Exploiting and Predicting System-Wide Behaviors for Predictive Process Monitoring

Xiaomeng He[1]

[1]Research Center for Information Systems Engineering (LIRIS), KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium

Abstract

Predictive Process Monitoring (PPM) focuses on forecasting the future progression of ongoing business process instances (cases). Despite growing interest in alternative viewpoints, many PPM approaches still tend to adopt a case-centric perspective inherited from traditional process mining, using an incomplete trace of a single case to generate case-specific predictions. However, this perspective overlooks system-level dynamics in both the input and output dimensions. Specifically, the exclusive reliance on intra-case data as inputs limits predictive accuracy by neglecting valuable contextual information from other concurrently running cases. Similarly, the field's predominant focus on case-level predictions has constrained the exploration of system-level predictions, such as system-wide event stream modeling or system load forecasting, which may enable applications including anomaly detection and resource allocation. This proposal aims to address these limitations by enhancing case-level prediction through the automated extraction of inter-case dynamics from event logs, and extending PPM to support system-level prediction tasks. Ultimately, we propose to integrate these two research directions into a unified modeling framework that enhances predictions at both the case and system levels, and supports data-driven decision-making and interventions to improve overall system performance.

Keywords

Predictive Process Monitoring, Suffix Prediction, Event Log Prediction, Inter-Case Dependencies, Deep Learning

1. Background and Motivation

Business processes supported and executed by information systems generate event data that serve as the foundation for process mining, a data-driven approach to analyzing and optimizing business processes [1]. Predictive Process Monitoring (PPM) is a subfield of process mining that focuses on developing predictive models to forecast the future progression of ongoing cases [2]. Accurate PPM models can offer valuable operational support by, for example, alerting staff of potential delays or recommending actions based on predicted outcomes.

Within PPM research, an instance of a process execution is referred to as a *case*, which comprises a series of *events*, each reflecting an executed activity with a timestamp. Events belonging to the same case are typically temporally ordered to form a *trace*. The case ID, activity label, and timestamp are three attributes required to define an event and capture the control-flow perspective of a process. Other perspectives can be reflected by optional event attributes (e.g., resources) or case attributes (e.g., product types), forming the basis for the growing interest in data-aware PPM models [3, 4, 5].

Inheriting the case-centric view from traditional process mining, PPM approaches conventionally use the incomplete trace of a single case (*trace prefix*) as input, and generate predictions concerning specific aspects of this given case. The main categories of predictions include next event prediction [6], suffix prediction [4, 7], remaining time prediction [8], and outcome prediction [9].

However, each case is executed within the broader context of a system, and a strictly case-centric approach is inherently limited as it overlooks system-level dynamics in both the inputs it considers and the outputs it seeks to predict.

With respect to input data, the exclusive use of intra-case data has been increasingly identified as a key limitation [10, 11, 12]. In response, inter-case approaches have emerged that incorporate

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 23rd International Conference on Business Process Management (BPM 2025), Seville, Spain, August 31st to September 5th, 2025.

xiaomeng.he@kuleuven.be (X. He)
https://github.com/Xiaomeng-He (X. He)

dependencies among cases as additional features [13, 14, 15], grounded in the observation that cases often influence one another or are jointly affected by overarching system conditions. Other lines of work enrich event logs with external contextual data [16, 17, 18] or with the interactions between objects and events [19, 20]. Our first research objective builds on inter-case PPM by addressing major shortcomings of existing solutions—reliance on manually selected and engineered features and a predominant focus on remaining time prediction. We aim to develop methods that automatically capture inter-case dynamics from event logs for a wider range of predictive tasks.

With respect to model outputs, the case-centric focus is even more pronounced, as case-level predictions dominate the PPM literature. Few exceptions exist, such as process model forecasting [21] or Work-In-Progress prediction [22]. Notably, a recent study [23] represents event logs as a single "traceless" event sequence ordered as being executed, and estimates entropy rates to assess its predictability. The findings suggest that system-level event sequences exhibit learnable structure, laying the groundwork for modeling system behavior. In a related line of work, streaming process mining also treats event streams as an infinite sequence of events [24, 25] . System-level predictions, such as "traceless" event stream modeling or system load forecasting, can enable applications like anomaly detection and dynamic resource allocation. Our second research objective addresses the current gap in system-level prediction by developing models capable of forecasting future system behavior.

Building from the first two objectives, the third research objective aims to enhance decision-making by integrating system behavior modeling with Prescriptive Process Monitoring (PresPM). PresPM prescribes interventions (treatment) to optimize process outcomes or efficiency [26]. A growing trend in this area is the application of causal inference to estimate treatment effects by comparing potential outcomes (with and without treatment) conditioned on features characterizing a case's current state [27, 28]. Although most research assumes that treatments affect only the targeted case, this assumption is challenged in dynamic environments where concurrently running cases may interact. For instance, as noted by [26], assigning a resource to a delayed case at the expense of others can lead to increased overall delays. Such inter-case dependencies should be considered to maximize the total net benefit of interventions. Furthermore, most existing approaches focus on case-level treatments, such as prescribing the next task or assigning a resource for a case [26]. Cross-case or system-level interventions, like reprioritizing or reordering cases, have received limited attention, despite their practical importance in real-world process management. Our third research objective focuses on developing PresPM approaches that account for inter-case dependencies and enable system-level interventions.

Based on the background and research objectives, the following three Research Questions (RQs) are formulated:

- RQ1: What approaches can be developed to automatically extract and utilize inter-case dependencies from prior process executions to improve case-level predictions?
- RQ2: What approaches can be developed to predict system-level behavior using historical event data?
- RQ3: How can inter-case dependency mining and system-level prediction be integrated and extended to support decision-making?

2. Research Methodology

This doctoral project is primarily guided by the Design Science Research (DSR) methodology in information systems [29]. The research process involves identifying the research problem and motivation, designing a suitable artifact (in this case, a deep learning technique), and demonstrating and evaluating its utility in addressing the problem.

This section outlines the planned methodological approach for each research question, along with the proposed solutions. An overview of the proposed solutions is presented in Figure 1.

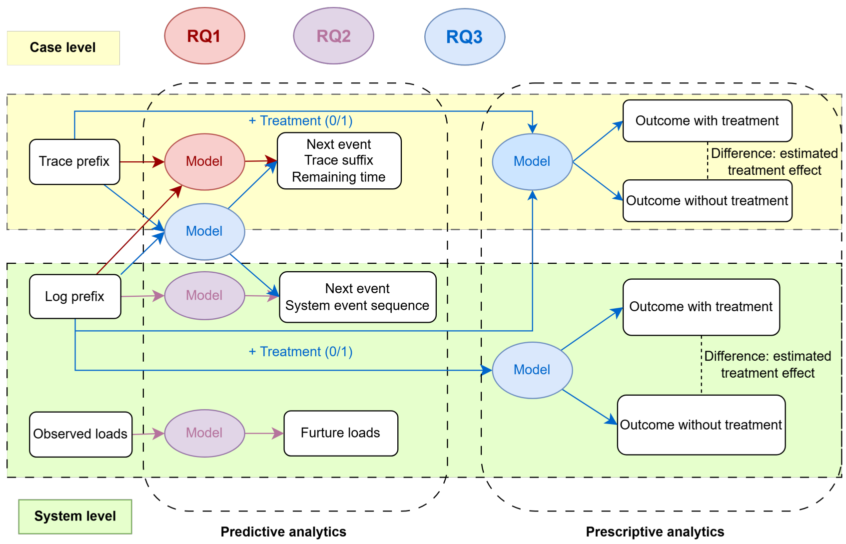

Figure 1: Overview of proposed solutions

2.1. RQ1: What approaches can be developed to automatically extract and utilize inter-case dependencies from prior process executions to improve case-level predictions?

Motivated by advances in Natural Language Processing (NLP) and the observed similarities between event logs and natural language text [30], sequential deep learning models have been adopted as effective tools for PPM [31]. In this context, a trace is modeled as a sequence of events. Following the same principle, to incorporate inter-case dynamics, we propose representing events from multiple cases as a single, temporally ordered sequence that reflects the global execution flow of the system, and using this system-level sequence as an additional input to the model, alongside the trace prefix.

One key challenge is designing a model that can automatically identify relevant inter-case patterns from this additional input sequence, while effectively filtering noise from the massive information in preceding events. We will address this by evaluating various deep learning architectures as base models and designing customized neural network structures to handle the multi-input characteristics of the tasks. Candidate base architectures include Long Short-Term Memory (LSTM) networks, which are widely used for PPM tasks [6, 30, 31], while LSTM with attention [32], Transformer-based models [7] and the more recent Extended Long Short-Term Memory (xLSTM) [33] will also be investigated for their superior performance in modeling long sequence.

Integrating heterogeneous input sources including intra- and inter-case information poses another challenge. Inspired by multimodal deep learning [34], which seeks to integrate data from diverse sources, distributions, and modalities into a unified representation space, we will explore various data fusion strategies, ranging from simple concatenation to more sophisticated hierarchical feature fusion methods [35].

2.2. RQ2: What approaches can be developed to predict system-level behavior using historical event data?

As system-level prediction remains a largely underexplored area in business process research, the first key challenge is defining meaningful prediction targets and establishing robust baselines and evaluation methods. Following the research on traceless event sequencing [23], we propose representing system behavior as an (infinite) temporally ordered event sequence, aiming to forecast the sequence of future events and their timestamps within a specified horizon, based on the sequence of observed consecutive events. Additionally, predicting system load (measured by the number of active cases over time) is also of practical relevance (e.g., for resource allocation) and can be formulated as a time series forecasting problem. Related works include research on Work-In-Progress (WIP) prediction which applies time series models to estimate future WIP levels with LSTM [36] or Temporal Convolutional Networks [22].

Modeling the complexity and stochastic behavior of the underlying system is inherently challenging, particularly when processing and generating exceptionally long and variable-length event sequences. Recent advances in architectures capable of capturing long-range dependencies in Large Language Models (LLMs) field, such as Transformer [37] and Mamba [38], or diffusion models [39] as an alternative to autoregressive generation approaches, provide valuable inspiration and opportunities. Nevertheless, despite their similarities, event data differs from natural language due to its strong temporal dependencies and complex interaction patterns embedded in additional case attributes and event attributes, such as interactions between resources. To effectively capture these characteristics, we aim to adapt modern neural architectures and develop specialized neural network architectures explicitly tailored to the structural and temporal nature of event data.

2.3. RQ3: How can inter-case dependency mining and system-level prediction be integrated and extended to support decision-making?

Building on the findings of RQ1 and RQ2, RQ3 investigates how to integrate case-level predictions (enriched with inter-case dependencies) and system-level predictions into a unified framework, and extend system behavior modeling to support decision-making.

We first aim to develop a unified modeling approach capable of generating both case-level and system-level predictions. This will be pursued through a Multi-Task Learning (MTL) framework, where both tasks share representations and parameters. MTL is expected to enhance generalization by allowing the model to leverage commonalities across related tasks.

To bridge the gap between prediction and proactive decision-making, we aim to extend system behavior modeling techniques into a PresPM framework. Inter-case dynamics relate closely to the broader notion of interference in the causal inference literature. A widely studied approach to handling interference is network interference [40, 41], where units are connected through a network structure, and causal inference is conditioned on features extracted from this network.

In our context, to incorporate inter-case interactions when optimizing case-level metrics, we propose grounding causal inference on a combination of intra-case features and inter-case features, which are extracted, for example, from system-level event sequences using neural networks. This approach allows treatment effect estimation to account for the broader inter-case context in which a case operates.

To optimize global system-level metrics while considering interactions among cases, we further propose conditioning treatment effect estimates on representations of the overall system state, such as system-level event sequences or interaction networks derived from observed event logs. In this setting, treatments could be prescribed event sequences, including interventions for specific cases (e.g., the next task to perform) or cross-case interventions (e.g., altering the execution order of multiple cases). The effects of such interventions will then be estimated by comparing predicted system trajectories, with and without the prescribed treatment, based on the system behavior model.

3. Research Progress and Intermediate Results

For RQ1, an initial artifact has been designed and evaluated, which is an multi-input encoder-decoder model for the task of suffix prediction. Work is ongoing to refine and extend this artifact. The research related to RQ2 and RQ3 is currently in the literature review and conceptualization phase.

The first technique designed for RQ1 adopts an LSTM-based encoder-decoder architecture (Seq2Seq model). The model takes both trace prefixes and log prefixes as input. The trace prefix captures the intra-case behavior of the case under prediction, while the log prefix represents a temporally ordered sequence of events from multiple cases across the entire system. The integration of the log prefix constitutes the primary novelty of this approach, as it allows the model to learn inter-case dependencies and system-level patterns in addition to case-specific information. The model architecture employs two separate LSTM encoders: one for the trace prefix and one for the log prefix. The final hidden states from both encoders are concatenated to form a joint context vector. A dual-decoder mechanism is then used to simultaneously predict the suffix of activity labels and the suffix of timestamps.

The proposed model has been evaluated on three real-life event logs: BPIC2017, BPIC2019, and BAC. To assess its effectiveness, the integrated Seq2Seq model is compared against several benchmark models. A key baseline among them is a trace-based Seq2Seq model, which shares the same encoder-decoder architecture but uses only the trace prefix as input. Table 1 presents the performance comparison between the proposed integrated Seq2Seq model and the trace-based baseline across all three datasets. The integrated model consistently outperforms the trace-based model in activity label suffix prediction. For timestamp suffix prediction, the integrated model also achieves superior performance in most cases. It outperforms the baseline on all datasets except BPIC2017 when evaluated using MAE.

Table 1
Performance of different models. Normalized Damerau–Levenshtein distance (DL distance) for activity label suffix prediction; Mean Absolute Error (MAE) and Mean Squared Logarithmic Error (MSLE) for timestamp suffix prediction.

Model	BPIC2017			BPIC2019			BAC		
	DL distance	MAE	MSLE	DL distance	MAE	MSLE	DL distance	MAE	MSLE
Trace-based Seq2Seq model	0.5888	908	14.09	0.1880	15911	24.71	0.2905	62.34	2.79
Integrated Seq2Seq model	0.5621	919	14.04	0.1764	15779	22.03	0.2689	61.27	2.74

4. Future Plan

Figure 2 illustrates the proposed timeline for future research activities. Currently, work is underway to extend the technique developed for RQ1, while a comprehensive literature review for RQ2 has recently commenced. Given the complexity of the topic, we anticipate that the conceptualization of RQ3 will require more time and careful investigation.

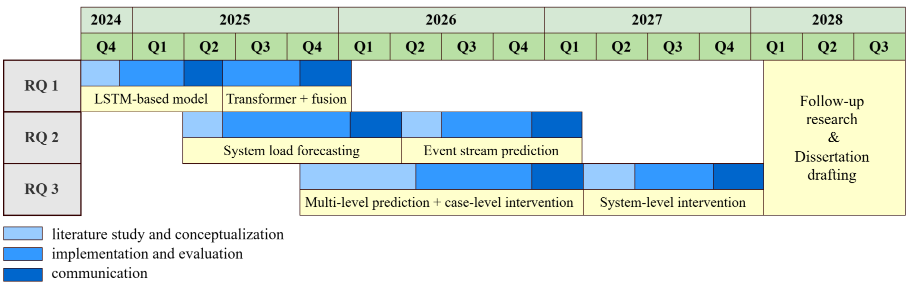

Figure 2: Research plan

Acknowledgments

This Ph.D. project is supervised by Prof. dr. Jochen De Weerdt and Prof. dr. Johannes De Smedt from KU Leuven, and Prof. dr. Seppe vanden Broucke from Ghent University.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT-4o for grammar and spelling check. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] W. M. P. van der Aalst, Process mining: A 360 degree overview, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, Springer, Cham, 2022, pp. 3–34. doi:10.1007/978-3-031-08848-3_1.

[2] P. Ceravolo, M. Comuzzi, J. De Weerdt, C. Di Francescomarino, F. M. Maggi, Predictive process monitoring: concepts, challenges, and future research directions, Process Science 1 (2024) 2. doi:10.1007/s44311-024-00002-4.

[3] N. Navarin, B. Vincenzi, M. Polato, A. Sperduti, LSTM networks for data-aware remaining time prediction of business process instances, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–7. doi:10.1109/SSCI.2017.8285184.

[4] B. R. Gunnarsson, S. v. Broucke, J. De Weerdt, A direct data aware LSTM neural network architecture for complete remaining trace and runtime prediction, IEEE Transactions on Services Computing 16 (2023) 2330–2342. doi:10.1109/TSC.2023.3245726.

[5] J. Roider, D. Zanca, B. M. Eskofier, Efficient training of recurrent neural networks for remaining time prediction in predictive process monitoring, in: A. Marrella, M. Resinas, M. Jans, M. Rosemann (Eds.), Business Process Management, Springer, Cham, 2024, pp. 238–255. doi:10.1007/978-3-031-70396-6_14.

[6] N. Tax, I. Verenich, M. La Rosa, M. Dumas, Predictive business process monitoring with LSTM neural networks, in: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering, Springer, Cham, 2017, pp. 477–492. doi:10.1007/978-3-319-59536-8_30.

[7] B. Wuyts, S. Vanden Broucke, J. De Weerdt, SuTraN: an encoder-decoder transformer for full-context-aware suffix prediction of business processes, in: 2024 6th International Conference on Process Mining (ICPM), 2024. doi:10.1109/ICPM63005.2024.10680671.

[8] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, I. Teinemaa, Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring, ACM Transactions on Intelligent Systems and Technology 10 (2019). doi:10.1145/3331449.

[9] I. Teinemaa, M. Dumas, M. L. Rosa, F. M. Maggi, Outcome-oriented predictive process monitoring: Review and benchmark, ACM Trans. Knowl. Discov. Data 13 (2019). doi:10.1145/3301300.

[10] C. Di Francescomarino, C. Ghidini, Predictive process monitoring, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, Springer, Cham, 2022, pp. 320–346. doi:10.1007/978-3-031-08848-3_10.

[11] A. Grinvald, P. Soffer, O. Mokryn, Inter-case properties and process variant considerations in time prediction: A conceptual framework, in: A. Augusto, A. Gill, S. Nurcan, I. Reinhartz-Berger, R. Schmidt, J. Zdravkovic (Eds.), Enterprise, Business-Process and Information Systems Modeling, Springer, Cham, 2021, pp. 96–111. doi:10.1007/978-3-030-79186-5_7.

[12] Y. Dubinsky, P. Soffer, I. Hadar, Detecting cross-case associations in an event log: toward a pattern-based detection, Software and Systems Modeling 22 (2023) 1755–1777. doi:10.1007/s10270-023-01100-w.

[13] A. Senderovich, C. D. Francescomarino, F. M. Maggi, From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring, Information Systems 84 (2019) 255–264. doi:10.1016/j.is.2019.01.007.

[14] E. L. Klijn, D. Fahland, Identifying and reducing errors in remaining time prediction due to inter-case dynamics, in: 2020 2nd International Conference on Process Mining (ICPM), 2020, pp. 25–32. doi:10.1109/ICPM49681.2020.00015.

[15] B. R. Gunnarsson, S. vanden Broucke, J. De Weerdt, LS-ICE: A load state intercase encoding framework for improved predictive monitoring of business processes, Information Systems 125 (2024) 102432. doi:10.1016/j.is.2024.102432.

[16] I. Teinemaa, M. Dumas, F. M. Maggi, C. Di Francescomarino, Predictive business process monitoring with structured and unstructured data, in: M. La Rosa, P. Loos, O. Pastor (Eds.), Business Process Management, Springer, Cham, 2016, pp. 401–417. doi:10.1007/978-3-319-45348-4_23.

[17] C. Di Francescomarino, C. Ghidini, F. M. Maggi, G. Petrucci, A. Yeshchenko, An eye into the

future: Leveraging a-priori knowledge in predictive business process monitoring, in: J. Carmona, G. Engels, A. Kumar (Eds.), Business Process Management, Springer, Cham, 2017, pp. 252–268. doi:10.1007/978-3-319-65000-5_15.

[18] A. Yeshchenko, F. Durier, K. Revoredo, J. Mendling, F. Santoro, Context-aware predictive process monitoring: The impact of news sentiment, in: H. Panetto, C. Debruyne, H. A. Proper, C. A. Ardagna, D. Roman, R. Meersman (Eds.), On the Move to Meaningful Internet Systems. OTM 2018 Conferences, Springer, Cham, 2018, pp. 586–603. doi:10.1007/978-3-030-02610-3_33.

[19] R. Galanti, M. de Leoni, N. Navarin, A. Marazzi, Object-centric process predictive analytics, Expert Systems with Applications 213 (2023) 119173. doi:10.1016/j.eswa.2022.119173.

[20] T. K. Smit, H. A. Reijers, X. Lu, HOEG: A new approach for object-centric predictive process monitoring, in: G. Guizzardi, F. Santoro, H. Mouratidis, P. Soffer (Eds.), Advanced Information Systems Engineering, Springer, Cham, 2024, pp. 231–247. doi:10.1007/978-3-031-61057-8_14.

[21] J. De Smedt, A. Yeshchenko, A. Polyvyanyy, J. De Weerdt, J. Mendling, Process model forecasting and change exploration using time series analysis of event sequence data, Data Knowledge Engineering 145 (2023) 102145. doi:10.1016/j.datak.2023.102145.

[22] Y. Mehrdad Bibalan, B. Far, F. Eshragh, B. Ghiyasian, Work in progress prediction for business processes using temporal convolutional networks, in: H. Fujita, R. Cimler, A. Hernandez-Matamoros, M. Ali (Eds.), Advances and Trends in Artificial Intelligence. Theory and Applications, Springer, Singapore, 2024, pp. 109–121. doi:10.1007/978-981-97-4677-4_10.

[23] P. Pfeiffer, P. Fettke, Trace vs. time: Entropy analysis and event predictability of traceless event sequencing, in: A. Marrella, M. Resinas, M. Jans, M. Rosemann (Eds.), Business Process Management Forum, Springer, Cham, 2024, pp. 72–89. doi:10.1007/978-3-031-70418-5_5.

[24] A. Burattin, Streaming process mining, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, Springer, Cham, 2022, pp. 349–372. doi:10.1007/978-3-031-08848-3_11.

[25] T. Verbeek, M. Hassani, Handling catastrophic forgetting: Online continual learning for next activity prediction, in: M. Comuzzi, D. Grigori, M. Sellami, Z. Zhou (Eds.), Cooperative Information Systems, Springer, Cham, 2025, pp. 225–242. doi:10.1007/978-3-031-81375-7_13.

[26] K. Kubrak, F. Milani, A. Nolte, M. Dumas, Prescriptive process monitoring: Quo vadis?, PeerJ Computer Science 8 (2022) e1097. URL: https://doi.org/10.7717/peerj-cs.1097.

[27] M. Shoush, M. Dumas, Prescriptive process monitoring under resource constraints: A causal inference approach, in: J. Munoz-Gama, X. Lu (Eds.), Process Mining Workshops, Springer, Cham, 2022, pp. 180–193. doi:10.1007/978-3-030-98581-3_14.

[28] Z. Dasht Bozorgi, I. Teinemaa, M. Dumas, M. La Rosa, A. Polyvyanyy, Prescriptive process monitoring based on causal effect estimation, Information Systems 116 (2023) 102198. doi:10.1016/j.is.2023.102198.

[29] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. C. and, A design science research methodology for information systems research, Journal of Management Information Systems 24 (2007) 45–77. doi:10.2753/MIS0742-1222240302.

[30] J. Evermann, J.-R. Rehse, P. Fettke, Predicting process behaviour using deep learning, Decision Support Systems 100 (2017) 129–140. doi:10.1016/j.dss.2017.04.003.

[31] E. Rama-Maneiro, J. C. Vidal, M. Lama, Deep learning for predictive business process monitoring: Review and benchmark, IEEE Transactions on Services Computing 16 (2023) 739–756. doi:10.1109/TSC.2021.3139807.

[32] H. Weytjens, J. De Weerdt, Process outcome prediction: CNN vs. LSTM (with attention), in: A. Del Río Ortega, H. Leopold, F. M. Santoro (Eds.), Business Process Management Workshops, Springer, Cham, 2020, pp. 321–333. doi:10.1007/978-3-030-66498-5_24.

[33] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, S. Hochreiter, xlstm: Extended long short-term memory, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 107547–107603. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/c2ce2f2701c10a2b2f2ea0bfa43cfaa3-Paper-Conference.pdf.

[34] J. Gao, P. Li, Z. Chen, J. Zhang, A survey on deep learning for multimodal data fusion, Neural Computation 32 (2020) 829–864. doi:10.1162/neco_a_01273.

[35] F. Zhao, C. Zhang, B. Geng, Deep multimodal data fusion, ACM Computing Surveys 56 (2024). doi:10.1145/3649447.

[36] V. Gallina, L. Lingitz, J. Breitschopf, E. Zudor, W. Sihn, Work in progress level prediction with long short-term memory recurrent neural network, Procedia Manufacturing 54 (2021) 136–141. doi:10.1016/j.promfg.2021.07.047.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL: https://arxiv.org/abs/2312.00752. arXiv:2312.00752.

[39] M. Zeng, F. Regol, M. Coates, Interacting diffusion processes for event sequence forecasting, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[40] R. Bhattacharya, D. Malinsky, I. Shpitser, Causal inference under interference and network uncertainty, in: R. P. Adams, V. Gogate (Eds.), Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, volume 115 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 1028–1038. URL: https://proceedings.mlr.press/v115/bhattacharya20a.html.

[41] Y. Ma, V. Tresp, Causal inference under networked interference and intervention policy enhancement, in: A. Banerjee, K. Fukumizu (Eds.), Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 3700–3708. URL: https://proceedings.mlr.press/v130/ma21c.html.