

The Impact of Event Log Noise on Predictive Process Monitoring Performance

Sinan Salman

Associate Professor, Systems Engineering
Zayed University, Abu Dhabi, UAE
sinan.salman@zu.ac.ae

Abstract

Predictive Process Monitoring (PPM) is a discipline within business process mining that leverages historical process event logs to predict the outcome of ongoing process instances. This capability enables organizations to transition toward proactive decision-making, optimizing their operations. However, the practical application of PPM is significantly hindered by the inherent noisiness of real-world process event logs. Such noise can result from deviations from standard business process definitions, infrequent process activities or paths, data recording errors, or an imbalance in the representation of process case variants in the log. This could negatively impact the performance and stability of machine learning models used in PPM, as atypical entries can prevent the models from effectively learning the underlying process, leading to poor predictive accuracy and limited generalization when deployed in live environments. This paper investigates the impact of event log noise on PPM performance. A PPM machine learning algorithm is trained on multiple, increasingly filtered versions of a real-world business process event log. Each version represents an attempt to systematically exclude noisy event log entries to achieve different levels of denoising. The experimental results quantitatively demonstrate the significant influence of event log data quality on the efficacy of machine learning models, showing that targeted denoising of the event log training data directly enhances the model's predictive performance and robustness. This study underscores the importance of pre-processing and noise mitigation as a crucial step in developing reliable and generalizable PPM solutions.

Keywords

Process Denoising, Predictive Process Monitoring, Business Process Mining, and Machine Learning.

1. Introduction

Predictive Process Monitoring (PPM) is a subfield of Process Mining (PM) that uses historical process event log data to forecast the future behavior of instances of ongoing business processes. As such, it is considered a predictive analytical approach that helps practitioners forecast next activities, process outcomes, or process completion times. This enables proactive decision-making, resource allocation, and performance optimization in business processes. However, inherent event log noise in real-world event logs can reduce the performance and stability of PPM Machine Learning (ML) models, eroding the benefits of PPM deployments in complex real-world environments.

Event log noise is typically the result of various factors, including deviations from standard business processes, infrequent activities, event log errors, and imbalances in case variants. A case variant is a unique path or sequence of activities within a business process followed by one or more business process cases. These issues can prevent the model from achieving good accuracy and generalization for practical use.

The impact of event log noise on process understanding can be visually demonstrated through the filtering process practitioners use during PM analysis. Figure 1 shows several Directly Follows Graphs (DFGs) for the business process event log used in this paper. Each DFG demonstrates a different level of filtering out infrequent process variants.

Of course, for human comprehension, removing more of the atypical cases renders the process more visible and understandable. However, this comes at the cost of excluding valid infrequent variants from the analysis and, in doing so, of losing some of the nuances of the process definition. While the impact on human comprehension of business processes is clear, ML models generally outperform humans in handling complex datasets and patterns; however, noise can still affect ML performance.

Therefore, this paper attempts to answer the following research question: What is the impact of event log noise reduction on PPM prediction accuracy? This is accomplished by training a PPM ML algorithm on multiple, increasingly filtered versions of a real-world business process event log, each representing an attempt to exclude more of the potentially noisy entries to assess its impact on the ML model's accuracy.

The rest of the paper is organized as follows. Section 2 briefly reviews related literature. Section 3 outlines the research methodology, and Section 4 presents the results and major findings. Finally, Section 5 concludes the paper and points to future work.

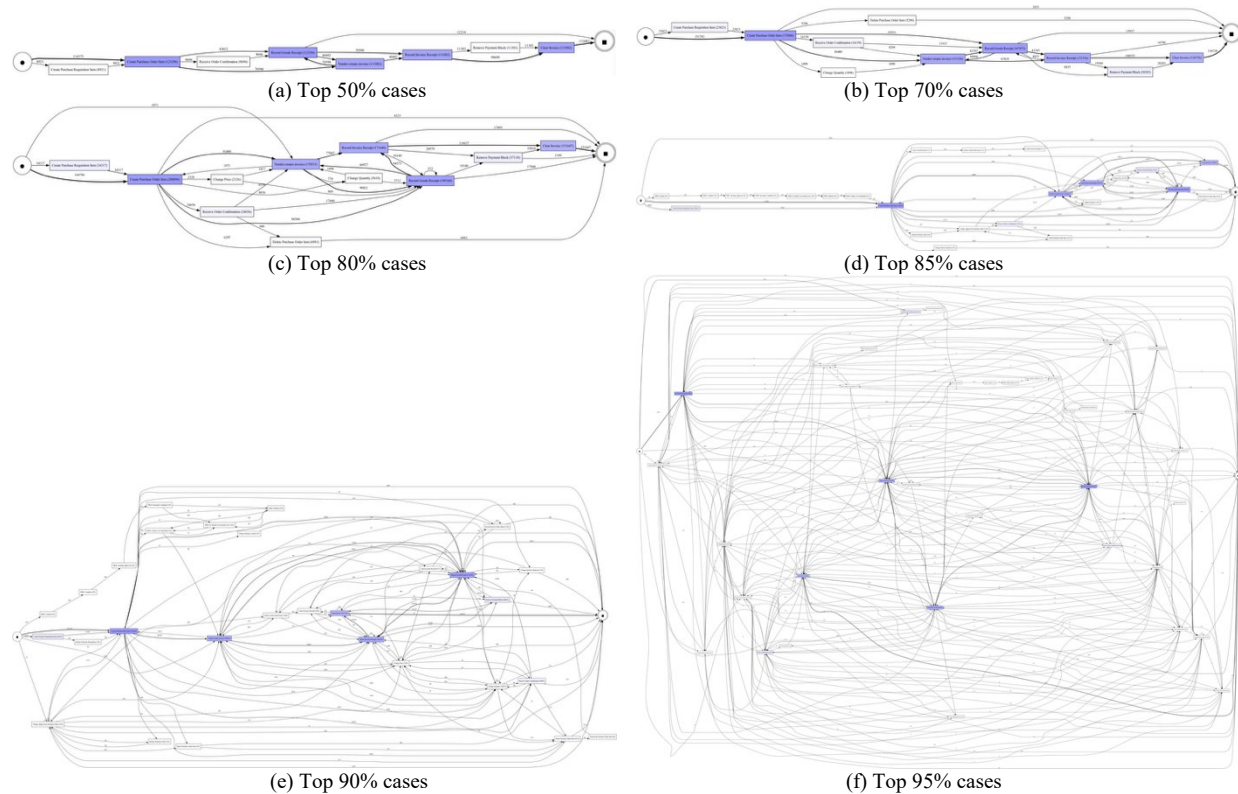


Figure 1. Directly Follow Graphs (DFG) after filtering the event log file for the variants representing the top $x\%$ cases (full-size diagrams available at GitHub¹)

2. Literature Review

Several studies in the literature have explored noise in event logs, their types, and their potential impact on process analysis. This section briefly reviews the literature and relevant work on the use of ML in PPM.

Infrequent activities are the most prevalent types of event log outliers that may be considered as data noise, depending on their relevance to the standard business process definition. PPM models often struggle with activities that rarely appear in the event log, leading to poor performance due to underrepresentation in the training data (Käppel et al., 2021). Similarly, redundant, insufficient, or Incomplete data in the event log, where some events or attributes are

¹ <https://github.com/SinanSalman/EventLog-de-noising-for-PPM>

missing, can also hinder the training of ML models that require large amounts of data to generalize well (Peeperkorn et al., 2023).

Spurious correlations are another source of noise in event logs. In such cases, changes in real-world processes, such as unauthorized alterations to process models or cross-departmental customizations, can introduce spurious biases and correlations into the data. These can adversely impact the accuracy of predictive models. Similarly, concept drift, gradual changes in the process over time, can shift the data distribution, leading to decreased model accuracy (Venkateswaran et al., 2021). Additionally, imbalanced target variables, where certain outcomes or activities occur significantly more frequently than others, can lead to instability in model performance. This is particularly problematic for deep learning techniques (Kratsch et al., 2021).

The above types of event log noise can significantly impact the PPM model performance. Several approaches have been proposed in the literature to identify noise and outliers in event logs. These methods can be broadly categorized into three categories. Frequency-based filtering excludes infrequent paths and activities based on their frequency of occurrence (Mannhardt et al., 2017). Data-aware techniques identify outliers by detecting unusual behavior patterns (Zhang et al., 2021). Anomaly detection can be achieved using Graph Autoencoders (Huo et al., 2021).

This paper uses a frequency-based filtering approach to investigate the impact of denoising an event log on PPM performance when predicting the next process activity. It uses the Long Short-Term Memory (LSTM) neural network, which has been shown to perform well in PPM applications. LSTMs are capable of learning long-term dependencies, which are crucial for predicting the next activity based on previous activities in a process execution path. LSTMs were first introduced by Hochreiter & Schmidhuber (1997) and subsequently applied to PPM (Navarin et al., 2017; Peeperkorn et al., 2023; Tax et al., 2017). Incorporating LSTM time-aware mechanisms to account for time dependencies between events has been credited with improving prediction accuracy in PPM applications (Nguyen et al., 2021).

3. Methodology

The research approach utilized here is to incrementally denoise an event log by filtering out infrequent process variants and to observe the corresponding effect on an LSTM model's next-activity prediction accuracy. The steps below are performed on each increasingly denoised event log, starting with no denoising and progressing to filtering out the least frequent 50% of cases. The underlying hypothesis is that removing statistically rare cases—which often represent noise, outliers, or exceptional behavior—will expose the model to a more coherent underlying process, thereby improving its predictive performance. For each denoising level, the following steps are executed:

1. *Event Log Filtering:* The raw event log is loaded and processed. Starting with the full event log (100% of cases), the log is iteratively filtered to retain the most frequent process variants, which collectively make up the target threshold. This threshold is decreased from 99% down to 50% in steps of 1%, discarding progressively more low-frequency variants.
2. *Sequence and Target Generation:* The filtered event log is transformed into a format suitable for sequential deep learning. For each case, a set of prefix-target pairs is generated to facilitate predicting the next activity. For example, consider a sequence of activities [A, B, C, D]. The generated pairs are: ([A], B), ([A, B], C), and ([A, B, C], D). A limit of 10 activities is set as the maximum prefix length. This data augmentation step is critical, as it enables the LSTM model to learn from and generalize to partially completed traces, mimicking real-world predictive monitoring scenarios.
3. *Model Training and Evaluation:* The dataset of prefix-target pairs is split into a training set (80%) and a hold-out test set (20%). A standard LSTM model architecture is implemented and trained on the training set. The model's predictive accuracy is then quantitatively evaluated on the unseen test set by comparing the predicted next activity to the ground-truth target for each prefix.

Variant-frequency filtering is chosen as the denoising method due to its efficacy in categorizing process execution pathways at scale based on their prevalence in the event log. This enables bulk elimination of divergent execution paths that would otherwise increase the challenge for the ML model to learn from the process.

The LSTM model is implemented with an architecture tailored to the sequential nature of the event log data. The input layer consists of 42 nodes, corresponding to the vocabulary size of unique activities in the process. This is followed

by an embedding layer with 32 nodes to capture activity representations in a lower-dimensional space, and a hidden LSTM layer with 64 units to learn temporal dependencies across event sequences.

The model is trained using the ADAM optimizer with a learning rate of 0.005, employing CrossEntropy as the loss function for multi-class next-activity prediction. Training is conducted over 10 epochs with a batch size of 32 to balance computational efficiency and gradient stability. Validation is performed by comparing the model's predictions against the ground-truth labels in the test set, and predictive accuracy is calculated as the percentage of correctly forecasted next activities.

The experiments are conducted using the publicly available event log from the 9th International Business Process Intelligence Challenge (BPI Challenge 2019) (van Dongen, 2019). This log documents the purchase-order handling process for a multinational coatings and paints corporation, including its 60 subsidiaries. The event log represents a complex, real-world business process with inherent variability and noise. The following are key event log file statistics: 42 activities, 251,734 cases, 1,595,923 events, and 11,973 variants.

The process mining library PM4Py is used to load the event log in XES format and perform variant-frequency-based filtering. The PyTorch framework is used to construct, train, and evaluate the LSTM neural network model. The experiment code is available at a supplementary repository².

4. Results

Before diving into the experimental results, it is important to visualize the relationship between the filtering threshold for case variants and the coverage of cases in the event log. Figure 2 illustrates the exponential relationship between the top n case variants and event log coverage. In the event log used here, only the top 45 out of the 11,973 variants are required to achieve an 80% coverage, and 1,425 variants are required to achieve 95% coverage of the cases found in the log. This means that approximately 88% of the variants only account for 5% of the event log data. This is typical in many real-world business processes, where a few highly frequent case variants account for the majority of the event log, while a large number of infrequent case variants make up the remainder of the log.

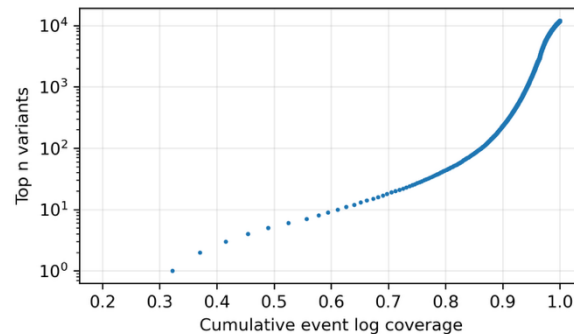


Figure 2. Relationship between the number of variants and case coverage

Figures 3a and 4a illustrate the impact of case variant filtering on the LSTM model's performance. The figures show that the LSTM next-activity model's prediction accuracy is significantly affected by the case variant filtering process. As more variant cases (i.e., potential noise) are included in the training dataset, the prediction accuracy decreases from around 86% (with a 50% filtering threshold) to around 74% (with no filtering), a 12 percentage-point drop.

This result suggests that careful preprocessing can significantly enhance model accuracy and deployability in complex real-world processes. However, caution should be used when determining the appropriate filtering threshold. Overzealous filtering could result in the exclusion of infrequent yet valid case variants, leaving the process definition incomplete. This may lead to higher prediction accuracy, but poorer model generalization as infrequent execution

² Code available at <https://github.com/SinanSalman/EventLog-de-noising-for-PPM> for reproducibility

paths are overlooked. The investigation of trade-offs between model accuracy and generalization is identified as a future research direction for this work, as discussed in the conclusion.

Although the filtered dataset produced higher next-activity prediction accuracy, these results should not be interpreted as evidence of universally better generalization. In the BPI Challenge 2019 log, a very small number of highly frequent variants account for most cases, while thousands of low-frequency variants represent a small but potentially meaningful portion of real process behavior. Consequently, frequency-based filtering may improve performance by simplifying the learning problem and reducing noise, but it may also reduce the model's exposure to legitimate but uncommon execution paths. A model trained on heavily filtered data may therefore perform well on test data drawn from the same dominant distribution, while being less robust when deployed in settings where rare yet valid variants occur. This highlights the distinction between higher in-sample predictive accuracy and broader operational generalization across the full diversity of process executions.

As expected, when fewer data points are filtered out, the training set size increases, requiring more training time, as shown in Figure 3b. This is also confirmed by the number of events included in the training dataset, as seen in Figure 4a. Interestingly, filtering out the least frequent 4% (leaving 96% of the cases) results in the most rapid decline in training time (approximately 25%), due to the large number of associated events and case variants.

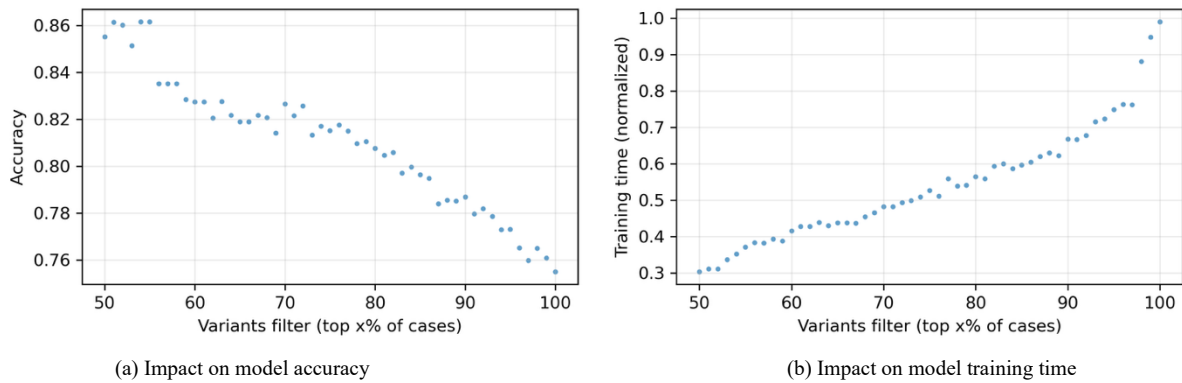


Figure 3. Impact of variants filter (x % of cases coverage) on the LSTM model accuracy and training time

Finally, Figure 4b shows an exponential relationship between the number of case variants in the training set and the model's prediction accuracy. Together with Figure 2, this demonstrates that a small number of case variants can cover the majority of cases in the event log, and that using this focused training set can produce a more accurate LSTM model for next-event predictions.

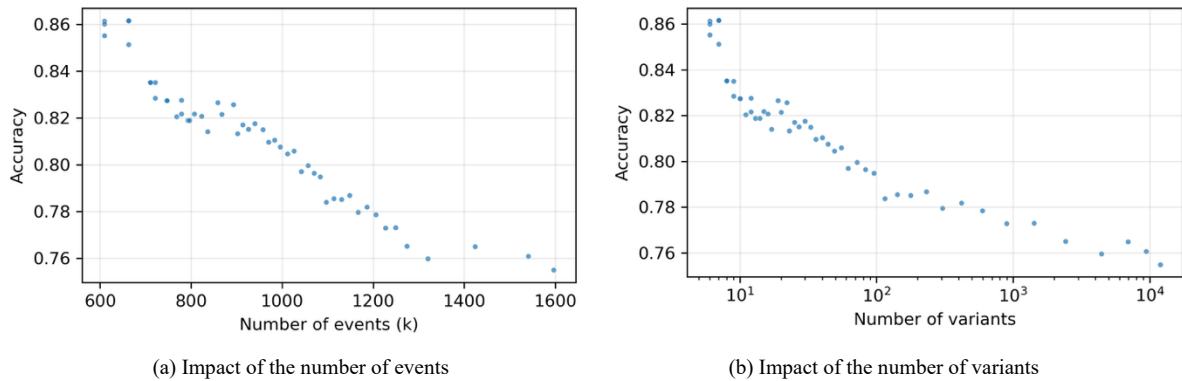


Figure 4. Impact of the number of events and variants included in the data set on the LSTM model accuracy

5. Conclusion

To evaluate the effect of event log noise on PPM accuracy, this work applies an incremental denoising approach to an LSTM ML model. By iteratively filtering out more of the least frequent process variants, we test the hypothesis that reducing event log noise could improve an LSTM model's ability to learn the core process and accurately forecast the next activity.

The results show that event log filtering can significantly impact the accuracy of next-event prediction using an LSTM model. The results show a 12% improvement in prediction accuracy when filtering to the top 50% most frequent case variants. This suggests that careful preprocessing can significantly enhance model deployability. Training time requirements are also impacted due to the training dataset size; however, filtering out the least frequent 4% of the case variants significantly reduces the number of associated events and cases, which consequently reduces the training time requirements more steeply (about 25% reduction).

Finally, the results demonstrate an exponential relationship between the number of case variants included in the training set and the model's prediction accuracy. It shows that limiting the training set to a small number of the top case variants can produce a more accurate LSTM model for next-event predictions.

While this study demonstrates the impact of noise filtering on a publicly available real-world log, further validation across domains and log types is still needed. Additionally, our filtering approach focused solely on variant frequency; future work should integrate multi-dimensional noise detection (e.g., temporal outliers, attribute inconsistencies).

While the results showed improvement in model prediction accuracy, an important question remains: How much of the filtered-out data represents valid process case variants, and how much is excluded to achieve higher accuracy? Put differently, how do we determine the signal-to-noise ratio at each filtration threshold, and where to draw the line to identify true noise from process diversity case variants in the event log? Answering this question could lead to the development of an adaptive auto-filtering approach.

Future work should therefore explicitly evaluate generalization by testing models trained at different filtering thresholds on datasets that retain a broader spectrum of infrequent but valid variants. Such analysis would help determine whether the observed gains arise from true noise reduction or from narrowing the prediction task to dominant process behaviors. This would also support the development of principled criteria for selecting a filtering threshold that balances predictive accuracy, robustness, and fidelity to the underlying business process.

There are also several limitations of this study that should be addressed in future studies, including the use of a single dataset and the focus on next-activity prediction. Furthermore, additional venues for analysis could include examining mispredicted prefix-target pairs to uncover patterns that could help avoid such errors.

References

- Hochreiter, S. and Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Huo, S., Völzer, H., Reddy, P., Agarwal, P., Isahagian, V., and Muthusamy, V., Graph Autoencoders for Business Process Anomaly Detection, in A. Polyvyanyy, M. T. Wynn, A. Van Looy, and M. Reichert (Eds.), *Business Process Management*, Springer International Publishing, pp. 417–433, 2021. https://doi.org/10.1007/978-3-030-85469-0_26
- Käppel, M., Jablonski, S., and Schöning, S., Cost-Sensitive Predictive Business Process Monitoring, in L. Bellatreche, M. Dumas, P. Karras, R. Matulevičius, A. Awad, M. Weidlich, M. Ivanović, and O. Hartig (Eds.), *New Trends in Database and Information Systems*, Springer International Publishing, pp. 14–26, 2021. https://doi.org/10.1007/978-3-030-85082-1_2
- Kratsch, W., Manderscheid, J., Röglinger, M., and Seyfried, J., Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction, *Business & Information Systems Engineering*, vol. 63, no. 3, pp. 261–276, 2021. <https://doi.org/10.1007/s12599-020-00645-0>
- Mannhardt, F., de Leoni, M., Reijers, H. A., and van der Aalst, W. M. P., Data-Driven Process Discovery—Revealing Conditional Infrequent Behavior from Event Logs, in E. Dubois and K. Pohl (Eds.), *Advanced Information Systems Engineering*, Springer International Publishing, pp. 545–560, 2017. https://doi.org/10.1007/978-3-319-59536-8_34

- Navarin, N., Vincenzi, B., Polato, M., and Sperduti, A., LSTM networks for data-aware remaining time prediction of business process instances, *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, 2017. <https://doi.org/10.1109/SSCI.2017.8285184>
- Nguyen, A., Chatterjee, S., Weinzierl, S., Schwinn, L., Matzner, M., and Eskofier, B., Time Matters: Time-Aware LSTMs for Predictive Business Process Monitoring, *Lecture Notes in Business Information Processing*, vol. 406, pp. 112–123, 2021. https://doi.org/10.1007/978-3-030-72693-5_9
- Peeperkorn, J., vanden Broucke, S., and De Weerd, J., Can recurrent neural networks learn process model structure?, *Journal of Intelligent Information Systems*, vol. 61, no. 1, pp. 27–51, 2023. <https://doi.org/10.1007/s10844-022-00765-x>
- Tax, N., Verenich, I., La Rosa, M., and Dumas, M., Predictive business process monitoring with LSTM neural networks, *International Conference on Advanced Information Systems Engineering*, pp. 477–492, 2017.
- van Dongen, B. F., Dataset BPI Challenge 2019, *4TU.Centre for Research Data*, <https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>, 2019.
- Venkateswaran, P., Muthusamy, V., Isahagian, V., and Venkatasubramanian, N., Robust and Generalizable Predictive Models for Business Processes, in A. Polyvyanyy, M. T. Wynn, A. Van Looy, and M. Reichert (Eds.), *Business Process Management*, vol. 12875, Springer International Publishing, pp. 105–122, 2021. https://doi.org/10.1007/978-3-030-85469-0_9
- Zhang, Z., Hildebrandt, R., Asgarinejad, F., Venkatasubramanian, N., and Ren, S., Improving Process Discovery Results by Filtering Outliers from Event Logs with Hidden Markov Models, *2021 IEEE 23rd Conference on Business Informatics (CBI)*, vol. 1, pp. 171–180, 2021. <https://doi.org/10.1109/CBI52690.2021.00028>

Biography

Sinan Salman is an associate dean and associate professor at the College of Technological Innovation at Zayed University. He received his Ph.D. and M.Sc. in Industrial Engineering from the University of Arkansas and Oklahoma State University, respectively, and a B.Sc. in Mechanical Engineering from Jordan University of Science and Technology. Before joining Zayed University in 2014, he worked in the US for 13 years in Supply Chain Management at Merck & Co. and Transplace (now Uber Freight). His professional career encompassed global supply chain and transportation management, logistics consulting, and the development of optimization software. His areas of interest include operations research and its applications in decision support, as well as modeling, optimization, and simulation of complex systems, particularly in supply chain management and transportation. He is also interested in the use of cutting-edge technology in higher education. He has authored over 20 publications in top journals and conferences in his field. Dr. Salman is a member of the Phi Kappa Phi (PKP) National Honor Society, Alpha Pi Mu (APM) Industrial Engineering Honor Society, Institute of Industrial and Systems Engineers (IISE), Institute for Operations Research and the Management Sciences (INFORMS), and received multiple certificates and awards, including Certified Supply Chain Professional (CSCP), Lean Six Sigma Black Belt (LSS-BB), Certified Project Manager (CPM), IBM's Artificial Intelligence mastery, and the outstanding teaching award from Zayed University. He is also the creator of the open-source X-Supply Game (XSG), an educational online multiplayer simulation designed to introduce students to supply chain management.