Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

FROM THE PAST TO THE FUTURE: 10 YEARS OF DISCRETE-EVENT SIMULATION AND MACHINE LEARNING THROUGH A SYSTEMATIC REVIEW OF WSC PROCEEDINGS

Yanina Andrea Döning[1] and María Julia Blas[1,2]

[1]Facultad Regional Santa Fe, Universidad Tecnológica Nacional, Santa Fe, ARGENTINA
[2]Instituto de Desarrollo y Diseño INGAR, CONICET-UTN, Santa Fe, ARGENTINA

ABSTRACT

Over the past few years, the interest in Machine Learning (ML) has grown due to its ability to improve solutions related to other fields. This paper explores the use of ML techniques in simulation through a systematic literature review of the Winter Simulation Conference proceedings from 2013 to 2023. Our research is focused on the Discrete-Event Simulation (DES) field, centering our attention on the Discrete-Event System Specification (DEVS) formalism as a particular case. The research questions were designed to examine the most frequent contexts, applications, methods, and software tools used in these studies. As a result, this review reports insights into 44 research studies. The main contribution of this paper is related to systematically gathering, analyzing, and discussing the knowledge disseminated in these two areas (ML and DES), aiming to support future research and expand the literature in this field.

1 INTRODUCTION

Literature review is an essential feature of academic research (Xiao and Watson 2017). Since knowledge advancement must be built, by reviewing relevant literature following a strict review protocol, we can improve our understanding of the existing body of knowledge related to a specific field and, therefore, identify gaps to be explored.

A Systematic Literature Review (SLR) is a methodology to generate the theoretical-scientific basis needed to understand a topic through the collection, understanding, synthesis, and evaluation of a set of scientific articles (Levy and Ellis 2006). This paper presents an SLR designed to identify the Machine Learning (ML) techniques used in the simulation field to improve the design and deployment of Discrete-Event Simulation (DES) models. Even though the scope of our research is focused on DES, we decided to segregate Discrete-Event System Specification (DEVS) studies as a particular case. By following this scope, research questions were designed to examine the most frequent contexts, applications, methods, and software tools used. Such questions were defined as follows (1) What are the main purposes of using ML in DES projects? (2) What are the main issues in DES solved with DEVS and ML? (3) What are the ML techniques/tools most frequently used to solve DES problems? (4) What are the results obtained for DES/DEVS projects? (5) To what areas do the results obtained in DES/DEVS projects refer? (6) Can these results be transferred to other situations?

Our review analyzes papers published in the Winter Simulation Conference (WSC) proceedings over the past 10 years. WSC is considered one of the leading events in the Modeling and Simulation (M&S) field, providing valuable insights into both theoretical and practical aspects of M&S and helping to shape the future of simulation industry. Aiming to provide a background suitable to support the next 25 years (as the theme of WSC'25), we believe our review provides an understanding of the latest advances in DES theory and applications related to ML techniques. Then, we show the current state-of-the-art on the subject, identifying boundaries and gaps in existing studies and providing guidelines for future works. As a result, this review reports insights into 44 research studies. The main contribution of this paper is related to systematically gathering, analyzing, and discussing the knowledge disseminated in these two areas (ML and DES), aiming to support future research and expand the literature in this field.

The remainder of this paper is structured as follows. Section 2 presents the research method used to support the SLR by describing how such a method was performed. Section 3 summarizes and discusses the results obtained during the reviewing process. Finally, Section 4 is devoted to conclusions and directions to be addressed in future research. Due to space reasons, considering the target audience is the M&S community, we decided not to include an overview of the DES/DEVS literature. Regarding the ML literature, we use concepts and basic notions through the sections.

2 RESEARCH METHODOLOGY

Denver and Tranfield (2009) indicate that an SLR should not be interpreted as a Literature Review. Indeed, an SLR is a research project that uses literature to respond to questions where all steps are well-defined and can be reproduced. Typically, these steps are defined in three stages: *(1) planning the review*, *(2) conducting the review*, and *(3) reporting the review* (Breretona et al. 2007; Kitchenham and Charters 2007). The *planning of the review* stage is devoted to defining the problem to be addressed by limiting its scope using a set of research questions. Once formulated, these questions are addressed in *conducting the review* stage through the analysis and synthesis of the research articles collected. Finally, at the *reporting of the review* stage, results and conclusions to the research questions are stated through descriptive statistics.

This paper is based on the review process discussed in (Xiao and Watson 2017). From the SLR perspective, the objective was defined as *i)* explore how DES/DEVS and ML have been collaboratively used in the past 10 years of the WSC, *ii)* analyze and synthesize the findings by answering a set of well-defined research questions, and *iii)* discuss the results by identifying boundaries and gaps along with guidelines to be addressed in the future. The following sections describe how each step of stages 1 and 2 was conducted to perform the reviewing process. Then, Section 3 presents the results obtained at stage 3. The data supporting these results is available in Appendix A.

2.1 Planning the Review

2.1.1 Formulate the Problem (Step 1)

Research questions drive the entire literature review process (Kitchenham and Charters 2007). According to Cronin et al. (2008), a common mistake is selecting a research question that is too broad. To avoid any issues related to the definition of research questions, we decided to define our research questions based on the Context, Intervention, Mechanism, and Outcomes (CIMO) framework (Denyer and Tranfield 2008).

The CIMO framework allows structuring and analyzing practical problems and solutions by helping to understand how different elements interact in a given context. By designing propositions using CIMO, several aspects of the research problem can be addressed as *i) Context*: Conditions or environment in which the problem occurs; *ii) Intervention*: Action, change, or strategy implemented to address the problem; *iii) Mechanism*: The underlying processes explaining how the intervention works to produce the desired outcomes; and *iv) Outcome*: The results or effects of the intervention. Following these guidelines, Table 1 summarizes the CIMO research questions used to guide the review.

2.1.2 Develop and Validate the Review Protocol (Step 2)

The review protocol to be employed for the systematic review was fully validated (goals, research questions, inclusion criteria, search strategies, quality assessment criteria, and so on) to keep the study on track. This step is essential to ensure a successful revision of the articles with sound results and conclusions.

2.2 Conducting the Review

2.2.1 Search the Literature (Step 3)

The quality of a systematic literature review is highly dependent on the literature collected for the review (Xiao and Watson 2017). Electronic databases are commonly used to collect the set of potential papers to be reviewed. Such collection is obtained by an advanced search executed using a well-defined string in a search engine. Table 2 presents the string employed in our search, explaining each part as a statement related to the goals of the review.

The advanced search was performed on the IEEE Xplore database on April 24, 2024. Such a search was configured to match the string with all the metadata and full text of articles to find as many results as possible. The meta-search for articles resulted in 1971 articles.

Table 1: Research questions designed using CIMO framework.

#	CIMO	Research Question
1	Context	What are the main purposes of using ML in DES projects?
2	Intervention	What are the main issues in DES solved with DEVS and ML?
3	Mechanism	What are the ML techniques/tools most frequently used to solve DES problems?
4	Outcome	What are the results obtained for DES/DEVS projects?
5	Outcome	To what areas do the results obtained in DES/DEVS projects refer?
6	Outcome	Can these results be transferred to other situations?

Table 2: String used in an advanced search over the IEEE Xplore database.

Full String	
("Discrete Event Simulation" OR "DEVS" OR "Discrete Event System Specification" OR "Discrete-Event System Specification") AND ("Machine Learning" OR "ML" OR "Artificial intelligence" OR "AI" OR "Computational Intelligence" OR "Deep Learning") AND ("goal" OR "application" OR "problem" OR "algorithm")	
Part	**Explanation**
("Discrete Event Simulation" OR "DEVS" OR "Discrete Event System Specification" OR "Discrete-Event System Specification")	Abbreviations and alternative spellings were defined for the search terms "Discrete Event Simulation" or "DEVS".
("Machine Learning" OR "ML" OR "Artificial intelligence" OR "AI" OR "Computational Intelligence" OR "Deep Learning")	Abbreviations, alternative spellings, and related terms were defined for the search term "Machine Learning".
("goal" OR "application" OR "problem" OR "algorithm")	Part defined to collect data linked to the research questions.

2.2.2 Screen for Inclusion (Step 4)

The screening step was conducted considering the following inclusion criteria: *(a)* the paper must be a full article published in the Proceedings of the Winter Simulation Conference; *(b)* the paper was published, at most, 10 years ago (i.e., 2013 to 2023); *(c)* have the full article available in IEEE Explore or WSC Archive; and *(d)* the article must not be a review, survey, or discussion paper. As a result, 1837 articles were excluded from the search (i.e., 134 articles were considered for quality assessment, as shown in Figure 1a).

2.2.3 Assess Quality (Step 5)

In this step, the researchers evaluated each remaining article to analyze its quality for future data extraction. Additional inclusion criteria were defined: *(e)* the paper must be focused on DES/DEVS and ML/AI, and *(f)* the paper must enunciate the ML paradigm or technique employed as complementary of DEVS/DES.

As a result, 90 papers were excluded since they failed to meet at least one of the criteria *(e)* and *(f)*. For example, 12 articles were excluded since they use the term "ML" in a discrete-event simulation model as an abbreviation of a term not related to machine learning (e.g., machine line and modeling language). Figure 1b summarizes the decisions made regarding article exclusions, showing that 44 of the 134 articles were selected for data extraction (Appendix A).

2.2.4 Extracting, Analyzing, and Synthesizing Data (Steps 6 and 7)

Figure 1c shows the interest in the proposed subject has stood out in the past 10 years of WSC. The data confirms a growing trendline on DES and ML as complementary techniques to address several problems.

Figure 1: (a) On the left side, the pie chart shows the results of the screening for inclusion step. (b) In the center, the bar chart shows the number of articles excluded due to the quality assessment step (pink bars). The green bar is the number of articles selected for data extraction. (c) On the right side, annual scientific production related to DES/DEVS and ML/AI in WSC from 2013 to 2023 (considering only the articles selected for data extraction). The dashed yellow line depicts the trendline.

The 44 articles were evaluated on 11 items divided into two categories (*features* and *methods*) to answer the research questions. The *features* category investigates the main features of articles through the *type of problem addressed* (industry or academia), *main domain of the problem* (simulation or artificial intelligence), *type of result* (approach, framework, research, methodology, development, tutorial or heuristic), *improvement addressed* (optimizing, estimating, etc.), *application field*, and *level of generality in the application field*. On the other hand, the *methods* category deals with the methods used to get results in the articles through the *level of generality in the simulation field* (simulation type, formalism), *machine learning technique used*, *machine learning software tool used*, and *simulation software tool used*. Relationships between categories and research questions to support data extraction are summarized here.

Descriptive statistics and analysis were performed after data extraction. In the following section, we present the results and conclusions obtained (i.e., step 8).

3 RESULTS AND DISCUSSION

To analyze the purposes of using ML in DES projects, we decided to study the relationship among the *type of problem addressed*, *the main domain of the problem*, and *improvement addressed* categories. Figure 2 summarizes the articles' classification due to the analysis performed based on these categories. As the figure shows, *optimization*, *decision-making*, and *scheduling* are the top three domains where improvements were addressed with both ML and simulation techniques.

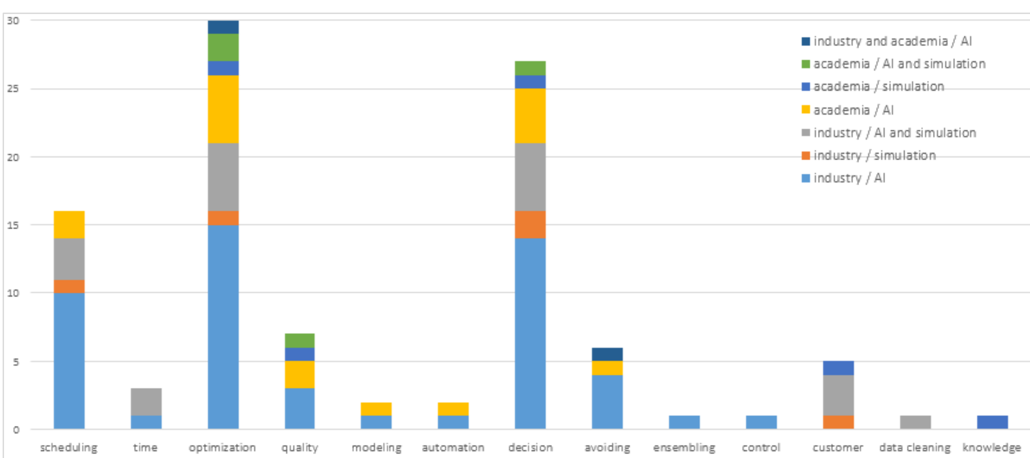

Figure 2: Articles by *type of problem addressed*, *main domain of the problem*, and *improvement addressed*.

For *optimization* subcategory (placed at #1), articles devoted to time (Feng et al. 2018, Biller et al. 2022), earth moving operations (Shitole et al. 2019), inventory (Afridi et al. 2020), emergency rooms (Rashwan et al. 2018, Prabhu et al. 2023), and lot dispatching (Stöckermann et al. 2023) optimization were found. Most of these articles are devoted to industrial applications, centered on ML techniques complemented with DES to support minor features of the model. This pattern is repeatedly exhibited in all *improvement* subcategories except for *data cleaning*, *customer service*, and *time estimation*, where both ML and simulation are employed on an equity basis. It is easy to see that neither academia/simulation nor industry/simulation are predominant fields using ML as complementary techniques for minor features.

A deeper analysis regarding the *type of problem addressed* and the *improvement addressed* following the *machine learning technique used* and the *level of generality in the simulation field* is presented in Figure 3a. As the reader can see, most articles using DES and ML are applied to industry, while just a few articles deal with academic research problems. However, when analyzing the same features for the DEVS portion, the proportion regarding academia/industry is reversed (i.e., most DEVS and ML learning papers are based on academic applications instead of addressing real-life industry issues).

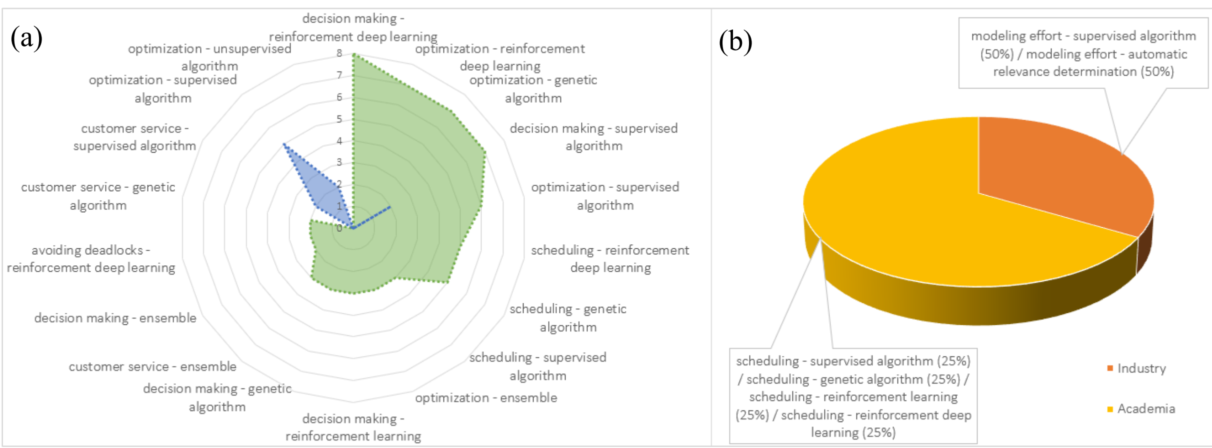

Figure 3: Distribution of DES articles by *type of problem addressed*, *improvement addressed*, *machine learning technique used*, and *level of generality in the simulation field*. (a) On the left, the green shape illustrates the number of articles dealing with a specific *type of problem* using a predefined *ML technique* in an industrial environment. The blue shape illustrates the same number but for academic applications. (b) On the right, the article distribution only considers DEVS papers (i.e., a subset of the DES articles considered in Figure 3a).

Figure 3a details the ML techniques employed to support the domains described in Figure 2. Following domain leadership in industry, *optimization* and *decision-making* problems are outstanding other domains employing *deep reinforcement learning* and *supervised learning* algorithms on top of other ML techniques. Specifically, *deep reinforcement learning* algorithms were used combined with DES, for example, to support real-time scheduling of flexible job shop production (Lang et al. 2020), to develop adaptive scheduling algorithms for production lines (Woo et al. 2021), to provide real-time decision making in manufacturing processes (Gros et al. 2020, Cheng et al. 2023), and to develop task selection in warehouses (Li et al. 2019). Regarding these combinations of *industry*, *deep reinforcement learning*, and *DES*, the three articles stand out from the others:

- A combination of both techniques for controlling a flexible flow shop using a gantry robot system as the transportation unit (Zisgen et al. 2023). Here, the agent learns autonomously the control policy to move the carriages. Such an agent is trained using iteratively a DES model of the manufacturing system and the Deep-Q-Network.

- A conceptual approach to handle logistic deadlocks with artificial neural networks implemented with an agent built with reinforcement learning techniques based on deep Q-networks (Müller et al. 2022). In this article, a DES of an automated guided vehicle system is used as the learning environment. As a result, the authors conclude that artificial neural networks can learn to handle deadlock in logistic systems with low complexity.
- The control of automated guided vehicles (AGVs) in modular production systems through reinforcement learning algorithms combined with simulation (Feldkamp et al. 2020). As in the previous case, the reinforcement learning algorithm is trained using a simulation model of a production system.

On the other hand, *supervised algorithms* were used to support manufacturing lead time predictions (Smith and Dickinson 2022), reducing response times (Pappert and Rose 2022), and building an infrastructure for a virtual factory (Jain et al. 2019), among other things (Rashwan et al. 2016, Singh et al. 2016, Rashwan et al. 2018, Shitole et al. 2019, Pappert and Rose 2022). An interesting approach is presented in (Jackson and Velazquez-Martinez 2021), where authors introduce a classification approach for getting candidate solutions in simulation models of logistic systems.

From Figure 3a, it is also worth mentioning the combination of the *optimization* domain with DES and *genetic algorithms* in *industry*. Using genetic algorithms as the optimization algorithm, (Ma et al. 2014) proposed parallel simulation-based optimization for scheduling a semiconductor manufacturing system. Can and Heavy (2016) use genetic programming and effective process time to predict cycle time using a DES model of a production line. Likely, Adan et al. (2022) and Ghasemi et al. (2022) promote the use of genetic algorithms for scheduling in manufacturing. More recently, Montevechi et al. (2023) propose a method based on several techniques (i.e., Latin hypercube design, hyper-parameters optimization, bagging, and genetic algorithm) for optimization of an acquisition function. In this field of research, a different approach is presented by Shrestha and Behzadan (2017). Such an article proposes a scientific methodology for generating more stable simulation models using an evolutionary algorithm that produces clean datasets by processing and significantly reducing noise in imperfect data (in this case, obtained from sensors).

Just a few articles are identified in Figure 3a for academic applications. It is easy to see that these papers work with *supervised* and *unsupervised algorithms* for *decision-making* and *optimization* domains (Bergmann et al 2015, Mayer et al. 2018, Cao et al. 2021, Feldkamp et al. 2022, Biller et al. 2022). These papers deal with more generic applications than industrial applications. We use the term "generic applications" to refer to solutions that can be translated to other domains besides the one they emerge from. For example, (Cao et al. 2021) and (Biller et al. 2022) propose a solution applicable to digital twins in general (not in particular). In (Bergmann et al. 2015), the authors study the suitability of several data mining and supervised machine learning methods for emulating job scheduling decisions by introducing binary decisions. The article presents a new step to effectively use data mining methods in the context of automatic simulation model generation. As evident, this is a common problem with generic domain applications. Additionally, Feldkamp et al. (2022) investigate the suitability of explainable artificial intelligence methods in real-world applications by using a DES model of a production line as a case study. However, the results are generic and can be used to support any DES model. Finally, Mayer et al. (2018) investigate a simulation-based supervised learning approach to determine the suitability of a particular algorithm from a set of algorithms for a given problem based on a set of characteristics. As described, the contribution of this paper is universal.

Figure 3b shows the subset of articles devoted to DEVS specifically. As the figure shows, most papers are devoted to academic applications centered on *supervised learning*, *reinforcement learning*, *deep reinforcement learning*, and *genetic algorithms*. In (Kessler et al. 2017), the authors propose a DEVS-based approach allowing the use of hierarchical Markov Decision Processes and reinforcement learning to solve planning or decision problems. Another example is (Sarjoughian et al. 2023). In such an article, the authors aim to use ML to study and predict the dynamics of discrete-event systems through the development of Parallel DEVS models.

Following our research analysis, Figure 4 uses the classification built over the *machine learning technique used*, the *level of generality in the simulation field*, and the *machine learning software tool used* to depict how ML techniques and tools are used for DES projects. By analyzing Figure 4a, it is easy to see that most articles do not have any data regarding the software tool used to support the ML technique. Then, three "tools" are used more frequently than others: *Python* (leading the ranking), *Java*, and *Matlab*.

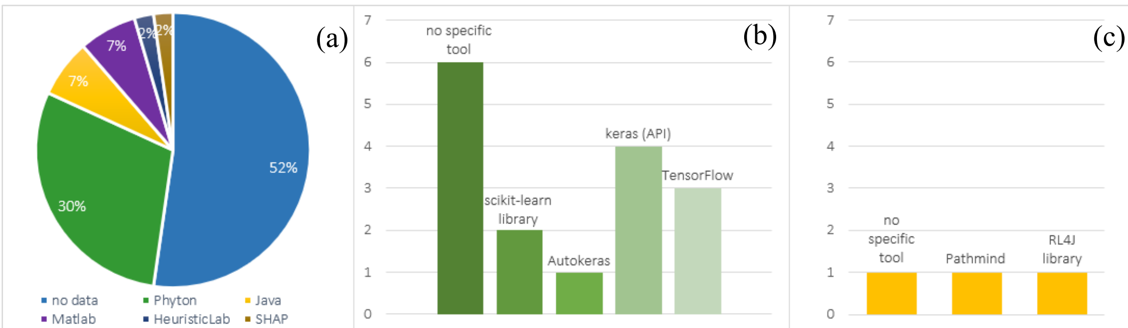

Figure 4: Distribution of DES articles by *machine learning technique used*, *level of generality in the simulation field*, and *machine learning software tool used*. (a) On the left: The distribution of machine learning tools used in the 44 articles analyzed. (b) In the middle: A detailed distribution centered on the use of Phyton (i.e., the software tool most used in the papers) from Figure 4a. (c) On the right: A detailed distribution centered on the use of Java (i.e., the second most used tool in the papers) from Figure 4a.

Python itself is just a programming language. When applied to ML, *Python*-based software tools (IDE, libraries, or frameworks) are used (Figure 4b):

- *Scikit-learn* is a popular open-source *Python* library used for ML and data mining. It allows the programmer to build supervised and unsupervised learning models. Articles using this tool are (Smith and Dickinson 2022, Sarjoughian et al. 2023).
- *TensorFlow* is an open-source ML framework based on *Python* developed by Google. It is designed to make it easier to build, train, and deploy ML models (particularly, neural networks). Articles using *TensorFlow* are (Feldkamp et al 2020, Gros et al 2020)
- *Keras* is a high-level API built on top of *TensorFlow*. It provides full control of the models using clean and simple *Python* code. For example, *Keras* is used to develop supervised and unsupervised learning in the article (Cao et al. 2021). In other cases, *Keras* is used as a complement to *TensorFlow* (Feldkamp et al. 2020, Smith and Dickinson 2022).
- *Autokeras* is an open-source Automated ML library built on top of *TensorFlow* and *Keras*. It helps automate the process of designing and training deep learning models. Hence, high-performing models can be built with minimal coding and ML expertise. In the paper (Pappert and Rose 2022), for example, the authors use *Autokeras* to implement supervised learning and evolutive algorithms.

Like *Python*, *Java* is a popular programming language over which ML models can be built by using specific libraries, such as:

- *Pathmind*: Focused on reinforcement learning for real-world applications, Pathmind offers tools and platforms that help developers resolve complex optimization problems using machine learning techniques. E.g., it is used in (Farhan et al. 2020).
- RL4J is a reinforcement learning library for Java. It implements several well-known reinforcement learning algorithms, such as DQN (Deep Q-network), A3C (Asynchronous Advantage Actor-Critic), DDPG (Deep Deterministic Policy Gradient), and PPO (Proximal Policy Optimization). E.g.: (Afridi et al. 2020).

Finally, *Matlab* is a high-level programming language and environment primarily used for numerical computing, data analysis, algorithm development, and visualization. It allows for building ML algorithms and models as in, for example, the paper (Feng et al. 2018).

Moving forward, Figure 5 shows the analysis regarding the *type of results* produced in the articles. In general, *approaches* are the most proposed (as a guide for researchers to solve a problem), followed by *methodologies* (a set of methods, principles, and rules to guide how specific research should be conducted).

For each *type of result*, Figure 5 details the *ML techniques* employed in the proposal and the *simulation software tool* combined with the *ML technique* under analysis. As the figure shows, several combinations were found: *i)* the same *ML tool* used for distinct *types of results* with different *simulation tools*, *ii)* the same *simulation tool* for the same *type of result* combined with different *ML techniques*, and *iii)* the same *simulation tool* with same *ML technique* for *DES* providing different *types of result*. An example of *ii)* is (Dodge et al. 2023), where the authors use DES with SimPy combined with genetic and evolutive algorithms. Similar cases are (Rabe and Dross 2015, Rashwan et al. 2016, Shitole et al. 2019, Gros et al. 2020, Jackson and Velazquez-Martinez 2021, Cao et al. 2021, Pappert and Rose 2022). On the other hand, in (Rashwan et al. 2018), DES is worked with AnyLogic by combining a supervised learning algorithm to produce both an approach and a framework also (i.e., this is an example of *iii)*).

Figure 5: Distribution of articles by *type of result*, *machine learning technique used*, *level of generality in the simulation field*, and *simulation software tool used*.

At this point, it is important to note that, in some cases, the same programming language is used to support the ML technique and the DES. For example, Jackson and Velazquez-Martinez (2021) use Python for both the creation and execution of the required simulation model and to develop the genetic algorithm and supervised learning model needed to support their study.

Regarding DEVS, only approaches and frameworks are reported. The tools used to support the development of DEVS studies are DEVSSimPy (https://github.com/capocchi/DEVSimPy), MS4Me (https://rtsync.com/ms4me), and DEVS Suite (https://acims.asu.edu/devs-suite/). These are some of the most popular software tools for DEVS. Specifically, DEVSSimPy is Python-based, while MS4Me and DEVS Suite are Java-based.

Finally, Figure 6 shows the analysis regarding the *main domain of the problem*, *application field*, and *level of generality in the application field*. As the figure illustrates, several application fields are attached to the articles studied, such as logistics, inventory, warehouses, etc. This list is quite like well-known domains where DES has been employed over time. A deeper analysis of DEVS articles reveals that the correlation between DEVS and the application field is preserved as in DES highlighting the AI contribution as the primary technique used.

Figure 6a highlights the importance of AI through ML in all domains. Factory systems solutions (Pappert and Rose 2022, Zhang et al. 2023) are higher than other domains since this type of dynamic system is well-represented by DES approaches. By using ML over such approaches, the studies produce more flexible, scalable, and efficient models than before.

On the other hand, Figure 6b uses the *level of generality of the proposed solution in the application field* to analyze if the proposal can be reused or translated to other fields. As the figure shows, in all domains, most solutions are domain-specific. However, by combining the subcategories *reusable in similar problems* and *domain-general*, there is a higher number of articles proposing solutions that can be translated to other situations. For example, articles (Biller et al. 2017, Jain et al. 2018, Biller et al. 2019, Devanga et al. 2022) indicate that the solution is developed at a general level and, therefore, can be translated to other domains. On the other hand, other articles indicate that the proposed solution can be employed to solve similar problems (e.g., (Leon et al. 2022)).

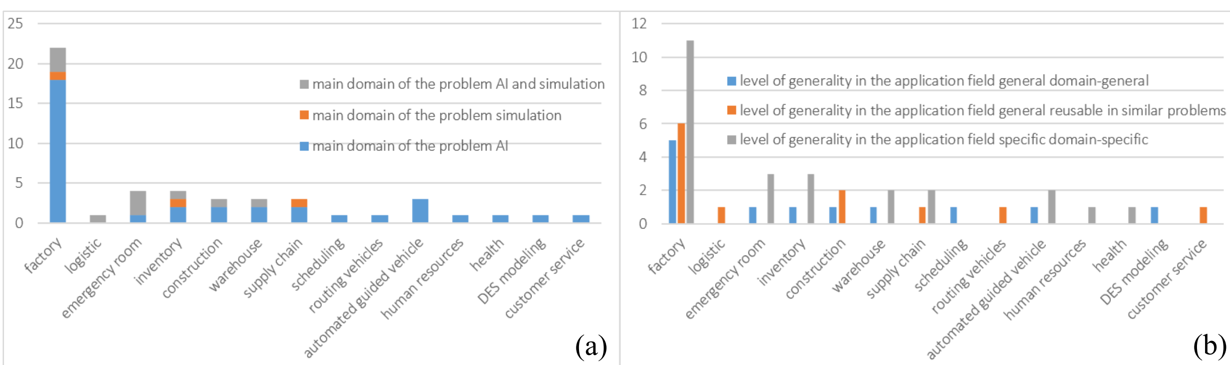

Figure 6: Distribution of articles by *main domain of the problem*, *application field*, and *level of generality in the application field*. (a) On the left, the chart illustrates the *problem domain* vs. the *application field*. (b) On the right, the chart illustrates the *problem domain* vs. the *level of generality in the field*.

4 CONCLUSIONS AND FUTURE DIRECTIONS

Simulation provides an understanding of the design, planning, and operation of complex systems in commerce, industry, and society. The SLR proved to be a satisfactory technique for exploring existing literature on DES/DEVS and ML. The purpose of this paper was not to conduct exhaustive research of all articles on the topic, but rather to systematically analyze WSC contributions from 2013 to 2023 to understand the information available and identify how the combination of both fields can grow in the future.

The following conclusions have been reached concerning our Research Questions (RQ). Using ML techniques as part of the optimization solutions is the most popular use (RQ1). Most research is supported by AI and applied to different industrial domains (RQ1/RQ6). This is also evidenced by the available software tools in the ML field that allow building ML models, mainly deep reinforcement learning models, without needing any background on ML (RQ3).

On the other hand, several types of results in several domains are achieved by combining DES and ML (RQ4/RQ5). Most of them use supervised, unsupervised, and deep-reinforcement learning combined with DES solutions. Evidence shows that ML techniques have been used as complementary approaches for solving traditional DES problems (RQ5). Here, the most common software tools used for simulation purposes are SIMIO (https://www.simio.com/es/), AnyLogic (https://www.anylogic.com/), and FlexSim (https://www.flexsim.com/) (RQ3/RQ4).

For academic applications, DEVS formalism is still highly used (RQ2). We believe that DEVS is not used in the industry field due to the level of knowledge required to use DEVS formalism for building models. Also, this is probably related to the software tools available to build DEVS models in a way that can be combined with ML models. Since DEVS models are formally defined but developed in general-purpose programming languages (or libraries based on these), the combination of DEVS models and simulators with ML models is not straightforward. This is easier if the modeler uses the same programming language to support both simulators and ML models, as in the cases of Python and Java (RQ3/RQ4). Moreover, the results of DEVS proposals are mainly domain-general, meaning that can be applied to other domains (RQ6). More generically, the academic proposals are all domain-general, while industrial applications are mainly domain-specific (or applicable to similar contexts).

The conclusions highlighted above were derived from the insights observed by the authors during this research. Of course, the reader can use our findings to obtain a more detailed explanation of these. For space reasons, we cannot provide a detailed answer to all research questions. However, the main features related to such questions have been addressed in Section 3.

A SYSTEMATIC LITERATURE REVIEW DATA

The data supporting this paper is available here.

REFERENCES

Adan, J., A. Akcay, J. Fowler, M. Albers, and M. Geurtsen. 2022. "Autonomous Scheduling in Semiconductor Back-End Manufacturing". In *2022 Winter Simulation Conference (WSC)*, 3489–3500 https://doi.org/10.1109/WSC57314.2022.10015252.

Afridi, M., S. Nieto-Isaza, H. Ehm, T. Ponsignon, and A. Hamed. 2020. "A Deep Reinforcement Learning Approach for Optimal Replenishment Policy in A Vendor Managed Inventory Setting For Semiconductors". In *2020 Winter Simulation Conference (WSC)*, 1753–1764 https://doi.org/10.1109/WSC48552.2020.9384048.

Bergmann, S., N. Feldkamp, and S. Strassburger. 2015. "Approximation of Dispatching Rules for Manufacturing Simulation using data mining Methods" In *2015 Winter Simulation Conference (WSC)*, 2329–2340 https://doi.org/10.1109/WSC.2015.7408344.

Biller, B., E. Wolf, and E. Yücesan. 2019. "Inventory Management Under Disruption Risk". In *2019 Winter Simulation Conference (WSC)*, 1918–1929 https://doi.org/10.1109/WSC40007.2019.9004888.

Biller, B., S. R. Biller, O. Dulgeroglu, and C. G. Corlu. 2017. "The Role of Learning on Industrial Simulation Design and Analysis". In *2017 Winter Simulation Conference (WSC)*, 3287–3298 https://doi.org/10.1109/WSC.2017.8248046.

Biller, B., X. Jiang, J. Yi, P. Venditti, and S. Biller. 2022. "Simulation: the Critical Technology in Digital Twin Development". In *2022 Winter Simulation Conference (WSC)*, 1340–1355 https://doi.org/10.1109/WSC57314.2022.10015246.

Breretona, P., B. Kitchenham, D. Budgenb, M. Turnera, and M. Khalilc. 2007. "Lessons from Applying the Systematic Literature Review Process within the Software Engineering Domain". *Journal of Systems and Software* 80(4):571–83.

Can B. and C. Heavy. 2016. "A Demonstration of Machine Learning for Explicit Functions for Cycle Time Prediction Using MES Data". In *2016 Winter Simulation Conference (WSC)*, 2500–2511 https://doi.org/10.1109/WSC.2016.7822289.

Cao, Y., C. Currie, B. Onggo, and M. Higgins. 2021. "Simulation Optimization for a Digital Twin Using a Multi-Fidelity Framework". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715498.

Cheng, L., J. Luo, W. Fan, Y. Zhang, and Y. Li. 2023. "A Deep Q-Network Based on Radial Basis Functions for Multi-Echelon Inventory Management". In *2023 Winter Simulation Conference (WSC)*, 1581–1592 https://doi.org/10.1109/WSC60868.2023.10407308.

Cronin, P., F. Ryan, and M. Coughlan. 2008. "Undertaking a Literature Review: A Step-by-Step Approach." *British Journal of Nursing* 17(1):38–43.

Denyer D. and D. Tranfield. 2009. "Producing a Systematic Review". In *The SAGE Handbook of Organizational Research Methods*, edited by D. Buchanan and A. Bryman, 671–689. New York: SAGE Publications.

Denyer, D., D. Tranfield, and J. Ernst Van Aken. 2008. "Developing Design Propositions Through Research Synthesis". *Organization studies* 29(3):393–413.

Devanga, A., E. Badilla, and M. Dehghanimohammadabadi. 2022. "Applied Reinforcement Learning for Decision Making in Industrial Simulation Environments". In *2022 Winter Simulation Conference (WSC)*, 2819–2829 https://doi.org/10.1109/WSC57314.2022.10015282.

Dodge, R., G. Pedrielli, and P. Jevtić. 2023. "A Testing Based Approach for Security Analysis of Smart Semiconductor Systems". In *2023 Winter Simulation Conference (WSC)*, 2286–2297 https://doi.org/10.1109/WSC60868.2023.10408730.

Farhan, M., B. Göhre, and E. Junprung. 2020. "Reinforcement Learning in Anylogic Simulation Models: A Guiding Example using Pathmind". In *2020 Winter Simulation Conference (WSC)*, 3212–3223 https://doi.org/10.1109/WSC48552.2020.9383916.

Feldkamp, N, S. Bergmann, and S. Strassburger. 2020. "Simulation-Based Deep Reinforcement Learning for Modular Production Systems". In *2020 Winter Simulation Conference (WSC)*, 1596–1607 https://doi.org/10.1109/WSC48552.2020.9384089.

Feldkamp, N., J. Genath, and S. Strassburger. 2022. "Explainable AI For Data Farming Output Analysis: A Use Case for Knowledge Generation Through Black-Box Classifiers". In *2022 Winter Simulation Conference (WSC)*, 1152–1163 https://doi.org/10.1109/WSC57314.2022.10015304.

Feng, K., S. Chen, and W. Lu. 2018. "Machine Learning Based Construction Simulation and Optimization". In *2018 Winter Simulation Conference (WSC)*, 2025–2036 https://doi.org/10.1109/WSC.2018.8632290.

Ghasemi, A., K. Kabak, and C. Heavy. 2022. "Demonstration of the Feasibility of Real Time Application of Machine Learning to Production Scheduling". In *2022 Winter Simulation Conference (WSC)*, 3406–3417 https://doi.org/10.1109/WSC57314.2022.10015436.

Gros, T. P., J. Groß, and V. Wolf. 2020. "Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning". In *2020 Winter Simulation Conference (WSC)*, 3032–3044 https://doi.org/10.1109/WSC48552.2020.9383884.

Ishlam Nazrul, M. N., C. Tropper, R. A. McDougal, and W. Lytton. 2017. "Optimizations for Neuron Time Warp (NTW) for Stochastic Reaction-Diffusion Models of Neurons". In *2017 Winter Simulation Conference (WSC)*, 1252–1263 https://doi.org/10.1109/WSC.2017.8247871.

Jackson I. and J. C. Velazquez-Martinez. 2021. "Automl Approach to Classification of Candidate Solutions for Simulation Models of Logistic Systems". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715416.

Jain, S., A. Narayanan, and Y. Lee. 2018. "Comparison of Data Analytics Approaches Using Simulation". In *2018 Winter Simulation Conference (WSC)*, 1084–1095 https://doi.org/10.1109/WSC.2018.8632330.

Jain, S., A. Narayanan, and Y. Lee. 2019. "Infrastructure for Model Based Analytics for Manufacturing". In *2019 Winter Simulation Conference (WSC)*, 2037–2048 https://doi.org/10.1109/WSC40007.2019.9004893.

Kessler, C., L. Capocchi, J. Santucci, and B. Zeigler. 2017. "Hierarchical Markov Decision Process based on DEVS Formalism". In *2017 Winter Simulation Conference (WSC)*, 1001–1012 https://doi.org/10.1109/WSC.2017.8247850.

Kitchenham, B. and Stuart Charters. 2007. "Guidelines for Performing Systematic Literature Reviews in Software Engineering". EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University, Keele, England.

Lang, S., F. Behrendt, N. Lanzerath, T. Reggelin, and M. Müller. 2020. "Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production". In *2020 Winter Simulation Conference (WSC)*, 3057–3068 https://doi.org/10.1109/WSC48552.2020.9383997.

Leon, J., P. Marone, M. Peyman, Y. Li, L. Calvet, and M. Dehghanimohammadabadi. 2022. "A Tutorial on Combining Flexsim with Python for Developing Discrete-Event Simheuristics". In *2022 Winter Simulation Conference (WSC)*, 1386–1400 https://doi.org/10.1109/WSC57314.2022.10015309.

Levy, Y. and T. Ellis. 2006. "A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research". *Informing Science* 9(1):181–211.

Li, M. P., P. Sankaran, M. E. Kuhl, R. Ptucha, A. Ganguly, and A. Kwasinski. 2019. "Task Selection by Autonomous Mobile Robots in A Warehouse Using Deep Reinforcement Learning". In *2019 Winter Simulation Conference (WSC)*, 680–689 https://doi.org/10.1109/WSC40007.2019.9004792.

Ma, Y., F. Qiao, W. Yu, and J. Lu. 2014. "Parallel Simulation-based Optimization on scheduling of a Semiconductor Manufacturing System". In *2014 Winter Simulation Conference (WSC)*, 2571–2579 https://doi.org/10.1109/WSC.2014.7020101.

Mayer, T., T. Uhlig, and O. Rose. 2018. "Simulation-Based Autonomous Algorithm Selection for Dynamic Vehicle Routing Problems with the Help of Supervised Learning Methods". In *2018 Winter Simulation Conference (WSC)*, 3001–3012 https://doi.org/10.1109/WSC.2018.8632452.

Montevechi, J., J. Amaral, R. Miranda, C. Dos Santos, F. Brito, and M. Machado. 2023. "Ensemble-Based Infill Search Simulation Optimization Framework". In *2023 Winter Simulation Conference (WSC)*, 3059–3070 https://doi.org/10.1109/WSC60868.2023.10408380.

Müller, M., T. Reggelin, I. Kutsenko, H. Zadek, and L. Reyes-Rubiano. 2022. "Towards Deadlock Handling with Machine Learning in a Simulation-Based Learning Environment". In *2022 Winter Simulation Conference (WSC)*, 1485–1496 https://doi.org/10.1109/WSC57314.2022.10015270.

Pappert F. S. and O. Rose. 2022. "Using Data Farming and Machine Learning to Reduce Response Time for the User". In *2022 Winter Simulation Conference (WSC)*, 1707–1718 https://doi.org/10.1109/WSC57314.2022.10015466.

Pfeiffer, A., D. Gyulai, Á. Szaller, and L. Monostori. 2018. "Production Log Data Analysis for Reject Rate Prediction and Workload Estimation". In *2018 Winter Simulation Conference (WSC)*, 3364–3374 https://doi.org/10.1109/WSC.2018.8632482.

Prabhu, V., K. Taaffe, R. Pirrallo, W. Jackson, M. Ramsay, and J. Hobbs. 2023. "Forecasting Patient Arrivals and Optimizing Physician Shift Scheduling in Emergency Departments". In *2023 Winter Simulation Conference (WSC)*, 1136–1147 https://doi.org/10.1109/WSC60868.2023.10407202.

Rabe, M. and F. Dross. 2015. "A Reinforcement Learning Approach for a Decision Support System for Logistics Networks". In *2015 Winter Simulation Conference (WSC)*, 2020–2032 https://doi.org/10.1109/WSC.2015.7408317.

Rashwan, W., H. Habib, G. Courtney, A. Arisha, and S. Kennelly. 2016. "An Integrated Approach of Multi-Objective Optimization Model for Evaluating New Supporting Program in Irish Hospitals". In *2016 Winter Simulation Conference (WSC)*, 1904–1915 https://doi.org/10.1109/WSC.2016.7822236.

Rashwan, W., J. Fowler, and A. Arisha. 2018. "A Multi-Method Scheduling Framework for Medical Staff". In *2018 Winter Simulation Conference (WSC)*, 1464–1475 https://doi.org/10.1109/WSC.2018.8632247.

S. Singh, S., R. Pimplikar, R. Chaudhuri, and G. Parija. 2016. "Outplacement Time and Probability Estimation using Discrete Event Simulation". In *2016 Winter Simulation Conference (WSC)*, pp. 989–1000 https://doi.org/10.1109/WSC.2016.7822159.

Sarjoughian, H. S., F. Fallah, S. Saeidi, and E. J. Yellig. 2023. "Transforming Discrete Event Models To Machine Learning Models". In *2023 Winter Simulation Conference (WSC)*, 2662–2673 https://doi.org/10.1109/WSC60868.2023.10407348.

Shitole, V, J. Louis, and P. Tadepalli. 2019. "Optimizing Earth Moving Operations Via Reinforcement Learning". In *2019 Winter Simulation Conference (WSC)*, 2954–2965 https://doi.org/10.1109/WSC40007.2019.9004935.

Shrestha P. and A. H. Behzadan. 2017. "An Evolutionary Method to Refine Imperfect Sensor Data for Construction Simulation". In *2017 Winter Simulation Conference (WSC)*, 2460–2471 https://doi.org/10.1109/WSC.2017.8247975.

Smith J. K. and C. Dickinson. 2022. “"Discrete-Event Simulation and Machine Learning for Prototype Composites Manufacture Lead Time Predictions". In *2022 Winter Simulation Conference (WSC)*, 1695–1706 https://doi.org/10.1109/WSC57314.2022.10015322.

Stöckermann, P. A. Immordino, T. Altenmüller, G. Seidel, M. Gebser, and P. Tassel. 2023. "Dispatching in Real Frontend Fabs with Industrial Grade Discrete-Event Simulations by Deep Reinforcement Learning with Evolution Strategies". In *2023 Winter Simulation Conference (WSC)*, 3047–3058 https://doi.org/10.1109/WSC60868.2023.10408625.

Woo, J., Y. Cho, S. Nam, and J. Nam. 2021. "Development of a Reinforcement Learning-Based Adaptive Scheduling Algorithm for Block Assembly Production Line". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715509.

Xiao, Y. and M. Watson. 2017. "Guidance on Conducting a Systematic Literature Review". *Journal of Planning Education and Research* 39(1):93–112.

Zhang, T., K. E. Kabak, C. Heavy, and O. Rose. 2023. "A Reinforcement Learning Approach for Improved Photolithography Schedules". In *2023 Winter Simulation Conference (WSC)*, 2136–2147 https://doi.org/10.1109/WSC60868.2023.10408616.

Zisgen, H., R. Miltenberger, M. Hochhaus, and N. Stöhr. 2023. "Dynamic Scheduling of Gantry Robots using Simulation and Reinforcement Learning". In *2023 Winter Simulation Conference (WSC)*, 3026–3034 https://doi.org/10.1109/WSC60868.2023.10407159.

AUTHOR BIOGRAPHIES

YANINA DÖNING is a master's student in Information Systems Engineering at the Facultad Regional Santa Fe – Universidad Tecnológica Nacional. She received her degree in Engineering from Universidad Nacional del Litoral in 2019. Her email address is ydoning@frsf.utn.edu.ar.

MARIA JULIA BLAS is an Assistant Researcher at INGAR and an Assistant Professor at UTN. She received her Ph.D. degree in Engineering from UTN in 2019. Her research interests include UML modeling and discrete-event M&S. Her email address is mariajuliablas@santafe-conicet.gov.ar.