Process-Aware Prediction from Event Logs using Machine Learning

ZSOMBOR IVANYI, University of Twente, The Netherlands

ABSTRACT - The increasing availability of detailed event logs from industrial and service processes offer new opportunities to support operational decision-making through predictive analytics. This work aims to investigate how predictive models trained on historical event logs perform under changing process conditions, such as disruptions, and to assess their generalization in logistics environments. An empirical study based on a logistics and manufacturing dataset is conducted, evaluating the accuracy and robustness of Long Short-Term Memory (LSTM) models in different scenarios, including process disruptions. The findings highlight not only which ML models perform best under various conditions, but also how predictive values can potentially augment logs to support proactive decision making in operational environments such as logistics and healthcare. This has the potential to enable downstream use in process discovery, conformance checking, and process enhancement.

Additional Key Words and Phrases: Process Mining, Predictive Process Monitoring, Machine Learning, Event Logs, Operational Decision Support

1 INTRODUCTION

The increasing digitalization of industrial and service processes has led to the widespread adoption of process mining techniques, which use event logs to discover, monitor, and enhance operational workflows [5, 14]. In domains such as logistics, manufacturing, and healthcare, event logs record the fine-grained execution of processes, often capturing timestamps, case identifiers, and activity names across heterogeneous systems. These logs offer a valuable source of information not only for retrospective analysis but also for forward-looking predictive insights [6].

Predictive Process Monitoring (PPM) extends traditional process mining by using historical event data to forecast future aspects of ongoing cases [11], such as the remaining cycle time, next activity, or likelihood of deviations. Such predictions are increasingly used to support time-sensitive decision making, mitigate process inefficiencies, and anticipate disruptions (e.g., [9, 12]). For instance, in logistics settings, forecasting the impact of vehicle unavailability or scheduling delays can help reallocate resources or prioritize urgent shipments. In healthcare, similar predictions can inform triage decisions or scheduling adjustments early on [16].

Despite promising advances, most existing predictive models are trained and evaluated under static conditions, assuming process stability. In practice, operational environments are dynamic and often subject to disruptions, such as equipment breakdowns, workload shifts, or unplanned reconfigurations, that challenge model robustness and generalizability.

This study investigates how machine learning models, particularly Long Short-Term Memory (LSTM) networks, perform under both normal and disrupted operational scenarios using simulated event logs from logistics and manufacturing domains.

Although prior research (see Section 3) has focused predominantly on prediction accuracy under static conditions, this work contributes a broader perspective by assessing model behavior under disruptions and highlighting integration points for predictive outputs within the process mining lifecycle. Beyond prediction, these predictive outputs could potentially be integrated directly into event logs to support downstream process mining tasks such as process discovery, conformance, and enhancement.

The remainder of this paper is organized as follows. Section 3 reviews related work on predictive process monitoring and machine learning approaches. Section 4 introduces the dataset and outlines the characteristics of the simulated disruption scenarios. Section 5 presents the modeling approach, including feature engineering, model design, and evaluation setup. Section 6 describes the experimental setup used. Section 7 reports on the empirical results, comparing predictive performance under normal and disrupted conditions. Finally, Section 8 discusses key insights, limitations, and opportunities for future work, and Section 9 concludes the study.

2 RESEARCH DESIGN

2.1 Research Objectives and Scope

The primary objective of this research is to evaluate the accuracy and generalization capability of predictive models trained on event logs, especially when applied under dynamic, non-standard process conditions. By simulating operational disruptions such as the removal of transport vehicles, an assessment is made about whether predictive models can maintain performance and deliver actionable insights even when the environment shifts away from its training distribution.

2.2 Research Questions

To guide this investigation, we define the following main research question (RQ):

- **RQ:** To what extent can machine learning models generalize their predictions of process outcomes under changing operational conditions (e.g., disruptions)?

To support this primary inquiry, two sub-questions (SQ) are also formulated:

TScIT 43, July 4, 2025, Enschede, The Netherlands

- **SQ1:** Which features extracted from event logs are most informative for predicting process outcomes such as remaining cycle time?
- **SQ2:** What are the practical implications and limitations of using predictive values to support operational decision-making?

2.3 Experimental Strategy

An empirical case study was conducted using a simulated logistics and manufacturing dataset [1] that includes both normal and disrupted process scenarios. An LSTM model is trained on event logs from standard settings and evaluated under both familiar and perturbed process conditions. Predictive accuracy is assessed using regression metrics, and additional qualitative analysis is used to interpret underestimation and overestimation patterns. This enables an investigation into both predictive performance and generalizability of the model across scenarios.

3 RELATED WORK

Process Mining (PM) has grown in the past decade and is a relatively new technique to analyze and improve processes. The main goal is to discover, monitor and improve real processes using event logs generated by the system itself [13]. However, PM techniques are useful for processes that already finished (i.e., post-mortem event logs) [10]. From this foundation Predictive Process Monitoring (PPM) emerged.

PPM combines the strengths of PM and machine learning to forecast process attributes [11] such as the next activity, remaining time, or the likelihood of non-compliance. In the work by Maggi et al. [11], supervised learning techniques were applied to runtime event streams to support early prediction of outcomes, introducing a new class of real-time, value-adding process mining tools. Since then, a rich body of literature has explored the application of regression, classification, and deep learning models to predict remaining cycle time and other operational variables [4, 10].

Remaining cycle time is especially relevant in manufacturing and logistics domains, where accurate estimates enable better planning, throughput optimization, and just-in-time resource allocation. Recent case studies (e.g., Friederich et al. [10]) benchmark several regression-based models, including Decision Trees (DT), Random Forests (RF), XGBoost (XGB), and k-Nearest Neighbors (KNN), on cycle time prediction tasks in automated production settings. These models vary in interpretability, scalability, and computational cost (see Table 1), but are typically trained and evaluated on static process conditions.

Table 1. Regression models comparison

Model	Attributes
DT	Good interpretability of results
RF, XGB	Powerful ensemble machine learning
KNN	Simple but large computation cost

Alongside classical machine learning approaches, deep learning techniques, particularly Recurrent Neural Networks (RNNs) and LSTM networks, have been proposed for capturing temporal dependencies in sequential event data. LSTM-based models have shown promise in tasks such as next activity prediction and time-to-completion estimation [8]. These models are particularly attractive for handling complex, high-variance processes, though at the cost of interpretability and training time.

While most existing studies focus on prediction accuracy under stable process conditions, relatively few explore how predictive models behave under process disruptions or concept drift. Initial explorations of robustness under drift are emerging in the literature [3], yet evaluations under disruption scenarios (e.g., resource unavailability, dynamic rerouting) remain sparse.

In this thesis, the focus is on a logistics and manufacturing environment in which product quality degrades over time [7]. Every product experiences a decline in quality, a process that can be accelerated or slowed by various internal factors within the system. However, this thesis specifically focuses on the remaining cycle time in the system, as it is closely related to the level of quality decay [2].

4 DATASET AND SIMULATION ENVIRONMENT

To evaluate the proposed predictive modeling approach, a dataset [1] corresponding to the study reported in [2] is used. The dataset was generated using a discrete event simulation model.

The factory layout of the generated dataset is shown in Figure 1.

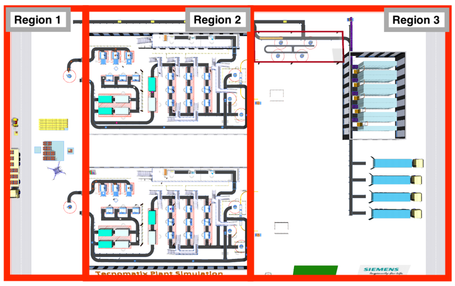

Fig. 1. Factory layout for simulated logistics-manufacturing operations.

4.1 Source and Composition of Event Logs

Each event log records a series of activities for individual process cases, capturing information such as event type, timestamps, the vehicle involved, and product quality decay. Table 2 provides an example of log entries. Every entry includes a unique Event ID, a Case ID linking related events, a categorical event label (e.g., PickUp, Transport, Deliver), start and end timestamps, the vehicle executing the task, and a decay score representing product quality at that moment. The event flow pipeline is visualized in Figure 1.

The dataset comprises 27 simulation scenarios, each executed 20 times, resulting in a diverse and extensive collection of event logs. Scenarios are defined by varying vehicle compositions (see Table 3) and dispatching rules. Vehicle types

Table 2. Example event log entries showcasing the most important features (representative values).

Event ID	Case ID	Event	Start Time	End Time	Vehicle	Decay
001	A	PickUp	08:00	08:05	UAV:1	100
002	A	Transport	08:05	08:20	UAV:1	97,13
003	A	Deliver	08:20	08:25	UAV:1	85,55
004	B	PickUp	08:10	08:15	HDF:1	100
005	B	Transport	08:15	08:30	HDF:1	98,24
006	B	Deliver	08:30	08:35	HDF:1	94,76

include Unmanned Aerial Vehicles (UAVs), Human-Driven Forklifts (HDFs), and Automated Guided Vehicles (AGVs).

Table 3. Vehicles used in experiments

Exp. Num.	Vehicles used
1–9	3 UAVs; 1 HDF; 1 AGV
10–18	3 UAVs; 2 HDFs; 2 AGVs
19–27	2 HDFs; 2 AGVs

4.2 Disruption Scenarios

To simulate dynamic operational conditions, disruption scenarios are mimicked into the dataset. These scenarios represent interruptions such as vehicle failures, removals, or additions, which alter the standard execution flow of cases. For example, in one scenario, an HDF and an AGV are removed from the system, leading to reassignment of transport tasks to other vehicles. This simulates capacity loss and introduces variation in cycle times and decay trajectories.

By comparing model performance on baseline (undisturbed) and disrupted scenarios, an assessment can be made of how well trained models generalize under real-world variability. This setup enables controlled experimentation around key research themes such as robustness, transferability, and concept drift in predictive process monitoring.

4.3 Feature Engineering and Preprocessing

Before model training, event logs are preprocessed into case-based traces suitable for supervised learning. The following steps are applied:

- **Trace structuring:** each case is transformed into a sequence of events, preserving timestamp order and including intermediate state values (e.g., decay).
- **Feature extraction:** features include event duration, inter-arrival time, event position in trace, vehicle type, and decay. Categorical features are one-hot encoded, and continuous values are normalized.
- **Label assignment:** for each partial trace, the remaining cycle time is calculated relative to the final event in the case.

Missing values are imputed using backward-fill (post-padding) strategies, and sequences are padded to a fixed length to accommodate batch processing in LSTM models. An example of a process model that could be obtained is shown in Figure 2.

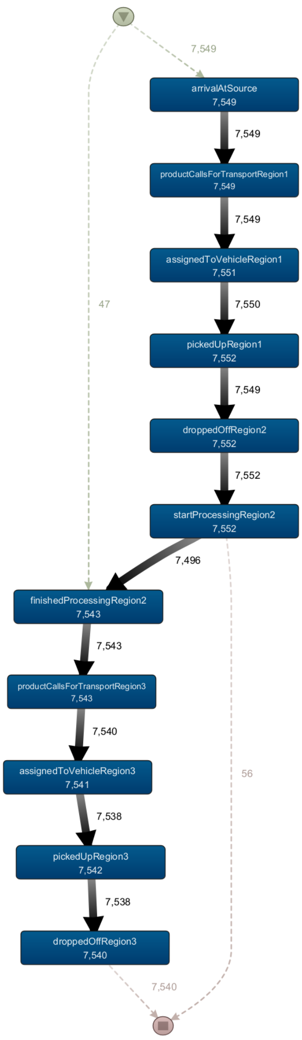

Fig. 2. Event flow in single simulated run execution.

5 MODELING APPROACH

This section outlines the architecture, training configuration, and evaluation methodology used to develop predictive models for estimating remaining cycle time in process executions. The primary focus is on an LSTM network, chosen for its suitability in modeling sequential dependencies in event log data.

5.1 Model Architecture (LSTM and Baselines)

The predictive model implemented in this study is an LSTM network, which is a specialized type of RNN designed to handle long-range dependencies in time series and sequential data. LSTM models have been widely used in predictive process monitoring tasks for their ability to capture temporal patterns across event traces [8].

The LSTM model architecture comprises the following components:

- An input masking layer to handle padded values in variable-length traces;
- A single LSTM layer with 50 hidden units, configured to return sequences;
- A dense output layer with one neuron and a linear activation function to perform regression on remaining cycle time.

In addition to the LSTM model, baseline comparisons were considered using traditional regression models, including RF and XGB, which have demonstrated strong performance in prior PPM studies [10]. However, given project constraints and the sequential nature of the task, the emphasis remained on optimizing and evaluating the LSTM-based solution.

5.2 Model Training and Hyperparameters

The model was implemented using the Keras API within TensorFlow. The training process used the Adam optimizer, a widely adopted adaptive learning algorithm, with the following hyperparameters:

- **Batch size:** 64
- **Loss function:** Mean Absolute Error (MAE)
- **Validation split:** 10% of the training data

Data scaling was applied to all numeric features, and categorical features (e.g., vehicle type) were one-hot encoded.

5.3 Evaluation Metrics and Strategy

The performance of the model was evaluated using two widely accepted regression metrics:

- **Mean Absolute Error (MAE):** Measures the average magnitude of errors in predictions, offering high interpretability.
- **Mean Squared Error (MSE):** Captures error variance and penalizes large deviations more severely than MAE.

To assess generalizability, the model was trained on event logs from non-disrupted scenarios and tested on both regular and disrupted traces. This allowed us to analyze how performance degrades (or persists) under process variability. Comparative evaluations in vehicle removal provided insight into the robustness of the LSTM architecture.

5.4 Modeling Tools and Implementation Setup

All model development and evaluation activities were conducted in Python, primarily within the Jupyter Notebook environment. The following libraries were used:

- **TensorFlow/Keras:** For deep learning model construction and training.
- **pm4py:** For event log preprocessing and conversion into case-based sequences suitable for ML input.
- **Scikit-learn:** For baseline model implementation and metric computation.
- **Matplotlib:** For visualizing model performance and comparative error plots.

Figure 3 illustrates the overall data flow from event logs to prediction and downstream process integration.

6 EXPERIMENTAL SETUP

This section outlines how the predictive modeling experiments were designed and executed to evaluate the robustness of the model under stable and disrupted operational conditions.

6.1 Train-Test Split and Disruption Strategy

To simulate realistic training and evaluation scenarios, the structure of the simulation dataset was used, which includes 27 experiments varying in vehicle composition (Table 3).

- **Training and Testing (Non-disrupted):** Experiments 10–18, each comprising 20 simulation runs, were used to train and evaluate the model under stable conditions. Data from 18 runs were used for training, and the remaining 2 runs per experiment were reserved for testing.
- **Disruption Scenario Evaluation:** Experiments 1–9 were used to evaluate the generalization of the model under disruptions, where one HDF and one AGV are removed compared to the training composition.

This configuration allows us to test the model's ability to maintain predictive accuracy when applied to unfamiliar and capacity-reduced scenarios.

6.2 Feature Extraction and Preprocessing

Event logs were parsed and transformed into case-based sequences using `pm4py` and `pandas`. Each sequence includes engineered features such as:

- Event duration and position in trace
- Inter-arrival times and vehicle type
- Product quality decay (numeric)

Categorical features were encoded and numeric features normalized. Remaining cycle time was used as the regression target and computed relative to the final event in each case. Padded sequences ensured fixed input lengths for the LSTM model, accommodating flow variability and early terminations.

6.3 Model Training and Data Volume

Training was carried out in batch mode, combining all event traces from 180 runs (10–18) into a unified training and testing set. Each file contains approximately $100,000$ event logs, resulting in a substantial training volume. This batch setup was preferred over incremental file-wise training for stability and convergence speed.

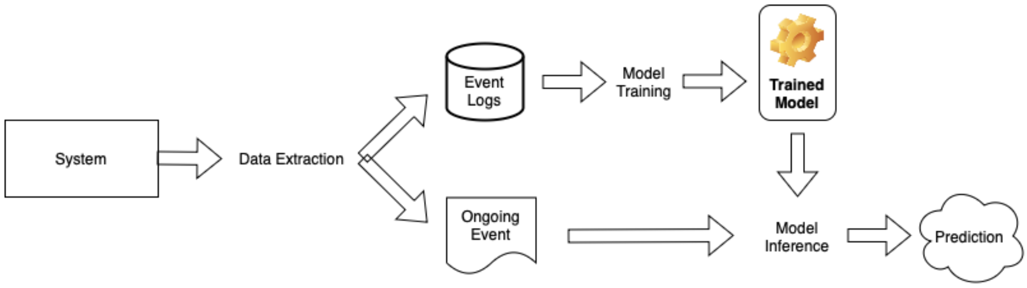

Fig. 3. Prediction using machine learning models in the PPM lifecycle.

6.4 Evaluation Metrics and Interpretation

Predictive performance was assessed using MAE) — for average prediction deviation
Root Mean Squared Error (RMSE) — for penalizing larger errors

Scatter plots and summary tables (e.g., Table 4) complement these metrics by visualizing error distributions across normal and disrupted settings. Classification metrics were not used, as the prediction task is continuous in nature.

	Vehicles			**Target**	
	UAV	**HDF**	**AGV**	**AMAE**	**ARMSE**
Normal	3	2	2	20 s	33 s
Disruption	3	1	1	32 s	49 s

Table 4. Average MAE and RMSE under normal vs. disrupted vehicle configurations.

This setup enables systematic investigation of predictive robustness and highlights how operational changes (e.g., vehicle removal) affect model accuracy.

7 EXPERIMENTAL RESULTS

The final goal is to show that machine learning can be used to predict disruption outcomes in logistics processes in a way that helps decision makers act earlier and more effectively, especially when the system is too complex to establish rule-based fail safes solely.

7.1 Prediction Performance under Stable Conditions

The model was trained on 162 files and tested on 18 files. The average MAE and RMSE values were calculated from the test file results, and Table 5 shows the predictions of the remaining cycle time.

Table 5. Baseline prediction error metrics

Metric	Value
MAE	19.85 s
RMSE	32.73 s

Meaning, the average absolute difference between the predicted and actual remaining cycle time is around 20 seconds, while if larger errors are taken more heavily then it is around 33 seconds. Scatter plots showcase the accuracy, where each spike represents a flow and each sample represents an event on figure 4. On figure 16 individual flows can be seen on each plot to see how accurately the model can predict the reamining time and capture trends in the data.

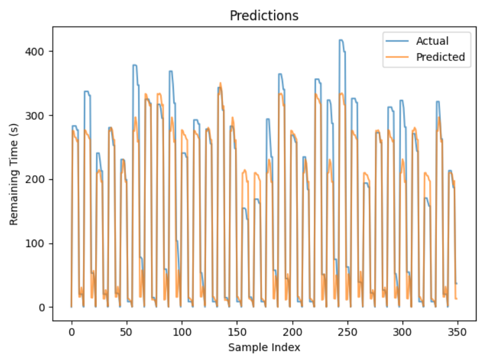

Fig. 4. Results snippet of remaining cycle time predictions of events, testing on similar vehicle composition as training data.

This shows that the model predicts somewhat precisely with data that is familiar to its algorithm.

7.2 Generalization to Disruptions

However, when it comes to testing the model on data where the vehicle composition changed (number of HDFs and AGVs decreased from 2 to 1), the results are shown in Table 6.

Table 6. Disruption prediction error metrics

Metric	Value
MAE	32.04 s
RMSE	49.49 s

It is visibly worse by approximately 10-20 seconds (See Table 4) than testing the data with the same vehicle composition as its training data. But this does not mean that the model breaks, since it is clearly visible on figure 5 and 17 that the model still predicts the correct trends, just with worse accuracy.

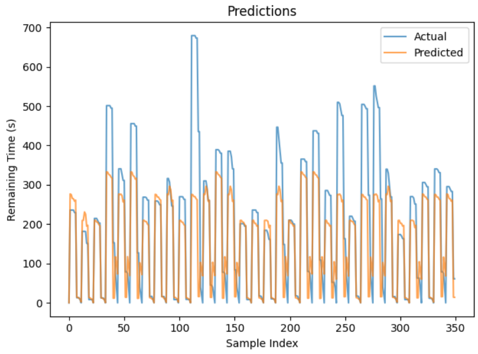

Fig. 5. Results snippet of remaining cycle time predictions of events, testing on disruption files where vehicle composition is different than training data.

7.3 Error Analysis and Failure Cases

Error analysis markers were incorporated into figure 4. and 5., resulting in enhanced figures 6. and 7. In these figures, red points indicate events where the model predicted a non-zero remaining cycle time, whereas the actual remaining cycle time was zero. Green points highlight instances where the absolute difference between the predicted and actual values exceeds 100 seconds. In particular, these larger errors occur more frequently when the model is tested on data that differs from the data it was trained on, illustrating the efficiency of the model in the event of a disruption.

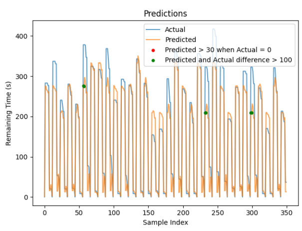

Fig. 6. Baseline Figure 4, with error analysis markers.

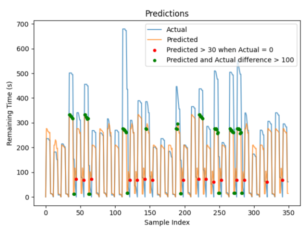

Fig. 7. Disruption Figure 5, with error analysis markers.

7.4 Error Distribution Analysis

In figures 8. and 9. both distribution error plots are centered around 0, which means that the model is generally unbiased in both cases.

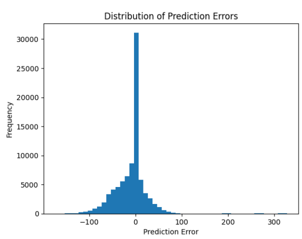

Fig. 8. Distribution of prediction errors at a normal test

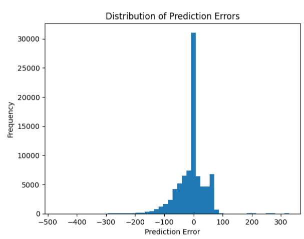

Fig. 9. Distribution of prediction errors at a disruption test

However, figures 10. and 9. are more informative in the sense of underestimation and overestimation. The model underestimates the high remaining times in both normal and disruption scenarios, but this effect is stronger in the disruption test. The prediction accuracy is better for lower actual values in both cases. The disruption test shows larger and more scattered errors, indicating that the model generalizes less well to disrupted or out-of-distribution data.

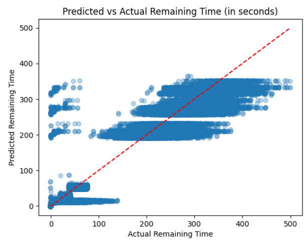

Fig. 10. Predicted vs. Actual Scatter Plot at a normal test

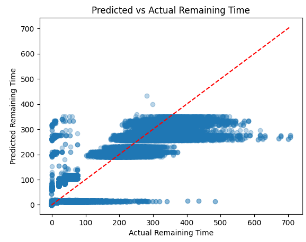

Fig. 11. Predicted vs. Actual Scatter Plot at a disruption test

Overall, the model is reasonably accurate for low remaining times but tends to underestimate when the actual remaining time is high, especially under disruption.

7.5 Special Cases

7.5.1 Rare Overestimation at Normal Testing. As discussed earlier, the model tends to underestimate the remaining cycle time. However, on rare occasions an overestimation can happen (See Figure 12). This flow may have unusual values for features (e.g., current decay level, event types, vehicle, etc.) that are rare or not well represented in the training data. The model may not have learned to generalize well for such rare combinations, leading to overestimation.

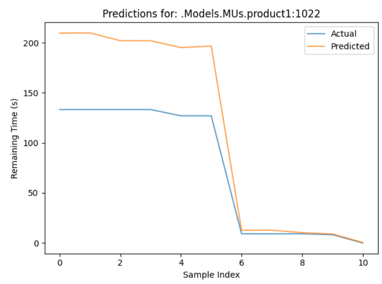

Fig. 12. Rare overestimated flow.

7.5.2 Rare Highly Underestimation at Disruption Testing. During testing on disruption data, results occasionally reveal significant underestimation errors, as an example is illustrated in Figure 13. This pattern primarily stems from the model's lack of exposure to disrupted vehicle compositions during training, as previously discussed.

However, when a further detailed analysis of the trends of product decay is performed, product quality can be closely connected to the characteristics of the model in its prediction nature. In figure 14 shows an abnormal value drop

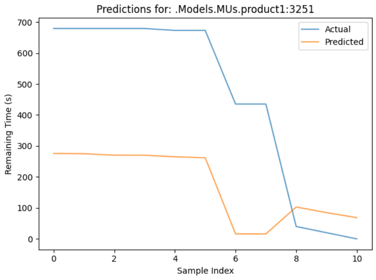

Fig. 13. Rare highly underestimated flow from disruption data.

from one event to another, which likely contributes to the observed almost 400-second underestimation in remaining time prediction. This abrupt decay pattern contrasts with the gradually decreasing decay progression over multiple events visible in normal operational flows (Figure 15). It is also important to note that flows where the product decay drops below 60 after concluding its journey over the stations are discarded.

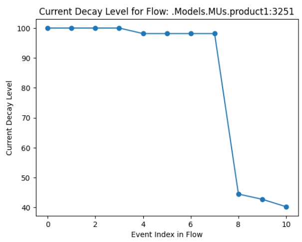

Fig. 14. Product decay decline (from 100 to 40) over events in the same flow as Figure 14.

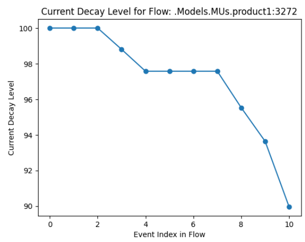

Fig. 15. Normal product decay decline (from 100 to 90) over events in another flow.

8 DISCUSSION

This study set out to evaluate the robustness of predictive process monitoring models under process disruptions. LSTM networks enabled the analysis of predictive performance in dynamic settings using simulated event logs.

8.1 Key Findings

The main contribution lies in demonstrating that LSTM models can maintain reasonable predictive accuracy even when applied to unseen and disrupted process configurations. Trained on stable scenarios, the model generalized to altered vehicle compositions, showing trend-aligned predictions despite increased error margins. This finding underscores the potential for predictive models to provide operational foresight even in the absence of disruption-specific training data.

8.2 Limitations

Two main limitations emerged. First, the model exhibited a consistent tendency to underestimate longer remaining times, likely due to a training distribution skewed toward shorter durations and the use of symmetric loss functions (MAE/MSE). These functions are known to bias predictions to the mean, particularly in the presence of rare or extreme values [15]. Second, the study tested only a single type of disruption (vehicle removal), limiting the scope of generalizability. Other forms of variability, such as speed changes or routing faults, remain unexplored. Lastly, as the data is simulation-based, external validity is constrained, though the simulation design closely mirrors real logistics environments.

8.3 Implications for Decision Support

The results hold practical relevance for predictive decision-making in logistics and manufacturing. By embedding predictive outputs into operational workflows, organizations can detect the early onset of disruptions, assess alternative scenarios, and make proactive adjustments. For instance, predictive cycle times under vehicle shortages can inform scheduling strategies or resource reallocation.

Moreover, such models can serve as the basis for what-if simulations, enabling the anticipation of disruption effects before they materialize, supporting both planning and resilience in volatile environments.

8.4 Future Work

Future research should expand the range of tested disruptions, integrate real-world event logs, and explore adaptive models capable of learning from concept drift. Incorporating weighted [15] or asymmetric loss functions may further improve predictions for rare events. A promising direction involves embedding predictive outputs into event logs for downstream tasks such as process enhancement and conformance checking—realizing the vision of prediction-as-input in the process mining lifecycle.

9 CONCLUSION

This study demonstrates that machine learning models, particularly LSTM networks, can effectively predict remaining cycle times from event logs, even under previously unseen disruptions in operational environments. While accuracy diminished under shift conditions, models retain sufficient predictive power to potentially support early warning and proactive decision making. Prediction errors increase when test data differs from the training data, especially for rare or high remaining time values.

By shifting the focus from static to dynamic process settings, this work contributes to a more resilient and adaptive vision of PPM. Limitations related to scope and data realism provide opportunities for future research, including more varied types of disruption, real-world deployments, and integration into process-aware decision support systems. Ultimately, this work underscores the potential of predictive analytics to extend the value of event logs beyond monitoring, toward intelligent, forward-looking process control in complex environments.

A APPENDIX: USE OF AI TOOLS

Throughout this paper, artificial intelligence tools were utilized to improve clarity and flow, while ensuring that the originality and integrity of the ideas remained intact. The entire manuscript was carefully revised and refined by Zsombor Ivanyi, who takes complete responsibility for the final content.

B APPENDIX: ADDITIONAL PLOTS

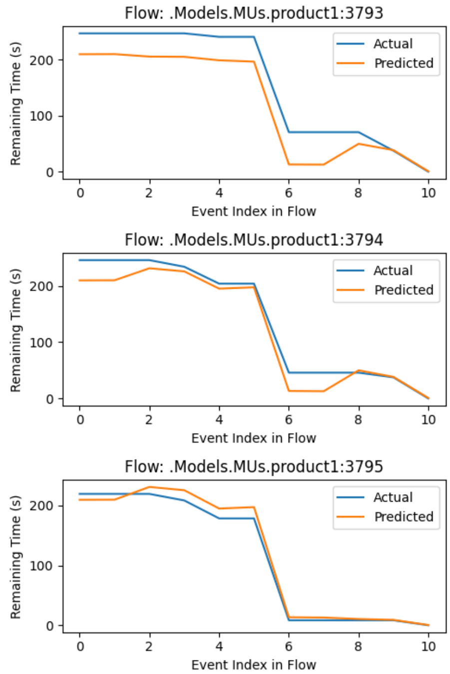

Fig. 16. Results of remaining cycle time predictions at 3 specific flows, testing on similar vehicle composition as training data.

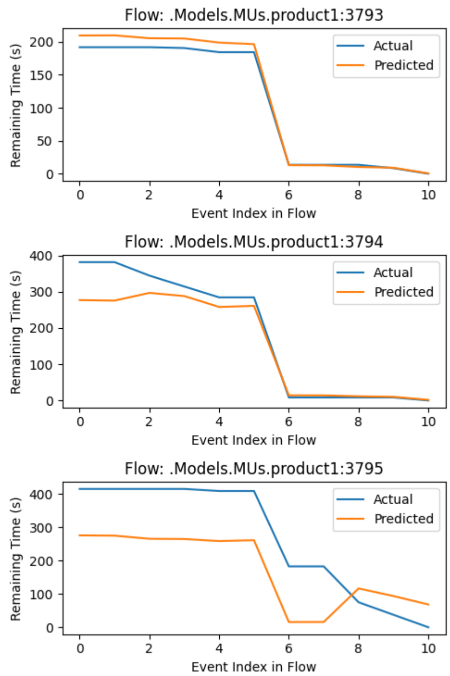

Fig. 17. Results of remaining cycle time predictions at 3 specific flows, testing on disruption files where vehicle composition is different than training data.

REFERENCES

[1] R. Bemthuis, M.R.K. Mes, M.E. Iacob, and P.J.M. Havinga. 2021. Data underlying the paper: Using agent-based simulation for emergent behavior detection in cyber-physical systems. https://doi.org/10.4121/14743263.v1 Dataset.

[2] Rob Bemthuis, Martijn Mes, Maria-Eugenia Iacob, and Paul Havinga. 2020. Using agent-based simulation for emergent behavior detection in cyber-physical systems. In *2020 Winter Simulation Conference (WSC)*. IEEE, 230–241.

[3] Paolo Ceravolo, Marco Comuzzi, Jochen De Weerdt, Chiara Di Francescomarino, and Fabrizio Maria Maggi. 2024. Predictive process monitoring: concepts, challenges, and future research directions. *Process Science* 1, 1 (2024), 2.

[4] Alexandre Checoli Choueiri, Denise Maria Vecino Sato, Edson Emilio Scalabrin, and Eduardo Alves Portela Santos. 2020. An extended model for remaining time prediction in manufacturing systems using process mining. *Journal of Manufacturing Systems* 56 (2020), 188–201.

[5] Jonathan E Cook and Alexander L Wolf. 1998. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 7, 3 (1998), 215–249.

[6] Dusanka Dakic, Darko Stefanovic, Teodora Lolic, Dajana Narandzic, and Nenad Simeunovic. 2020. Event log extraction for the purpose of process mining: a systematic literature review. In *Innovation in Sustainable Management and Entrepreneurship: 2019 International Symposium in Management (SIM2019)*. Springer, 299–312.

[7] Marlies de Keizer, Renzo Akkerman, Martin Grunow, Jacqueline M Bloemhof, Rene Haijema, and Jack GAJ van der Vorst. 2017. Logistics network design for perishable products with heterogeneous quality decay. *European journal of operational research* 262, 2 (2017), 535–549.

[8] Chiara Di Francescomarino and Chiara Ghidini. 2022. Predictive process monitoring. In *Process mining handbook*. Springer International Publishing Cham, 320–346.

[9] Chiara Di Francescomarino, Chiara Ghidini, Fabrizio Maria Maggi, and Fredrik Milani. 2018. Predictive process monitoring methods: Which one suits me best?. In *International conference on business process management*. Springer, 462–479.

[10] Jonas Friederich, Jonas Kristoffer Lindeløv, and Sanja Lazarova-Molnar. 2023. Predictive process monitoring for prediction of remaining cycle time in automated manufacturing: a case study. In *2023 IEEE 28th international conference on emerging technologies and factory automation (ETFA)*. IEEE, 1–8.

[11] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini. 2014. Predictive monitoring of business processes. In *Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings 26*. Springer, 457–472.

[12] Alfonso Eduardo Márquez-Chamorro, Manuel Resinas, and Antonio Ruiz-Cortés. 2017. Predictive monitoring of business processes: a survey. *IEEE Transactions on Services Computing* 11, 6 (2017), 962–977.

[13] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter Van Den Brand, Ronald Brandtjen, Joos Buijs, et al. 2012. Process mining manifesto. In *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I 9*. Springer, 169–194.

[14] Wil MP Van der Aalst and Anton JMM Weijters. 2004. Process mining: a research agenda. *Computers in industry* 53, 3 (2004), 231–244.

[15] Ziyang Wang, Masahiro Mae, Takeshi Yamane, Masato Ajisaka, Tatsuya Nakata, and Ryuji Matsuhashi. 2024. Novel Custom Loss Functions and Metrics for Reinforced Forecasting of High and Low Day-Ahead Electricity Prices Using Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) and Ensemble Learning. *Energies* 17, 19 (2024), 4885.

[16] Yuxuan Zhang. 2019. *Event log mining on sepsis clinical path: classification and regression study*. Master's thesis. Eindhoven University of Technology. https://research.tue.nl/en/studentTheses/event-log-mining-on-sepsis-clinical-path