A MULTI-AGENT RETRIEVAL-AUGMENTED FRAMEWORK FOR WORK-IN-PROGRESS PREDICTION

Yousef Mehrdad Bibalan [1], Behrouz Far [1], Mohammad Moshirpour [2], and Bahareh Ghiyasian [3]

[1] University of Calgary, Canada,
[2] University of California, Irvine, USA,
[3] Google, USA

ABSTRACT

Work-in-Progress (WiP) prediction is critical for predictive process monitoring, enabling accurate anticipation of workload fluctuations and optimized operational planning. This paper proposes a retrieval-augmented, multi-agent framework that combines retrieval-augmented generation (RAG) and collaborative multi-agent reasoning for WiP prediction. The narrative generation component transforms structured event logs into semantically rich natural language stories, which are embedded into a semantic vector-based process memory to facilitate dynamic retrieval of historical context during inference. The framework includes predictor agents that independently leverage retrieved historical contexts and a decision-making assistant agent that extracts high-level descriptive signals from recent events. A fusion agent then synthesizes predictions using ReAct-style reasoning over agent outputs and retrieved narratives. We evaluate our framework on two real-world benchmark datasets. Results show that the proposed retrieval-augmented multi-agent approach achieves competitive prediction accuracy, obtaining a Mean Absolute Percentage Error (MAPE) of 1.50% on one dataset, and surpassing Temporal Convolutional Networks (TCN), Long Short-Term Memory (LSTM), and persistence baselines. The results highlight improved robustness, demonstrating the effectiveness of integrating retrieval mechanisms and multi-agent reasoning in WiP prediction.

KEYWORDS

Predictive Process Monitoring, Work-in-Progress, Retrieval-Augmented Generation, Large Language Models, Multi-Agent Framework

1. INTRODUCTION

Predictive process monitoring is essential in modern management because it helps forecast workload changes and supports effective resource planning [1]. A key task in predictive process monitoring is work-in-progress (WiP) prediction—estimating the number of active tasks at any given moment—which improves staffing, capacity planning, and overall operational efficiency. Traditional predictive process monitoring techniques employ deep learning models such as recurrent neural networks, long short-term memory networks, and transformers to capture sequential dependencies in event logs [2–5]. More recently, narrative encoding has been introduced to enrich these models by transforming raw event traces into structured textual stories. Methods such as LUPIN and SNAP fine-tune pre-trained large language models (LLMs) to leverage this natural-language representation, yielding improved predictive accuracy [6,7].

David C. Wyld et al. (Eds): DPPR, CNDC, AIAA, DSA, WeST, ACITY, IoTE – 2025
pp. 107-118, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.152409

Although narrative approaches enhance context modeling, they remain purely generative and are typically confined to a single contextual perspective.

Retrieval-augmented generation (RAG) offers a compelling way to ground generative predictions in concrete past cases by fetching relevant documents or data fragments that augment the model's input [8]. RAG has achieved notable success in domains such as open domain question answering and time-series forecasting, yet it has not been systematically applied to predictive process monitoring [9]. Separately, agentic LLM architectures-where multiple specialized agents collaborate under an orchestrator-have been proposed for diagnostic tasks in process mining but focus solely on retrospective analysis and lack retrieval components [10,21]. To the best of our knowledge, no existing work combines retrievalaugmented generation with multi-agent reasoning for WiP prediction.

To address these gaps, this paper proposes a retrieval-augmented, multi-agent framework for predicting WiP. Event logs are transformed into temporal and semantic narratives, indexed in a vector-based memory for RAG-based retrieval. Predictor agents, each aligned with a specific narrative view, generate individual predictions in a zero-shot manner, without any task-specific fine-tuning. A decision-making assistant extracts high-level signals, such as momentum and variability, to guide interpretation. A fusion agent then integrates these predictions and insights using ReAct-style reasoning to produce a robust WiP prediction. By integrating retrieval at every stage and orchestrating agent collaboration, our framework grounds its outputs in both agent-generated forecasts and relevant historical cases, ensuring accuracy and contextual sensitivity. This paper makes three main contributions as follows:

- Transform raw event logs into rich narrative stories that capture multiple temporal and semantic dimensions of process behavior, enabling LLMs to reason over structured, human-readable contexts.
- Show that RAG can serve as a dynamic memory-replacing traditional in-memory storage by fetching relevant past cases to inform forecasts and improve transparency.
- Develop a zero-shot, multi-agent system in which specialized predictor agents, a decision-making assistant agent, and a fusion agent collaborate-integrating narrative perspectives and retrieved evidence-to produce accurate WiP predictions.

2. RELATED WORK

Predictive process monitoring aims to forecast the future behaviour of ongoing process instances using historical event logs [24,18]. Traditional approaches often rely on sequential models such as LSTMs, GRUs, or encoder-decoder networks, which model the process trace prefix and predict the next activity or suffix [22,2,23]. More recent advancements have explored the potential of LLMs for these tasks [15]. For example, Xu and Fang proposed LLM4NT, a domain-specific adaptation of the Qwen 2.5B decoder with a cross-attention mechanism, to predict next timestamps using event trace inputs [12]. Their approach demonstrates strong performance on business process logs, proving that LLMs can generalize beyond natural language domains for predictive tasks. However, most of these LLM-based methods operate in a monolithic fashion and do not incorporate retrieval or agent-based modularity [12,16].

Another prominent development is the generation of semantic narratives from structured event logs. The SNAP framework [7] addresses the limitations of conventional models by constructing natural-language semantic stories from process traces. These stories encode multiple event attributes into a single narrative, enabling small and large language models to learn meaningful representations for next activity prediction. SNAP outperforms 11 state-of-the-art models across

six benchmark datasets and shows significant gains in contexts rich in textual or conversational data. However, SNAP relies on a single-shot story template generated by an LLM and lacks a retrieval or agentic layer for dynamic adaptation.

Despite the rapid adoption of RAG in NLP, its application in process mining and predictive monitoring is still emerging and several agentic frameworks exhibit RAG-like behavior on XES-based logs. For instance, Jessen et al. proposed a conversational LLM agent that uses prompt grounding via process ontologies and dynamic SQL generation for query refinement [13]. Similarly, the CrewAI architecture by Berti et al. embeds LLM agents that invoke deterministic tools like PM4Py and SQL to carry out analytic tasks such as conformance checking or root-cause diagnosis [10]. These architectures capture the essence of RAG, but are primarily analytical and lack integration with predictive modeling workflows.

The use of AI agents in process mining has been explored through orchestrated LLM powered systems that assign specific responsibilities to different agents. CrewAI, for example, structures agents around process mining tasks, allowing modular decomposition and traceable outputs. Each agent is equipped with specialized prompts and access to deterministic APIs, creating a "think-act" architecture suited for explainable automation [10]. Aratchige and Ilmini provide a comprehensive survey of such agentic architectures and highlight their applicability for transparent decision-making [11].

While foundational work exists on narrative generation, LLM-based suffix prediction, and prompt-grounded analytics, the integration of RAG and multi-agent LLMs for predictive monitoring—especially over structured and semantic process logs—remains an open challenge. The proposed multi-agent RAG+LLM framework aims to bridge this gap by combining retrieval, reasoning, and modular orchestration in a scalable predictive pipeline.

3. PROPOSED APPROACH

This section describes the architecture and design principles of the proposed framework. We begin by defining the WiP event log structure, then outline how narrative stories and process memory are constructed, followed by the agents' design and their interactions.

3.1. WiP Event Log

An event log, a crucial input for predictive process monitoring (PPM), is defined as a sequence of events $L = \{e_i\}$, where each event is defined as

$$e_i = (c,a,t,(k_1,v_1),\ldots,(k_j,v_j))$$

Here, c denotes the case identifier, a is the activity name, t is the timestamp, and each pair (k_j,v_j) represents an attribute and its corresponding value. This structure captures the temporal and contextual information of a process execution over time. We define a WiP Event Log as an event log that includes WiP events, where each *WiP event* is represented as a 10-dimensional vector:

$$x = (x_t^{(d_w)}, x_t^{(d_m)}, x_t^{(d_y)}, x_t^{(o)}, x_t^{(h)}, x_t^{(l)}, x_t^{(c)}, x_t^{(n)}, x_t^{(d)}, x_t^{(s)})^T$$

In this representation, $x_t^{(o)}, x_t^{(h)}, x_t^{(l)}, x_t^{(c)}$ denote the Open, High, Low, and Close values of WiP. The components $x^{(d)}_t$, $x^{(n)}_t$, $x^{(s)}_t$ indicate the number of done, new, and started items, respectively, during the time period t.

3.2. Transform WiP Event Log to Stories

To bridge structured WiP data and natural language reasoning within the framework, we convert each WiP event into a semantically meaningful textual *story* using a LLM. The input to this transformation is a WiP Event Log, and the output is a human-readable story that captures the operational dynamics of the process during a specific time interval. These stories represent the process state in natural language and support retrieval within the framework. Specifically, we generate two types of stories per time unit:

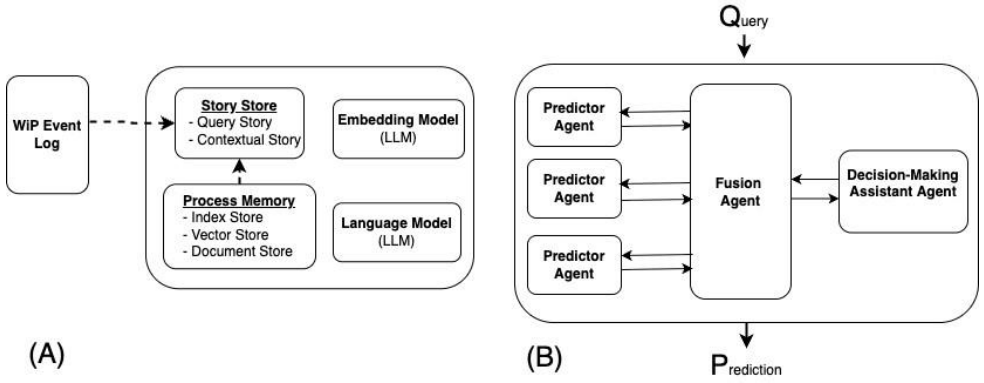

Figure 1. Architecture Overview. (A) Narrative generation and memory construction pipeline: WiP event logs are transformed into query and contextual stories via LLMs and stored in a semantic vector-based process memory. (B) Multi-agent prediction framework, where multiple predictor agents and a decision-making assistant feed into a fusion agent that generates the final WiP prediction.

- **Query Story**: A concise, information-rich description used as direct input to the LLM prompt during inference.
  Example: "The WiP items opened at 55, reached a high of 70 and a low of 55, before closing at 66, with 10 items completed, 24 new items added, and 21 items started."
- **Contextual Story**: An extended version used to construct the retrieval corpus, which includes the target value.
  Example: "The WiP items opened at 55, reached a high of 70 and a low of 55, before closing at 66, with 10 items completed, 24 new items added, and 21 items started, while the next WiP was expected to remain at 71."

These query and contextual stories function as structured, interpretable knowledge units for retrieval and inference, supporting robust multi-scale forecasting in the framework.

Figure 1(A) provides a high-level view of this narrative generation pipeline. As shown, raw WiP event logs are transformed into query and contextual stories through an LLMbased abstraction process. These stories are then prepared for downstream use in the broader prediction architecture, which will be described in the following sections.

3.3. Retrieval-Augmented Generation as Process Memory

To enable RAG-based prediction, we incorporate a *Process Memory Module* that functions as a semantic retrieval system, storing and retrieving contextual knowledge from previously observed stories. Each story is embedded using an embedding model. Each document in memory corresponds to a story and is stored in a vector store. During inference, a new query story is used

to retrieve the top-n most relevant contextual stories based on cosine similarity. This process memory supports *Contextual Grounding* by injecting real, comparable past examples into the prediction prompt, and *Temporal Generalization* by facilitating pattern recognition over multiple time scales. This memory-augmented structure enables effective reasoning without task-specific fine-tuning while maintaining high contextual relevance and semantic fidelity. By dynamically incorporating similar historical process states, the memory-based retrieval mechanism supports more accurate and temporally grounded predictions within the RAG framework.

3.4. Agents

Our proposed framework employs a team of specialized agents, each assigned a distinct role in the prediction process. Each agent is equipped with its own process memory, and these agents operate collaboratively to interpret historical context, assist in decision-making, and synthesize predictions using information retrieved from their respective memories. The remainder of this section outlines the roles and interactions of the three types of agents in the framework.

3.4.1. Predictor Agent

The prediction component is built around a modular ensemble of Predictor Agents, each specialized for a different temporal view of the process history. Forecasting is decomposed into agents that independently retrieve and reason over historical data slices. Each agent accesses a process memory module for relevant context and operates as a callable tool within the decision strategy. This design enables the system to capture short-term fluctuations, periodic patterns, and long-term trends simultaneously, improving accuracy. The framework is scalable and extensible-new agents can be added to support emerging patterns or domain-specific needs. Together, these agents form the core of a robust, modular, and data-grounded WiP predicting architecture.

3.4.2. Decision-Making Assistant Agents

Complementing the Predictor Agents, Decision-Making Assistant Agents extract high-level patterns and signals-such as trends, anomalies, or workload shifts-from recent process behavior. While they do not produce numeric forecasts, they provide contextual insights that help guide the prediction strategy. Each agent analyzes recent data using lightweight statistical or heuristic methods. Their qualitative outputs, such as trend direction or volatility, inform how the final decision maker (the Fusion agent) weighs and combines predictions. By interpreting evolving process dynamics, assistant agents enhance the system's robustness and adaptability, allowing it to respond to concept drift and structural changes.

3.4.3. Fusion Agent

At the center of the architecture is the Fusion Agent, responsible for coordinating the Predictor and Assistant agents for final prediction. Unlike specialized agents focused on individual tasks, this agent performs higher-level reasoning-evaluating retrieved evidence, incorporating decision-support insights from assistant agents, and selecting or combining predictions. Using a ReAct-style prompting strategy, it interacts iteratively with tools, queries the process memory, and reconciles agent outputs via heuristics or consensus logic. This flexible workflow enables context-sensitive forecasting that adapts to evolving process conditions. The agent abstracts multi-agent complexity and provides a unified prediction interface. Its modular design supports extensibility, allowing the integration of new agents.

Figure 1(B) illustrates how agent outputs flow into the fusion agent to produce the final prediction, completing the multi-agent reasoning layer of the framework.

4. EXPERIMENTAL RESULTS

This section presents the experimental setup used to evaluate the proposed framework. We detail the datasets, benchmarks, and story-generation procedure, describe how the process memory and agents are launched, and finally report and analyze the obtained results.

4.1. Datasets and Benchmarks

We evaluate our framework on two widely used real-world event logs from the 4TU Center for Research Data: BPIC13 Incidents (783 days of incident-management records) and Helpdesk (1,452 days of support-ticket handling) [20,14]. For baselines, we adopt the three best-performing Temporal Convolutional Network (TCN) configurations from [14], denoted as TCN#1–#3, which vary in activation functions and optimizers and achieved the lowest MAPE in prior work. We also include two classical baselines: LSTM and Persistence (Yesterday's WiP), both reported in [14]. This set of benchmarks supports a robust comparison against our multi-agent framework.

4.2. Building Story Storage on Benchmark Datasets

Each WiP event is transformed into a semantically rich *story* using GPT-3.5-Turbo, capturing variation across three temporal granularities:

- **Daily Stories**: Capture fine-grained, day-level fluctuations in WiP dynamics. These stories describe each day's activity in isolation and do not include any explicit temporal information.
- **Weekday-Aware Stories**: Encode cyclical behavior by incorporating the weekday context (e.g., "on Monday"), enabling the model to learn patterns tied to recurring time-based phenomena.
- **Windowed Stories**: Aggregate daily data over a rolling window (set to seven in our experiments, but configurable), providing a higher-level view.

These story types enable multi-resolution temporal reasoning, capturing short-term patterns (daily), cyclical behaviours (weekday-aware), and longer-term trends (windowed), each can influence future workload differently. By supporting diversity, the architecture integrates complementary insights and improves overall forecast robustness and adaptability. For each WiP event, we generate two story variants: a *Query Story* and a *Contextual Story* (defined in Section 3.3). The Query Story is used during inference to construct the input prompt, while the Contextual Story is indexed in the process memory for retrieval.

4.3. Enabling Process Memory for Story Storage

The narrative stories are embedded using the BAAI/bge-base-en-v1.5 model and indexed via the LlamaIndex framework [19], enabling efficient dense retrieval over semantically encoded text. Each story is stored as a separate document in a semantic vector index. At inference, the current event is converted into a query story, which retrieves the top five most similar contextual stories based on cosine similarity. These retrieved narratives are incorporated into the input prompt for the predictor agent, grounding the forecast in relevant historical context. To preserve causality, only stories with timestamps prior to the target event are available during retrieval, forming a temporally bounded memory aligned with real-world forecasting constraints.

Although the current setup retrieves all prior stories, the architecture supports configurable strategies, such as limiting to recent weeks or filtering based on semantic similarity or predictive

utility. These options can enhance retrieval precision and efficiency, particularly in dynamic or resource-limited environments.

4.4. Launch Agents

4.4.1. Predictor Agents

To support prediction across multiple temporal granularities, we design three specialized predictor agents. The *DailyMemoryAgent* captures short-term day-to-day variations, the *WeekdayAwareAgent* models recurring weekday cycles, and the *WindowedAgent* abstracts longer-term trends through using rolling seven-day aggregates. Each agent follows a fourstep workflow:

1. *Query construction*: Convert the current WiP event into a query story.
2. *Context retrieval*: Retrieve the top five semantically closest stories from the process memory.
3. *Prompt assembly*: Build a structured prompt that integrates the query story, task instructions, and retrieved examples.
4. *LLM Inference*: Send the prompt to the language model and return the predicted WiP.

The inference is performed using OpenAI O3-mini, though the framework allows substitution with any compatible LLM. This modular design enables reasoning across diverse temporal patterns and supports the addition of new agents. It also improves transparency, reusability, and robustness in forecasting WiP under dynamic process conditions.

4.4.2. Decision-Making Assistant Agent

The system includes a dedicated Decision-Making Assistant Agent, called the *Trend Analyst*, which performs lightweight trend analysis over a rolling 7-day window using a simple moving average. It evaluates the directional change in smoothed values to classify recent workload momentum. Based on the observed trend, the agent outputs a short textual description such as “WiP has been increasing significantly” or “WiP has been relatively stable.” These insights are passed to the Fusion Agent to guide prediction weighting, enabling the system to adapt to recent process dynamics while maintaining interpretability.

4.4.3. Fusion Agent

The Fusion Agent serves as the central orchestrator, responsible for integrating outputs from all agents to generate the final WiP prediction. Using a ReAct-style prompting mechanism, it combines intermediate predictions and qualitative trend insights through multi-step reasoning. The Fusion Agent performs the following operations:

1. Queries all Predictor Agents and collects their individual predictions.
2. Retrieves trend descriptions from the Decision-Making Assistant Agent.
3. Compares and contrasts the numerical predictions and trend indicators.
4. Applies decision rules or ensemble heuristics to resolve discrepancies–e.g., prioritizing WindowedAgent in stable periods and DailyMemoryAgent during rapid shifts.
5. Delivers the final forecast and, optionally, a rationale detailing how each agent was weighted.

This strategy enables adaptive, interpretable fusion of agent outputs and supports seamless integration of new agents without altering the core architecture.

4.5. Results

This subsection reports our framework's performance on both datasets, comparing agent accuracy and adaptability via visuals and error metrics.

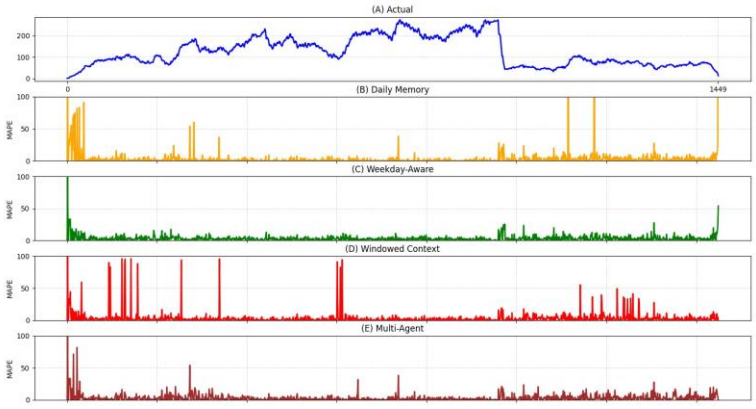

Figure 2. Prediction comparison for the Helpdesk dataset. Top panel shows the actual WiP; bottom panel displays the MAPE error for Daily Memory, Weekday-Aware, Windowed Context, and Multi-Agent models.

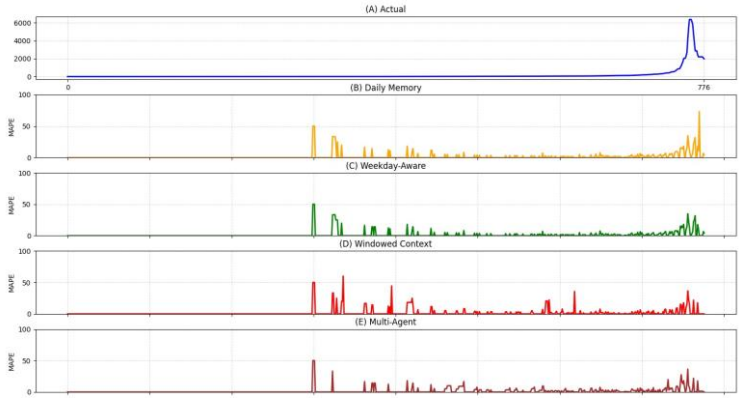

Figure 3. Prediction comparison for the BPIC13 Incidents dataset. Top panel shows the actual WiP; bottom panel displays the MAPE error for Daily Memory, Weekday-Aware, Windowed Context, and Multi-Agent models.

4.5.1. Visual Comparison of Model Predictions

Figures 2 and 3 display the actual WiP values and corresponding MAPE trends for each model configuration across the Helpdesk and BPIC13 Incidents datasets. The top panel in each figure shows the actual WiP value, followed by model-specific MAPE plots for Daily Memory, Weekday-Aware, Windowed Context, and Fusion or Multi-Agent.

In the Helpdesk dataset, Daily Memory displays large error spikes during unstable periods, reflecting its limited ability to adapt to rapid changes. Adding weekday context reduces early variance by capturing cyclical patterns, while the windowed model improves mid-range performance but still suffers from spikes. In contrast, the Multi-Agent model shows the most stable behavior, maintaining a consistently low MAPE throughout the timeline, demonstrating its robustness across volatile periods. In the BPIC13 Incidents dataset, all methods experience

difficulty during the sharp workload increase near the end. However, the multi-agent model maintains significantly lower error across this transition, underscoring its strength in generalizing over abrupt process changes by leveraging multiple memory and prediction pathways, outperforming simpler models during regime shifts.

These visual trends reveal a clear performance hierarchy: the Daily Memory model exhibits the highest error spikes, followed by moderate improvements in the Weekday-Aware and Windowed Context models, with the Multi-Agent model achieving the most consistent and lowest MAPE. These observations support the quantitative findings in Table 1; simple memory models struggle with dynamics and regime shifts, while the Multi-Agent framework delivers robust and adaptive performance across both datasets, achieving the lowest MAPE and MAE, confirming its superior adaptability in dynamic process environments.

4.5.2. Quantitative Performance Comparison

Table 1 presents the quantitative evaluation of all models across the Helpdesk and BPIC13 Incidents datasets using MAPE and MAE. The benchmark models-drawn from previous studies-primarily report MAPE and generally fall in the range of 2.3–3.1% on both datasets, with Persistent Forecast performing best on the BPIC13 Incidents dataset.

Among our proposed methods, the Fusion Predictor consistently outperforms the others, achieving the lowest MAPE and MAE across both datasets. On the BPIC13 Incidents dataset, it reduces the MAPE to 1.50%, outperforming all benchmarks and baselines. It also achieves the lowest MAE (9.45), indicating stable predictions even during sharp fluctuations. For the Helpdesk dataset, while all our methods yield slightly higher MAPE than the benchmarks, the Fusion model still delivers the best performance within our framework (MAPE = 2.91, MAE = 2.90). Notably, the LSTM model-despite being a deep learning based approach-performs poorly on the Helpdesk dataset (MAPE = 34.15), likely due to data sparsity or lack of temporal consistency. These results highlight the Fusion Predictor's ability to generalize across dynamic process regimes, leveraging multiple contextual retrieval strategies and predictive agents to achieve superior adaptability, as evidenced by its consistently low error rates in both stable and volatile conditions.

Table 1. Prediction Results (MAPE and MAE) for Helpdesk and BPIC13 Datasets

Model	**Helpdesk**		**BPIC13 Incidents**	
	MAPE	**MAE**	**MAPE**	**MAE**
Benchmarks				
TCN#1	2.35	–	1.53	–
TCN#2	2.39	–	2.94	–
TCN#3	2.45	–	3.14	–
LSTM	34.15	–	2.99	–
Persistent Forecast	2.65	–	0.86	–
Proposed Methods				
Daily Memory Predictor	4.34	2.62	1.59	15.93
Weekday-Aware Predictor	3.13	3.29	1.55	13.55
Windowed Context Predictor	3.19	3.08	1.66	9.56
Multi-Agent Predictor	2.91	2.90	1.50	9.45

Overall, these results confirm the benefit of combining multiple contextual retrieval strategies and predictive agents. While simple models such as Daily Memory or Persistent Forecast can

achieve reasonable performance under stable conditions, they struggle in volatile regions. In contrast, the Fusion Predictor generalizes better across dynamic process regimes, as supported by both numerical and visual evaluations.

5. CONCLUSION AND FUTURE WORK

We proposed a novel framework for predictive process monitoring that integrates RAG with a multi-agent architecture to predict WiP. By converting structured event logs into natural language stories and deploying agents focused on distinct temporal views, the framework enhances predictive accuracy and interoperability through semantic reasoning. Key advantages include dynamic, context-aware retrieval via a process memory, extensibility through agent-based modularity, and robust decision fusion that adapts to process trends. Experiments on real-world datasets validate the effectiveness of the approach, highlighting its competitiveness and the benefit of combining symbolic and neural reasoning.

Future directions include incorporating adaptive retrieval mechanisms that optimize memory usage over time, enabling interactive forecasting through user feedback, and extending the architecture to support additional predictive tasks such as cycle time estimation, anomaly detection, and compliance monitoring. Integration with domain-specific knowledge graphs may also enhance the transparency and reliability of the predictions.

REFERENCES

[1] Ceravolo, Paolo, Comuzzi, Marco, De Weerdt, Jochen, Di Francescomarino, Chiara, Maggi, FabrizioMaria, Predictive process monitoring: concepts, challenges, and future research directions, Process Science, vol. 1, no. 1, Springer, 2024, pp. 2.

[2] Rama-Maneiro, Efr´en, Deep learning for predictive business process monitoring: Review and benchmark, IEEE Transactions on Services Computing, vol. 16, no. 1, IEEE, 2021, pp. 739-756.

[3] Wang, Jiaxing, Lu, Chengliang, Cao, Bin, Fan, Jing, MiTFM: A multi-view information fusion method based on transformer for Next Activity Prediction of Business Processes, in: Proceedings of the 14th Asia-Pacific Symposium on Internetware, 2023, pp. 281-291.

[4] Tax, Niek, Verenich, Ilya, La Rosa, Marcello, Dumas, Marlon, Predictive business process monitoring with LSTM neural networks, in: Proceedings of International Conference on Advanced Information Systems Engineering, 2017, pp. 477-492.

[5] Nguyen, An, Chatterjee, Srijeet, Weinzierl, Sven, Schwinn, Leo, Matzner, Martin, Eskofier, Bjoern, Time Matters: Time-Aware LSTMs for Predictive Business Process Monitoring, in: Process Mining Workshops, 2021, pp. 112-123.

[6] Pasquadibisceglie, Vincenzo, Appice, Annalisa, Malerba, Donato, Lupin: A llm approach for activity suffix prediction in business process event logs, in: Proceedings of 2024 6th International Conference on Process Mining (ICPM), 2024, pp. 1-8.

[7] Oved, Alon, Shlomov, Segev, Zeltyn, Sergey, Mashkif, Nir, Yaeli, Avi, SNAP: Semantic Stories for Next Activity Prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025.

[8] Cheng, Mingyue, Luo, Yucong, Ouyang, Jie, Liu, Qi, Liu, Huijie, Li, Li, Yu, Shuo, Zhang, Bohou, Cao,Jiawei, Ma, Jie, et al., A survey on knowledge-oriented retrieval-augmented generation, arXiv preprint arXiv:2503.10677, 2025.

[9] Gruver, Nate, Finzi, Marc, Qiu, Shikai, Wilson, Andrew G, Large language models are zero-shot timeseries forecasters, Advances in Neural Information Processing Systems, vol. 36, 2023, pp. 19622-19635.

[10] Berti, Alessandro, van der Aalst, Wil M. P., CrewAI: Modular LLM Agents for Process Analytics, in: CAiSE Workshops, 2024.

[11] Aratchige, R. M., Ilmini, W. M. K. S., LLMs Working in Harmony: A Survey on the Technological Aspects of Building Effective LLM-Based Multi-Agent Systems, arXiv preprint arXiv:2405.12345, 2025.

[12] Xu, Yifei, Fang, Huan, Next timestamp prediction in business process monitoring using large language models, in: Proceedings of Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024), vol. 13550, 2025, pp. 1381-1388.
[13] Jessen, Urszula, Sroka, Michal, Fahland, Dirk, Chit-Chat or Deep Talk: Prompt Engineering for Process Mining, arXiv preprint arXiv:2307.09909, 2023.
[14] Mehrdad Bibalan, Yousef, Far, Behrouz, Eshragh, Faezeh, Ghiyasian, Bahareh, Work in Progress Prediction for Business Processes Using Temporal Convolutional Networks, in: Advances and Trends in Artificial Intelligence. Theory and Applications (IEA/AIE 2024), 2024, pp. 109-121.
[15] Berti, Alessandro, Kourani, Humam, van der Aalst, Wil MP, PM-LLM-Benchmark: Evaluating large language models on process mining tasks, in: International Conference on Process Mining, 2024, pp. 610-623.
[16] Beheshti, Amin, Yang, Jian, Sheng, Quan Z, Benatallah, Boualem, Casati, Fabio, Dustdar, Schahram, Nezhad, Hamid Reza Motahari, Zhang, Xuyun, Xue, Shan, ProcessGPT: transforming business process management with generative artificial intelligence, in: 2023 IEEE International Conference on Web Services (ICWS), 2023, pp. 731-739.
[17] Berti, Alessandro, Schuster, Daniel, van der Aalst, Wil MP, Abstractions, scenarios, and prompt definitions for process mining with LLMs: A case study, in: Proceedings of International Conference on Business Process Management, 2023, pp. 427-439.
[18] Redis, Andrei Cosmin, Sani, Mohammadreza Fani, Zarrin, Bahram, Burattin, Andrea, Skill Learning Using Process Mining for Large Language Model Plan Generation, in: International Conference on Process Mining, 2024, pp. 650-662.
[19] Liu, Jerry, LlamaIndex, 2022.
[20] Mehrdad Bibalan, Yousef, Far, Behrouz, Moshirpour, Mohammad, Ghiyasian, Bahareh, Using Meta Learning to Predict Work-in-Progress: An Approach for Small Datasets, Proceedings of IEA/AIE, 2025.
[21] Xu, Junjielong, Zhang, Qinan, Zhong, Zhiqing, He, Shilin, Zhang, Chaoyun, Lin, Qingwei, Pei, Dan, He,Pinjia, Zhang, Dongmei, Zhang, Qi, OpenRCA: Can Large Language Models Locate the Root Cause of Software Failures?, in: The Thirteenth International Conference on Learning Representations, 2025.
[22] Evermann, Joerg, Rehse, Jan-Reiner, Fettke, Peter, Predictive business process monitoring with LSTMs, Business Process Management Journal, vol. 23, no. 3, Emerald Publishing Limited, 2017, pp. 361-383.
[23] Rama-Maneiro, Efr´en, Encoder-Decoder Model for Suffix Prediction in Predictive Monitoring, arXiv preprint arXiv:2211.16106, 2022.
[24] Alessandro Berti, Mahnaz Sadat Qafari, Leveraging Large Language Models (LLMs) for Process Mining (Technical Report), arXiv preprint arXiv:2307.12701, 2023.
[25] DeGrandis, Dominica, Making Work Visible: Exposing Time Theft to Optimize Work & Flow, IT Revolution, 2022.
[26] Anderson, David J, Carmichael, Andy, Essential Kanban Condensed, Blue Hole Press, 2016.
[27] Jachmann, Thomas, Transforming a large medical organization towards speed and flow, in: Proceedings of 2019 IEEE/ACM 1st International Workshop on Software Engineering for Healthcare (SEH), 2019, pp. 17-20.
[28] Jessen, Urszula, Sroka, Michal, Fahland, Dirk, Chit-chat or deep talk: prompt engineering for process mining, arXiv preprint arXiv:2307.09909, 2023.
[29] Estrada-Torres, Bedilia, del R´ıo-Ortega, Adela, Resinas, Manuel, Mapping the landscape: Exploring large language model applications in business process management, in: Proceedings of International Conference on Business Process Modeling, Development and Support, 2024, pp. 22-31.
[30] Oved, Alon, Shlomov, Segev, Zeltyn, Sergey, Mashkif, Nir, Yaeli, Avi, SNAP: semantic stories for next activity prediction, arXiv preprint arXiv:2401.15621, 2024.
[31] Rebmann, Adrian, Schmidt, Fabian David, Glava˘s, Goran, van Der Aa, Han, Evaluating the ability of LLMs to solve semantics-aware process mining tasks, in: Proceedings of 2024 6th International Conference on Process Mining (ICPM), 2024, pp. 9-16.
[32] Berti, Alessandro, Kourani, Humam, Ha¨fke, Hannes, Li, Chiao-Yun, Schuster, Daniel, Evaluating large language models in process mining: Capabilities, benchmarks, and evaluation strategies, in: Proceedings of International Conference on Business Process Modeling, Development and Support, 2024, pp. 13-21.

[33] Xiao, Mengxi, Jiang, Zihao, Qian, Lingfei, Chen, Zhengyu, He, Yueru, Xu, Yijing, Jiang, Yuecheng,Li, Dong, Weng, Ruey-Ling, Peng, Min, et al., Enhancing Financial Time-Series Forecasting with Retrieval-Augmented Large Language Models, arXiv e-prints, 2025, pp. arXiv-2502.