Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

METAMODEL-BASED ORDER PICKING FOR AUTOMATED STORAGE AND RETRIEVAL SYSTEMS

Andrea Ferrari[1], and Canan Gunes Corlu[2]

[1]Department of Management and Production Engineering, Politecnico di Torino, Torino, ITALY
[2]Metropolitan College, Boston University, Boston, USA

ABSTRACT

Automated warehouses, a key component of modern supply chain processes, have been widely introduced in various industries. Their automation, including sophisticated systems such as automated storage and retrieval systems, plays a crucial role in improving efficiency and reducing operational costs. This paper focuses on the order picking problem in a multi-level shuttle system, which aims at predicting the time required to fulfill a set of picking orders, i.e., the makespan. A metamodel based on a neural network architecture that exploits long short-term memory and linear layers is proposed. The metamodel was trained and tested on synthetic data from a stochastic discrete event simulation model. Extensive experiments illustrate the validity of the metamodel in accurately predicting the makespan. The study not only advances theoretical modeling in the context of automated warehousing, but also outlines future research directions for improved metamodel performance and broader applications, such as stochastic optimization and deadlock prediction.

1 INTRODUCTION

Automated warehouses play a pivotal role in shaping and facilitating modern supply chain processes (Marolt et al. 2022). Nowadays, their widespread adoption spans various industry sectors, including manufacturing, large-scale distribution, e-commerce, and healthcare. These facilities can be broadly characterized as innovative warehouses that incorporate varying levels of automation (Mahdavisharif et al. 2022). This automation involves activities related to the storage and retrieval of goods, as well as internal material handling within the facility. An illustrative implementation of such automation can be found in automated storage and retrieval systems (AS/RSs), which are characterized by the storage and retrieval of unit loads (ULs) without the need for human intervention (Azadeh et al. 2019). Compared to traditional warehousing systems, AS/RSs offer significant benefits including reduced space utilization, lower labor costs, faster retrieval, and improved inventory control (Boysen and Stephan 2016).

Many approaches have been used to investigate and optimize AS/RSs. Various studies in the AS/RSs literature have focused on the use of analytical models to optimize storage and retrieval operations (Lehmann et al. 2021). When the analytical evaluation is complex, simulation approaches can be used to generate accurate performance measures for a specific system configuration (Epp et al. 2017). Other studies have addressed the optimization of a specific process, such as the task scheduling problem, through the application of heuristics (Li et al. 2020) or metaheuristic (Fandi et al. 2022) techniques. However, each of these approaches has its weaknesses. The analytical models tend to become difficult to formulate and computationally intractable when applied to complex systems and large problem instances (Suemitsu et al. 2022). In traditional simulation, the domain of input parameters may be too large to allow a comprehensive evaluation of all combinations of values to determine the optimal response (Nag and Pal 2022). Although heuristics and metaheuristics are widely used, they have their shortcomings. For example, heuristics can only find a general solution, while metaheuristics are limited by their capabilities when used alone (Hsu and Wang 2023).

An alternative modeling technique, also employed in this paper, is the use of metamodels or surrogate models. These are often constructed on the basis of simulation outputs with the objective of facilitating the analysis. The main advantage of using metamodels over direct simulation lies in the fact that the surrogate model is able to provide quick predictions without the need to run time-consuming simulation models (Xu et al. 2015). Types of metamodels can be Gaussian process models, response surface methodology, and artificial neural network (NN).

While metamodel-based approaches have been effectively used in various warehouse contexts, there is a limited number of studies specifically addressing AS/RS challenges using these techniques. This paper presents a comprehensive approach to tackle the order-picking problem (OPP) in AS/RSs by leveraging metamodeling techniques. The OPP poses a significant challenge in the domain of automated warehousing, since it has been demonstrated that between 50 % and 65 % of the overall operating costs in warehousing can be attributed to the pivotal process of order picking (Chirici and Wang 2014). Recognizing the significance of mitigating labor-intensive and costly order picking, there is a rapid evolution in warehouse systems and processes, with a notable shift toward automation (Li et al. 2022). Therefore, the objectives of this paper are:

1. to define the structure of an NN-based metamodel capable of efficiently replacing a discrete event simulation (DES) model in predicting the makespan required to fulfill a set of picking orders in a multi-level shuttle (MLS) system,
2. to test the accuracy of the metamodel predictions, and
3. to evaluate the time difference between the metamodel and the DES model in predicting the makespan.

The remainder of this paper is structured as follows. Section 2 discusses the related work. Section 3 presents the MLS system and the characteristics of the order-picking system under study. Section 4 describes the stochastic DES model, the metamodel based on a deep NN, and the procedure used to generate the input data, while Section 5 presents the training, validation, and testing procedure for the metamodel, as well as a performance comparison with the DES model. Finally, Section 6 concludes with potential future research opportunities.

Table 1 provides a list of acronyms used throughout this study and their corresponding definitions for clarity and reference.

Table 1: List of acronyms.

Acronym	Definition	Acronym	Definition
AS/RS	Automated storage and retrieval system	MTTR	Mean time to repair
DES	Discrete event simulation	NN	Neural network
HM	Handling machine	OPP	Order-picking problem
LSTM	Long-short term memory	PS	Picking station
MAE	Mean absolute error	RNN	Recurrent neural network
MAPE	Mean absolute percentage error	SKU	Stock keeping unit
MLS	Multi-level shuttle	UL	Unit load
MTBF	Mean time between failure		

2 RELATED WORK

This section provides an overview of key contributions aimed at optimizing operations in warehouse environments by synthesizing findings from the recent literature. Specifically, it explores the broad spectrum of research efforts in this field and reveals how researchers have used various techniques, such as metamodeling, simulation-based optimization, and neural networks, to address the complexities of warehouse logistics.

In Kirchhoff et al. (2023), a Gaussian process-based metamodel was used to address operational planning challenges in high-bay warehouses. The research focused on a shift scheduling problem to minimize labor and energy costs while reducing task delays. The optimization was subject to various parameters, including the number of employees per shift, shift start times, mid-shift duration, storage strategy, and transportation task assignment strategy. The authors highlighted the utility of Gaussian process models, emphasizing their ability to perform point estimation and uncertainty quantification. In a different approach, Suemitsu et al. (2022) presented a rapid simulation-based method for optimizing order sequences in a complex warehouse system consisting of automated guided vehicles, picking robot arms, and conveyor modules. The optimization goal was to minimize makespan while avoiding deadlocks. The methodology combined DES, Bayesian recurrent neural network, and simulated annealing. In particular, the approach offered accelerated computation and considered order sequence optimization. However, a potential drawback was the need to pre-train the NN. For more conventional warehousing systems, Zhao and Wang (2021) demonstrated the integration of metamodeling in the form of a combination of DES and response surface methodology. The goal was to minimize both makespan and deadlock occurrences within a distribution system. Response surface methodology was highlighted for its efficiency in reducing the number of tests and shortening the experimental cycle. This approach demonstrated the benefits of metamodeling in improving the optimization of traditional warehouse systems.

Automation solutions often employ parts-to-picker methods, wherein products autonomously move from storage locations to the order picker, streamlining the process (Vijayakumar and Sgarbossa 2021). While some research has investigated picking activities within AS/RS contexts (Ma et al. 2023; Liu et al. 2021; Yang et al. 2022), there is a noticeable dearth of studies proposing metamodel-based solutions for optimizing OPP in AS/RS setups. Furthermore, within the broader scope of metamodeling techniques applied to warehousing, a distinct gap exists in developing NN-based metamodels tailored to the unique characteristics of AS/RS operations. Filling this research need, which is the goal of this paper, could lead to innovative methods and solutions for improving the efficiency and performance of warehouse logistics systems.

3 DESCRIPTION OF THE PROBLEM

3.1 The Multi-Level Shuttle System

Previous studies have dealt with MLS systems (Ferrari et al. 2022; Ferrari et al. 2023) but generally focused on a single handling machine (HM) and a single aisle. This paper expands the perspective by integrating multiple aisles, multiple HMs, and multiple picking stations (PSs). The overall system configuration (Figure 1) was inspired by Wang et al. (2016), who studied a modified mini-load AS/RS designed to meet the needs of e-commerce in logistics.

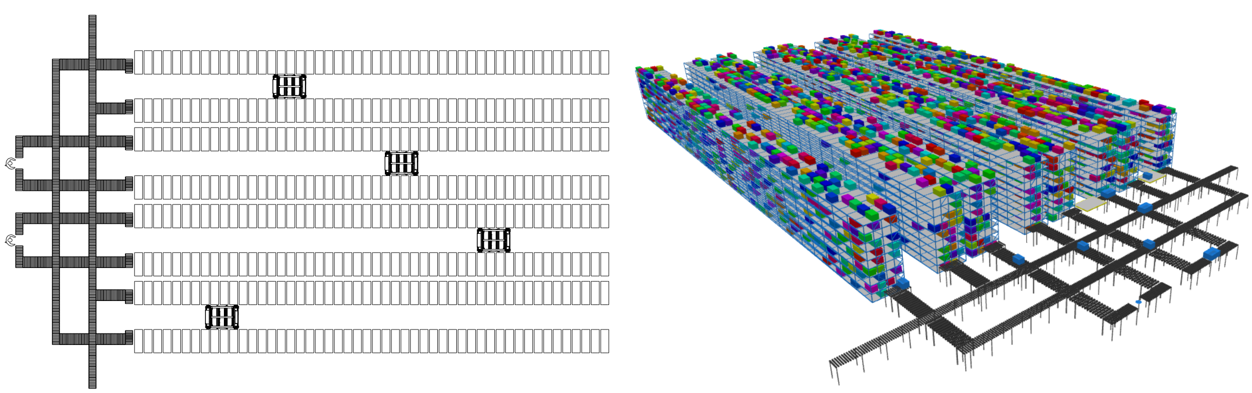
From a design perspective, the HM can move small ULs like plastic totes along three axes. The horizontal and vertical movements occur simultaneously. The system is also designed to handle two different types of ULs with different dimensions. The HM can move, store, and retrieve two ULs of the same type simultaneously. The MLS is an aisle-captive system, i.e., the HM cannot change the working aisle. Four aisles constitute the AS/RS, each with a double-front rack with multi-deep storage locations. Specifically, two large ULs or four small ULs can be stored in the same location at the same time. The MLS system also has an input/output roller conveyor system at one end of the aisle. In front of the storage area, a closed-loop conveyor system is used to supply a PS with the ULs needed to complete an order and to return ULs within the racks. If a UL is empty after picking, it is removed from the system; otherwise, it is returned to the automated warehouse for replenishment.

From an operational point of view, the HM operates either on single-command, dual-command, or multi-command type. In particular, a single-command cycle can be defined as the round trip from an input/output point to a storage location and back again. In a dual-command cycle, a storage operation of a UL is directly followed by a retrieval of another UL. Finally, in a multi-command cycle, the HM can perform storage and retrieval in complex sequences, with the ability to move more than one UL simultaneously. The dwelling policy is the point-of-service-completion, meaning that the HM remains at the position of the last operation when it completes the backlog of storage and retrieval missions to be performed. Finally, the MLS system works with class-based storage and nearest-neighbor relocation policies (Ferrari et al. 2023).

3.2 Problem Setting

Order processing in a multi-PS parts-to-picker system involves several critical decisions, including order sequencing, order allocation, UL selection, and UL sequencing (Wang et al. 2022). Each decision represents a sub-problem, many of which have been studied extensively in the literature. Notably, the majority of these sub-problems are recognized as NP-hard, requiring sophisticated methods for effective solutions and optimal results. Given the holistic approach adopted in this study, the primary focus is on the overall functionality of the system, rather than on the optimization of individual sub-problems that typically characterize AS/RSs. Consequently, each sub-problem is addressed using simple and efficient heuristics. This methodology ensures a reliable approximation of real system operations while maintaining the generalizability and reliability of the study.

Incoming orders are assigned to PSs based on the idea of evenly distributing the workloads among the available stations (Tadumadze et al. 2023). To do that, the next order is assigned to the least busy station with the least pre-assigned orders. Once an order is assigned to a PS, the ULs containing the Stock Keeping Units (SKUs) requested in the order lines must be selected. The same SKU could be contained in more than one UL. It can be reasonable to select ULs that can be obtained in the aisle in correspondence with the PS to reduce the UL travel distance as much as possible (Wang et al. 2016). Therefore, for each order line j, a list of potential ULs containing the SKU of that order line is selected. The ULs are first sorted based on the corresponding aisles, meaning that ULs stored in the aisles closest to the PS assigned to the order will be ranked first. Then, for each UL t, the maximal hit rate, i.e., the maximal number of items that can be picked for the order line from the UL, is calculated as $min\{o_j, a_t\}$, where o_j is the quantity required in the order line j and a_t in quantity stored in the UL t. Then, the ULs are sorted according to the descending hit rate value to select the UL that best matches the order line and minimizes the number of ULs moved (Tadumadze et al. 2023). Moreover, within each hit rate group, the ULs are sorted again based on descending values of the fill rate, calculated as o_j/a_t. This will prioritize ULs that contain a quantity of items closer to the one requested in the order and try to empty the UL as much as possible. If no UL contains all the items required in the order line, more than one UL is selected (respecting the hit rate sorting) until the number of items satisfies the order.



(a) (b)

Figure 1: The multi-level shuttle system a) Floor view b) 3D view.

Orders are triggered when 70 % of the previous picking order is completed. This prevents the PS and the operators from waiting long before the next UL associated with the incoming order arrives at the station. The ULs selected for an order need to be retrieved by the HM. The sequence of missions to retrieve them is generated based on a dynamic programming algorithm that aims to minimize the total completion time of a retrieval set. The shorter conveyor path to transport the ULs to the pre-assigned PS is then calculated. Consistent with the selection of the least busy PS, the definition of the shorter path also takes into account the number of ULs currently being conveyed and the number of ULs that have been preassigned to pass through the conveyor.

After completing the picking process, if the ULs assigned to the order contain some leftover items, they need to be restored in the AS/RS. To do that, the least busy aisle – the aisle with the least number of tasks to be completed in the queue, plus the incoming UL from previously completed picking orders, plus the storage capacity used – is selected.

The demand distribution of SKUs can be approximated with the typical ABC curve, where 20 % of SKUs account for 80 % of the total demand (Silva et al. 2022; Xu et al. 2019; Hausman et al. 1976). Therefore, the warehouse racks can be divided into three zones each (A, B, C), based on the SKU turnover rate. The number of storage locations assigned to each zone is proportioned based on the number of SKUs falling in the class assigned to the zone (A: 20 %, B: 30 %, C: 50 %) (Dijkstra and Roodbergen 2017). The assignment of a storage location to a particular class is defined by the distance, in terms of travel time, of the storage location from the input/output point (Zhou et al. 2022). Besides, the across-aisle storage strategy is used, which implies the presence of SKUs of all the classes in all the aisles (Gagliardi et al. 2015). Inside the zone, the SKUs are assigned randomly to storage locations (Yu et al. 2022).

HMs and conveyor stops are also considered. In this study, machine failures do not represent heavy and disruptive breakdowns, but easily manageable errors that can occur during the day-by-day warehouse operations. Examples of causes of system downtime could be electrical and sensor errors, mechanical errors, human errors, or UL defects (Giner, J., D. Katic, K. Kovacs, R. Glawar, and W. Sihn 2023), which could be solved quickly without blocking the system from functioning for a long period of time.

4 MODELING

4.1 The Discrete Event Simulation Model

To obtain the data to train the metamodel, a DES model of the MLS system was developed. A DES model allows for the examination of complex systems in detail and the development of virtual experiments to be performed. It is particularly useful when systems are too complex to be analyzed directly through observation or mathematical analysis (Law and Kelton 2007). The modeling software used is AnyLogic, which is designed to simulate diverse logistics and supply chain structures. This choice was supported by its ability to use pre-defined constructs while providing the flexibility to programmatically tailor the digital model to accurately represent the specific requirements of the simulated scenario. In developing the DES model, a set of assumptions was made:

- Each product UL can accommodate only one SKU at a time.
- The total number of SKUs available is 500.
- Inbound and outbound processes are not considered; hence, ULs with SKUs are pre-stored at the beginning of the simulation, and products picked are removed from the simulation.
- No replenishment is required, thus, the number of items in stock is sufficient to satisfy the order set, which is a group of picking orders.
- Both for HM and conveyors, the mean time between failures (MTBF) and the mean time to repair (MTTR) are assumed to follow a Weibull distribution.

Table 2 displays the main parameters of the DES model. In order to verify, calibrate, and validate the DES model, the outputs and performance of the MLS system modeled were compared to those of a similar system installed in a university laboratory. Details about the validation methodology, validation data, and specific model parameters can be found in Ferrari et al. (2022) and Ferrari et al. (2023).

Table 2: DES model parameters.

Parameter	Value	Parameter	Value
Columns	50	Speed conveyor	0.5 m/s
Tiers	10	Acceleration conveyor	0 m/s^2
Horizontal speed HM	4 m/s	Deceleration conveyor	0 m/s^2
Horizontal acceleration HM	1.5 m/s^2	Transfer time conveyors	$\sim$ Triangular(1.5, 2, 2.5) s
Horizontal deceleration HM	1.5 m/s^2	Picking time / item	$\sim$ Weibull(3.3,4) s
Vertical speed HM	0.8 m/s	MTBF HM	$\sim$ Weibull(120,1) min
Vertical acceleration HM	1.6 m/s^2	MTTR HM	$\sim$ Weibull(5.5,4) min
Vertical deceleration HM	1.6 m/s^2	MTBF conveyor	$\sim$ Weibull(180,1) min
Speed HM UL transfer	0.12 m/s	MTTR conveyor	$\sim$ Weibull(5.5,4) min

4.2 The Metamodel

The metamodel was based on an NN architecture designed for a regression task and implemented using the PyTorch framework. In the context of machine learning, a regression task involves predicting a continuous numeric value, e.g., the makespan, based on input data, e.g., the picking orders set. The choice of an NN as a metamodel was motivated by the fact that certain architectures, for example, recurrent neural networks (RNNs), excel in handling sequential data, such as the sequence of picking orders. RNNs are designed to capture and model dependencies in sequential data by maintaining a hidden state that evolves as the network processes each element in the sequence. Therefore, RNNs represent a promising approach when addressing problems involving the sequence of picking orders, as it has been already demonstrated by Suemitsu et al. (2022). However, traditional RNNs may face challenges such as vanishing gradients, limiting their effectiveness in handling long-term dependencies. Long Short-Term Memory (LSTM) networks, a specific type of RNN, address these issues by incorporating memory cells and gating mechanisms.

The recurrent layers were implemented as a stack of LSTM modules to capture sequential dependencies in the input data. Dropout was applied after each LSTM layer to prevent overfitting. Following the recurrent layers, the metamodel employed linear (fully-connected) layers. Dropout was also applied after each linear layer to further regularize the models. The metamodel had an output layer to produce the final output.

The number of recurrent and linear layers, their hidden sizes, and the dropout rates were defined during initialization. For the LSTM layers, weight initialization was performed using Xavier initialization for input weights and orthogonal initialization for hidden weights. For the linear layers and the output layer, weight initialization was again performed using Xavier initialization for hidden and constant initialization for biases. The forward pass involved passing the input through the LSTM layers, applying dropout, and then passing it through the linear layers with the corresponding dropout. Rectified linear unit activation functions were employed after each linear layer, contributing to the NN's non-linearity and expressive power. This activation function has become a popular choice in NN architectures due to its simplicity and effectiveness in mitigating the vanishing gradient problem. Figure 2 shows the architecture of the metamodel inspired by Suemitsu et al. (2022).

4.3 Feature Generation

The DES model was used to simulate random order sets and to track the makespan. Each simulation began by loading a random scenario. More specifically, the initial full capacity of the warehouse was randomly varied between 50 % and 80 %. Considering both the available capacity and the demand value of each SKU (Section 3.2), each UL was assigned a specific SKU in a predetermined quantity. In addition, the position of the HMs, denoted by their last visited storage location, was randomly selected from all available storage locations within the HM aisle. Then, an order set composed of a random number of orders between 10 and 50 was generated. Each order was composed of several order lines between 1 and 10. The quantity of each SKU requested in the order line was randomly sampled from a normal distribution, with a mean equivalent to the SKU demand rate and standard deviation set to 20 % of the demand rate.

In the quest to distill meaningful insights from the raw order data, 33 features were generated from them. Two main classes of features can be recognized, calculated for each class of products (3 classes). The product diversity class consisted of a set of features representing the nature and distribution of products within the order. To extract relevant information, the products were ordered by the demand rate, and their ID was encoded with a sequential integer number. Therefore, products with a lower ID were products with higher demand rates. After that, the features were generated. The number of unique products in an order was calculated (1 feature), shedding light on the diversity of items requested by customers. Moreover, a range of statistics, including the sum, minimum, average, maximum, and standard deviation of product IDs (5 features), unveiled the structure of the orders. The second class can be referred to as quantity dynamics, which describes the dynamics of order sizes. It comprises features like the sum, minimum, average, maximum, and standard deviation of product quantities (5 features). This information was crucial for gauging the distribution and variability in the quantities of products requested.

The simulation was run until the last order line of the last order was completely picked. The makespan was traced to be later used as the target variable for the metamodel. Because of the stochasticity of the DES model, each order set was replicated 100 times. The initial state of the system, such as the composition of the order set, the position of the HMs, the position of the ULs within the racks, and the content of each UL, was kept constant between the 100 replicates. The results of the replicates were then averaged and an average makespan was calculated. A total of 3,500 independent order sets were simulated. Therefore, the initial dataset was composed of 3,500 simulations · 100 replicates = 350,000 data points. The sample size was chosen to ensure a comprehensive representation of the parameter space, enabling thorough exploration of various operational scenarios. Furthermore, the random generation of input states facilitated the evaluation of a wide range of conditions. Therefore, selecting 3,500 independent order sets struck a balance between acquiring statistically robust data and managing computational resources, thereby facilitating a practical and feasible approach to model development and analysis. Given the stochastic nature of the DES model and the multitude of possible parameter combinations, this sample size was deemed adequate to capture the inherent variability in MLS system operations. After removing outliers, the dataset was divided following the 80–20 rule: 80 % of the sample was used for training and validation, while the remaining 20 % was used to test the trained metamodel on unseen data and to evaluate its performance.

LSTM used sequential input data whose shape is (N_b, N_s, N_f), where N_b is the number of picking order sets, N_s is the maximum sequence length, and N_f is the number of the features. In this experiment, N_s = 50 orders and N_f = 33 features were set to cover all the training dataset.

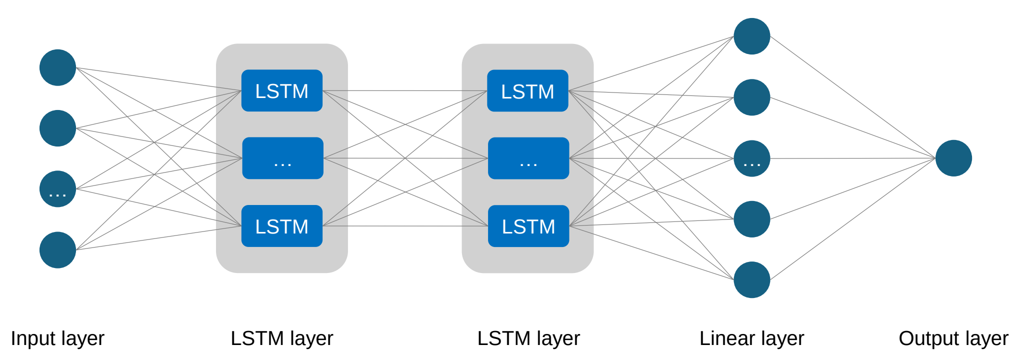


Figure 2: The structure of the metamodel.

5 RESULTS

5.1 Metamodel Training and Validation

The training process commenced by initializing crucial hyperparameters that define the architectural and learning characteristics of the metamodel. The hyperparameters encompassed the configuration of LSTM layers (number and size), their bidirectionality, the configuration of linear layers (number and size), the configuration of dropout layers (number and dropout rates), learning rate, batch size, and weight decay for regularization. Additional hyperparameter values that could have been investigated during the hyperparameter search were defaulted to the values specified by the PyTorch library. Hyperparameter tuning was conducted by a preliminary screening of the best combination of hyperparameters via random search, followed by manual fine-tuning by leveraging prior knowledge from tested combinations.

Before the beginning of the training, the input data underwent a min-max normalization to ensure consistent and stable training across varying scales of input features. The 5-fold cross-validation strategy, which mitigates the risk of overfitting and provides a more trustworthy evaluation of the metamodel's generalization capabilities, was adopted to robustly assess the metamodel's performance. The heart of the training process was in an iterative training loop executed for each fold. Batches of data were fed into the metamodel, and the training loop involved calculating the loss, performing backpropagation, and updating the metamodel parameters using the Adam optimizer. Mean Absolute Error (MAE) was chosen as the loss function, because it provides a simple and interpretable measure of prediction accuracy by capturing the average magnitude of errors, making it a robust metric for evaluating regression models. To avoid overfitting, L2 regularization was integrated into the training loop. Throughout the training process, the Mean Absolute Percentage Error (MAPE) was also tracked. Table 3 summarizes the hyperparameters of the metamodel, the range of values searched during the tuning of the metamodel, and the final value chosen to train the metamodel.

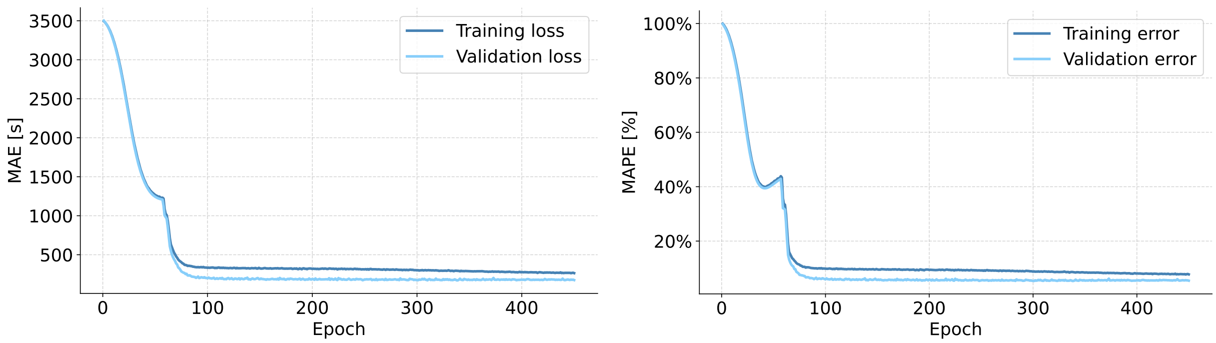
Table 3: Metamodel hyperparameters.

Hyperparameter	Range of values	Value selected
Hidden LSTM layers	1, 2	2
Hidden LSTM sizes	16, 24, 32, 40, 48, 56	16, 16
LSTM bidirectionality	True, False	True
LSTM layers dropouts	0 %, 10 %, 20 %, 30 %	0 %, 0 %
Hidden linear layers	1, 2	1
Hidden linear sizes	16, 24, 32, 40, 48, 56	24
Linear layers dropouts	0 %, 10 %, 20 %, 30 %, 40 %, 50 %	10 %
Learning rate	50 logarithmically spaced points between 0.0005-0.05	$7.5 \cdot 10^{-4}$
Batch size	16, 32, 64	32
Epochs	200, 250, 300, 350, 400, 450	450
Weight decay (L2 regularization)	$1 \cdot 10^{-5}$, $1 \cdot 10^{-4}$, $1 \cdot 10^{-3}$, $1 \cdot 10^{-2}$	$1 \cdot 10^{-4}$

Figure 3 presents the training and validation curves. The curves represent the averaged results between the 5 folds. The metamodel exhibits small MAE and MAPE values, indicating its proficiency in predicting the total makespan for completing a sequence of picking orders. The slightly higher training MAE and MAPE values compared to the validation values are due to the absence of regularization during validation inference.

5.2 Metamodel Testing

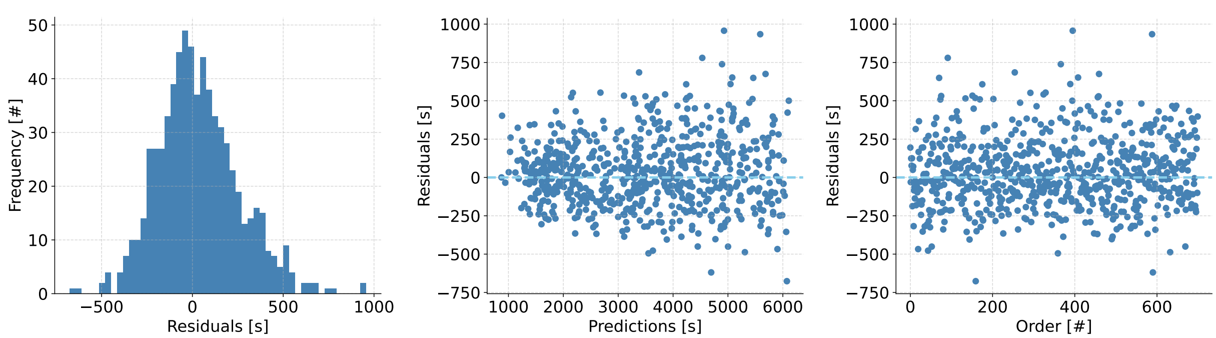
The predictive capabilities of the metamodel were rigorously evaluated on the test dataset, which consisted of data that had not been used during the training and validation phases and, thus, had not been seen by the metamodel until the testing phase. During this crucial evaluation step, an analysis of the residuals, which represent the deviations between the metamodel predictions and the actual target values, was performed.



(a) (b)

Figure 3: Training and validation curves: a) Mean Absolute Error; b) Mean Absolute Percentage Error.

The comprehensive insights into the metamodel's performance are illustrated by graphical plots (Figure 4). The histogram in Figure 4a illustrates the frequency distribution of the residuals. The nearly normal distribution observed in the residuals confirms the reliability of the prediction intervals and underscores the robustness of the metamodel. Figure 4b presents a scatter plot of the residuals against the predictions, revealing good homoscedasticity across all the predictions. Furthermore, Figure 4c presents a dispersion of the residuals around zero, with no discernible correlation between them. In conclusion, the analysis of the residuals strongly suggests that the metamodel performed well on the test data. In addition to the residual analysis, the performance of the metamodel was also assessed by calculating MAE and MAPE key metrics. Upon examination of the entire test dataset, the MAE was found to be 175.80 s, while the MAPE was 5.41 %. The metrics results obtained provide evidence supporting the metamodel capability to reliably predict the total makespan of order set fulfillment, suggesting that it exhibited good predictive performance within the scope of the analysis.



(a) (b) (c)

Figure 4: Residuals plots: a) histogram of residuals; b) residuals vs predictions; c) residuals vs order.

5.3 Models Comparison

One of the objectives of this study is to develop and evaluate a metamodel designed to replace the computationally intensive and resource-demanding DES model. To demonstrate the metamodel's superior performance in predicting the makespan of picking orders, the time required to complete the simulation of an order set using the DES model was compared to the time needed to predict the makespan using the already trained and tested NN. Various scenarios were considered, including order sets of varying sizes (10, 20, 30, 40, 50) and different warehouse capacities (50 %, 65 %, 80 %). The results of this comparison are presented in Table 4. The simulation runs and predictions were performed on a computer with an Intel Core i9-13900K 3.00 GHz 32-core CPU and 64 GB RAM. The findings demonstrate that the metamodel performs better in predicting the makespan in terms of time required. This efficiency supports the adoption of the metamodel for facilitating quicker decision-making while maintaining a good balance between predictive accuracy and computational efficiency.

Table 4: Discrete event simulation model and metamodel comparison.

Number of orders	Warehouse capacity	DES time [s]	Metamodel inference time [s]	Difference %
10	50 %	1.21958	0.00215	5.67E+04
10	65 %	1.15768	0.00247	4.68E+04
10	80 %	1.04001	0.00238	4.36E+04
20	50 %	2.40945	0.00221	1.09E+05
20	65 %	2.28850	0.00247	9.26E+04
20	80 %	1.90644	0.00230	8.29E+04
30	50 %	3.47374	0.00221	1.57E+05
30	65 %	3.35356	0.00250	1.34E+05
30	80 %	2.62179	0.00241	1.09E+05
40	50 %	4.89953	0.00226	2.17E+05
40	65 %	4.40991	0.00239	1.84E+05
40	80 %	3.52436	0.00224	1.58E+05
50	50 %	6.15346	0.00223	2.76E+05
50	65 %	5.66116	0.00242	2.33E+05
50	80 %	4.42533	0.00234	1.89E+05

6 DISCUSSION AND CONCLUSIONS

This study focuses on the OPP in the context of AS/RS. The primary objective is to propose the development of an NN-based metamodel tailored to predict the total time required to complete a sequence of picking orders. This metamodel, developed as a surrogate model for a DES model representing the target system, leverages an advanced architecture composed of LSTM and linear layers. While it is important to acknowledge the complexity of predictive modeling and the potential for variability within different datasets or scenarios, the results obtained bolster the confidence in the metamodel ability to effectively estimate the time required for order fulfillment. The findings contribute valuable insights into the metamodel's practical applicability and its potential to enhance operational efficiency in real-world settings.

It is important to clarify that this study focuses on the development and evaluation of a metamodel designed to serve as a replacement for the computationally intensive and resource-consuming DES model, by using simulation-generated data. The goal of this approach remains to provide a practical solution that strikes a balance between predictive accuracy and computational efficiency, thereby facilitating faster decision-making processes in complicated operational scenarios. In practical settings, where rapid predictions are often essential, the metamodel offers a lighter, faster, and more cost-effective alternative. Even if the metamodel may sacrifice some degree of explainability inherent in deep NN, it is crucial to note that its development and training were rooted in a comprehensive DES model which incorporates domain-specific intricacies like UL selection, task sequencing, and SKU allocation. Consequently, the metamodel essentially integrates the granular details of the lower-level DES model, providing a higher-level perspective that captures the nuanced dynamics of the system. It can be stated that from a theoretical point of view, the use of a NN-based metamodel is a step forward in the modeling of the complex dynamics of AS/RS. This metamodel not only demonstrates predictive capabilities for order completion times, but also serves as a valuable tool for exploring more-advanced challenges arising from the study of automated warehouse systems. In practice, the metamodel provides tangible benefits for operational decision-making in AS/RS. Accurate predictions could provide logistics managers with valuable insights for improving workflow efficiency.

However, the study acknowledges some limitations. For instance, the simulation model did not take into account disruptive mechanical issues, such as component repairs within the warehouse system, that could heavily impact maintenance activities. In addition, the DES model operates under the assumptions that inbound and outbound processes are not considered, and stock-outs of SKUs do not occur. Future studies could benefit from relaxing these assumptions to evaluate the approach reliability in more-realistic scenarios, where such events can occur and impact warehouse operations. Moreover, the metamodel was trained only with data relative to the picking orders, without introducing contextual variables such as the initial location of HMs, stock availability, and SKU distribution within racks and aisles. Furthermore, the study did not theorize on the generalizability of the findings to other types of warehouses or logistics systems. Exploring how well the proposed metamodel and methodology might adapt to different contexts would enhance the study's broader relevance and applicability.

For future applications, the metamodel shows promise for solving stochastic optimization problems that arise in automated warehouses, such as the order-sequencing problem. Optimization techniques such as genetic algorithms, simulated annealing, or tabu search could be used to explore the solution space while the metamodel evaluates the objective function based on the proposed solution. To increase the robustness and applicability of the metamodel, further research avenues can also be explored. First, the performance of the metamodel could be compared to other machine learning models. Then, integrating stochasticity into the predictions, expanding the initial dataset for training purposes, and incorporating contextual data could also help refine the performance of the metamodel. In addition, metamodels for predicting the occurrence of deadlocks or other system variables like the energy consumption could also be explored based on the architecture presented in this research. This suggests ongoing opportunities for study and improvement in the optimization of picking processes within AS/RS.

REFERENCES

Azadeh, K., R. De Koster, and D. Roy. 2019. "Robotized and Automated Warehouse Systems: Review and Recent Developments". *Transportation Science* 53(4):917–945.

Boysen, N. and K. Stephan. 2016. "A Survey on Single Crane Scheduling in Automated Storage/Retrieval Systems". *European Journal of Operational Research* 254(3):691–704.

Chirici, L. and K.-S. Wang. 2014. "Tackling the Storage Problem Through Genetic Algorithms". *Advances in Manufacturing* 2:203–211.

Dijkstra, A. S. and K. J. Roodbergen. 2017. "Exact Route-Length Formulas and a Storage Location Assignment Heuristic for Picker-To-Parts Warehouses". *Transportation Research Part E: Logistics and Transportation Review* 102:38–59.

Epp, M., S. Wiedemann, and K. Furmans. 2017. "A Discrete-Time Queueing Network Approach to Performance Evaluation of Autonomous Vehicle Storage and Retrieval Systems". *International Journal of Production Research* 55(4):960–978.

Fandi, W., S. Kouloughli, and L. Ghomri. 2022. "Multi-Shuttle AS/RS Dimensions Optimization Using a Genetic Algorithm – Case of the Multi-Aisle Configuration". *The International Journal of Advanced Manufacturing Technology* 120(1–2):1219–1236.

Ferrari, A., A. Carlin, C. Rafele, and G. Zenezini. 2023. "A Method for Developing and Validating Simulation Models for Automated Storage and Retrieval System Digital Twins". *The International Journal of Advanced Manufacturing Technology* 131:5369–5382.

Ferrari, A., G. Zenezini, C. Rafele, and A. Carlin. 2022. "A Roadmap Towards an Automated Warehouse Digital Twin: Current Implementations and Future Developments". *IFAC-Papersonline* 55(10):1899–1905.

Gagliardi, J.-P., J. Renaud, and A. Ruiz. 2015. "Sequencing Approaches for Multiple-Aisle Automated Storage and Retrieval Systems". *International Journal of Production Research* 53(19):5873–5883.

Giner, J., D. Katic, K. Kovacs, R. Glawar, and W. Sihn 2023. "A Computer Vision Based Approach to Reduce System Downtimes in an Automated High-Rack Logistics Warehouse". *Procedia CIRP* 118:1078–1083.

Hausman, W. H., L. B. Schwarz, and S. C. Graves. 1976. "Optimal Storage Assignment in Automatic Warehousing Systems". *Management Science* 22(6):629–638.

Hsu, H.-P. and C.-N. Wang. 2023. "Hybridizing Whale Optimization Algorithm with Particle Swarm Optimization for Scheduling a Dual-Command Storage/Retrieval Machine". *IEEE Access* 11:21264–21282.

Kirchhoff, D., M. Kirberg, S. Kuhnt, and U. Clausen. 2023. "Metamodel-Based Optimization of Shift Planning in High-Bay Warehouse Operations". *Quality and Reliability Engineering International* 39(2):590–608.

Law, A. M. and W. D. Kelton. 2007. *Simulation Modeling and Analysis*. 3 ed. Mcgraw-hill New York.

Lehmann, T., J. Huß, and Others. 2021. "Travel Time Model for Multi-Deep Automated Storage and Retrieval System with a Homogeneous Allocation Structure". *Logistics Research* 14(1).

Li, L., Z. Chen, and T. Zhang. 2020. "Hybrid-Grouping-Matching Heuristic for the Storage Assignment in Gravity AS/RSs with Duration-of-Stay Storage Policy". In *Proceedings of the 4th International Conference on Computer Science and Application Engineering*, 18. New York, NY: ACM.

Li, Y., R. Zhang, and D. Jiang. 2022. "Order-Picking Efficiency in E-Commerce Warehouses: a Literature Review". *Journal of Theoretical and Applied Electronic Commerce Research* 17(4):1812–1830.

Liu, J., H. Liao, and J. A. White Jr. 2021. "Stochastic Analysis of an Automated Storage and Retrieval System with Multiple In-The-Aisle Pick Positions". *Naval Research Logistics (NRL)* 68(4):454–470.

Ma, W., D. Yang, Y. Wu, and Y. Wu. 2023. "Order Dividing Optimisation in Crane & Shuttle-Based Storage and Retrieval System". *Journal of Control and Decision* 10(2):215–228.

Mahdavisharif, M., A. C. Cagliano, and C. Rafele. 2022. "Investigating the Integration of Industry 4.0 and Lean Principles on Supply Chain: A Multi-Perspective Systematic Literature Review". *Applied Sciences* 12(2):586.

Marolt, J., N. Kosanić, and T. Lerher. 2022. "Relocation and Storage Assignment Strategy Evaluation in a Multiple-Deep Tier Captive Automated Vehicle Storage and Retrieval System with Undetermined Retrieval Sequence". *The International Journal of Advanced Manufacturing Technology* 118(9):3403–3420.

Nag, B. and R. Pal. 2022. "Simulation Optimization for Supply Chain Decision Making". In *2022 Winter Simulation Conference (WSC)*, 2853–2863. IEEE.

Silva, A., K. J. Roodbergen, L. C. Coelho, and M. Darvish. 2022. "Estimating Optimal Abc Zone Sizes in Manual Warehouses". *International Journal of Production Economics* 252:108579.

Suemitsu, I., H. K. Bhamgara, K. Utsugi, J. Hashizume, and K. Ito. 2022. "Fast Simulation-Based Order Sequence Optimization Assisted by Pre-Trained Bayesian Recurrent Neural Network". *IEEE Robotics and Automation Letters* 7(3):7818–7825.

Tadumadze, G., J. Wenzel, S. Emde, F. Weidinger, and R. Elbert. 2023. "Assigning Orders and Pods to Picking Stations in a Multi-Level Robotic Mobile Fulfillment System". *Flexible Services and Manufacturing Journal*:1–38.

Vijayakumar, V. and F. Sgarbossa. 2021. "A Literature Review on the Level of Automation in Picker-To-Parts Order Picking System: Research Opportunities". *IFAC-Papersonline* 54(1):438–443.

Wang, B., X. Yang, and M. Qi. 2022. "Order and Rack Sequencing in a Robotic Mobile Fulfillment System with Multiple Picking Stations". *Flexible Services and Manufacturing Journal* 35:1038–1075.

Wang, W., Y. Wu, and Y. Wu. 2016. "A Multi-Stage Heuristic Algorithm for Matching Problem in the Modified Miniload Automated Storage and Retrieval System of E-Commerce". *Chinese Journal of Mechanical Engineering* 29:641–648.

Xu, J., E. Huang, C.-H. Chen, and L. H. Lee. 2015. "Simulation Optimization: A Review and Exploration in the New Era of Cloud Computing and Big Data". *Asia-Pacific Journal of Operational Research* 32(03):1550019.

Xu, X., X. Zhao, B. Zou, and M. Li. 2019. "Optimal Dimensions for Multi-Deep Storage Systems Under Class-Based Storage Policies". *Cluster Computing* 22:861–875.

Yang, P., G. Jin, and G. Duan. 2022. "Modelling and Analysis for Multi-Deep Compact Robotic Mobile Fulfilment System". *International Journal of Production Research* 60(15):4727–4742.

Yu, Y., Y. Liu, and H. Yu. 2022. "Optimal Two-Class-Based Storage Policy in an AS/RS with Two Depots at Opposite Ends of the Aisle". *International Journal of Production Research* 60(15):4668–4692.

Zhao, N. and M. Wang. 2021. "Research on Parameter Optimization of the Express Warehousing and Distribution System Based on the Box–Behnken Response Surface Methodology". *Advances in Civil Engineering* 2021:1–9.

Zhou, L., J. Zhao, H. Liu, F. Wang, J. Yang, and S. Wang. 2022. "Stochastic Models of Routing Strategies Under the Class-Based Storage Policy in Fishbone Layout Warehouses". *Scientific Reports* 12(1):12876.

AUTHOR BIOGRAPHIES

ANDREA FERRARI is a Ph.D. candidate in Management, Production and Design at the Department of Management and Production Engineering at the Politecnico di Torino. From August 2023 to March 2024 he was a visiting researcher at Boston University, Metropolitan College. His research activities are focused on the application of simulation approaches and statistical models to the field of industrial logistics systems and supply chains. His email address is andrea.ferrari@polito.it and his website is https://www.polito.it/personale?p=andrea.ferrari.

CANAN GUNES CORLU is an Associate Professor in the Administrative Sciences Department of the Metropolitan College at Boston University. She holds a Ph.D. in Operations Management from the Tepper School of Business at Carnegie Mellon University. Her research interest is in the area of the design and analysis of stochastic simulation with applications in operations management. Her email address is canan@bu.edu and her website is https://www.bu.edu/met/profile/canan-gunes-corlu/.