, C., Flesca, S., Veltri, P.: On the identification of pois in glucosimeter data. In: 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI). pp. 542–544 (2022). https://doi.org/10.1109/ICHI54592.2022.00104
26. Weytjens, H., De Weerdt, J.: Process outcome prediction: Cnn vs. lstm (with attention). In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) Business Process Management Workshops. pp. 321–333. Springer International Publishing, Cham (2020)
27. Zhao, W.X., Zhou, K., Li, J., Tang, T., Dong, Z., Hou, Y., Zhang, B., Min, Y., Zhang, J., Liu, P., Wang, X., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Hu, Y., Nie, J.Y., Wen, J.R.: A survey of large language models. Frontiers of Computer Science **20**(12), 2012627 (2026). https://doi.org/10.1007/s11704-026-60308-3, https://doi.org/10.1007/s11704-026-60308-3