Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

CLOUD-BASED HYBRID SIMULATION MODEL FOR OPTIMIZING WAREHOUSE YARD OPERATIONS

Mohammed Farhan
Pascalin Ngoko
Farouq Halawa
Raashid Mohammed

Global Engineering Support Services
Amazon.com
410 Terry Ave N
Seattle, WA 98109, USA

ABSTRACT

Fulfillment centers in the E-commerce industry are highly complex systems that houses inventory and fulfill customer orders. One of the key processes at these centers involves translating customer demands into trucks and yard operations. Truck yards with operational issues can create delays in customer orders. In this paper, we show how a scalable cloud-based hybrid simulation model is used to improve yard operations, optimize flow and design, and forecast yard congestion. Cloud experimentation along with automated database connectivity allows any user to run simulation analyses to derive data driven operational decisions. We tested the model on two real world case studies, which results in cost savings for the organization. This paper also proposes a robust automated framework for setting simulation validation benchmarks and measuring model accuracy.

1 INTRODUCTION

Yard in a fulfillment center is an enclosed space outside a warehouse where customer packages get picked up or dropped off by trucks or other transportation vehicles. Yard congestion is a known issue in the supply chain industry that stems from situations like high peaks in volume or non-optimized scheduling or processes (Weisbrod and Fitzroy 2010). It could cause delays in customer orders, and can lead to blockage of public roads. Heavy truck congestion inside the facility can cause gridlocks (traffic jam) that can take hours to resolve. Yards being highly complex systems have conventional operational challenges like labor, scheduling, footprint constraints, etc.; over-the-road challenges like traffic, weather, equipment, etc.; and safety challenges from operating large equipment at all hours of the day in confined spaces. Yard systems are also unstable, where minor defects like downed equipment can quickly cascade to many other processes. Those minor defects can lead to quick materialization of gridlock, queuing, or other costly or customer-facing issues. To overcome these challenges, we often need high precision for strategic decisions about the yard operations. This high precision is not consistently achievable with simple historical benchmarks, conventional mathematical models, back-of-the-napkin math, or operator judgment. Therefore, there is a need to identify a tool/solution to establish a safe and seamless flow of trailers through yards, and increase end to end transportation effectiveness.

One of the widely used tool for risk management in the supply chain industry is Simulation Optimization (Oliveira et al. 2019). Lin and Cheng (2009) have showed a simulation framework to build rail yards to conduct what-if analyses on rail yard operations. Beaudoin et al. (2012) have used Discrete Event Simulation (DES) to determine optimal load count for truck allocation and improve yard operations. This paper covers the build of a real-world truck yard using hybrid simulation modeling (agent-based and discrete-event). The yard simulation model acts as a framework to scale multiple yards at ease. This paper highlights the power of cloud computing in simulation modeling. It can improve the performance and scalability of simulation models by providing access to vast computing resources on demand. Users can access cloud-based simulation models from anywhere with an internet connection, enabling collaboration and facilitating remote work. Taylor et al. (2014), Guo et al. (2011), and Cayirci (2013) all mention the use of simulation as a cloud service, but no research was found discussing this approach in industrial truck yard optimization.

With an increased popularity for simulation modeling, the validity of models is crucial for acquiring user trust and credibility. However, simulation model validation is frequently not performed or reported by researchers (Farhan et al. 2021). In Brailsford et al. (2019), 18 simulation papers out of 69 papers reviewed have shown validity by some standard statistical methods. In this paper, we also propose an automated framework for validating time-series simulation models using a two-phase validation method. The validation framework can apply to other time-series simulation models.

Section 2 provides a brief description of the hybrid yard simulation model, along with the cloud experimentation framework. Section 3 covers the model validation methodology. Section 4 showcases two use cases of the cloud-based yard simulation model, and we conclude the paper in Section 5.

2 CLOUD-BASED HYBRID SIMULATION MODEL

This section covers the hybrid agent-based and discrete-event simulation model build and the cloud-based experimentation framework. Hybrid simulation technique enables modelers to accurately represent complex systems in a digital environment. The cloud-based experimentation allows multiple users to run simulation experiments and make data driven informed decisions. The model boundaries make up of only the yard operations of the facility and do not account for inside the roof operations. We built the model using AnyLogic 8 Professional software. In this section, we display a novel approach to create a base simulation model depicting all yard operations. The base model acts as a framework to scale multiple different yards in less than two hours. Each yard comprises check-in and check-out guard shacks, dock doors, parking spots, hostler trucks, bobtail trucks, boxtrucks and trailers.

2.1 Agent-Based Model Structure

We modeled all elements with individual characteristics as agents, as shown in Figure 1. *BaseTruck* agent defines the basic characteristic of all moving entities with parameters such as speed and truck rotation angles. *BaseTruck* agents act as transporters (vehicles used to transport items), moving trailers from source to destination. *Truck* agent (bobtail) has the functionality to pick up or drop trailers of size 53 feet. *ParkingSpots* agent is parking location for a single trailer. *ParkingArea* defines a collection of *ParkingSpots* and differentiates between parking and dock doors (trailer processing location). *Hostler* is a type of transporter that resides solely inside the yard and helps move trailers in the yard. *Boxtrucks* are a type of truck that always has a fixed trailer size of 26 feet. Each *Trailer* can carry a varying number of packages inside or outside the yard. *GuardShack* agents check trucks and boxtrucks in and out of the yard. Each guard shack can have multiple lanes to allow multiple trucks to check in/out. *RefNetwork* agent allows modelers to create a new *yardNetwork* with unique configurations.

2.1.1 RefNetwork Agent

With many yards in the fulfillment network to model, building a framework for scalability and quicker build time is crucial. We got yard images along with yard dimensions using google maps or nearmap. The yard image covers the aerial view of the yard, including the nearest public road for entry and exit. With the help of an intake form provided to every facility, we manually recorded the network for truck movement (one way or two-way), location of gates, count of dock door and unique spatial characteristics of yards.

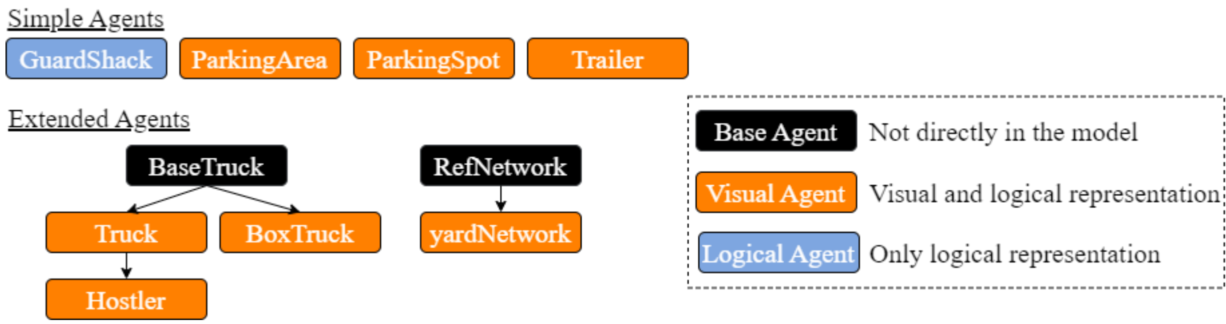

Figure 1: Model agent's structure.

2.2 Discrete-Event Process Flow

To understand how the yard operates, we performed field trips to multiple yards. Conversations with several yard operators resulted in building process flow maps for all elements in the yard. We translated these flowcharts into the model as model blocks using process and transporter libraries. Figure 2 shows the process map from an inbound loaded truck's perspective. Trucks enter the yard based on the time specified in the arrival schedule database. Live loads are 53FT trucks that do not drop trailers in the parking spots and directly process loads at dock doors. Depending on the availability of the specific load type dock door, the truck drops the trailer at the dock door. If no dock doors are available, the trucks look for an available parking spot. If no parking spots or temporary parking spots are available, the truck looks for a secured offsite yard. Secured offsite yards are big parking lots used to house loaded trailers. If secured offsite spots are not available, the model would "gridlock" because of no parking spot availability and provide details on what caused the gridlock. We created similar process flows for outbound flow, live load, empty trailer moves, hostler, inbound and outbound boxtrucks, and offsite yard process.

2.2.1 Transporter Movement

Trucks find the shortest path to reach their destination with the help of network paths and nodes. Transporters recognize each other as obstacles and avoid collisions by automatically decelerating or speeding up. We drew all primary paths (paths that connect parking areas, dock areas, guard shacks, and hostler parking areas) manually for each yard by referring to the intake form received from the facility. The model generated all secondary paths (paths used for parking) and parking nodes through logic.

2.3 Model Inputs

Beyond the manual spatial inputs received from intake forms, model ingested data inputs using external data tables: Arrival schedule of trucks provides information on the type and quantity of volume, processing time for trailers, and type of truck (53FT or box trucks). Yard configuration table has information on active hostler truck counts, availability of external parking areas (offsite yard), number of check-in/out gates and lanes. Yard initialization table has information on the current state of the yard such as parking spots being occupied or empty. All simulation models run would initialize the current state of the yard based on the specified start date in the model to avoid running an arrival schedule without existing trailers in the yard. Fulfilment centers have specific dock doors to handle specific type of volume such as fluid loads, palletized loads, and non-inventory loads. Dock door configuration table provides historical information on what loads can specific dock doors process so trucks can only drop off trailers to those specific doors.

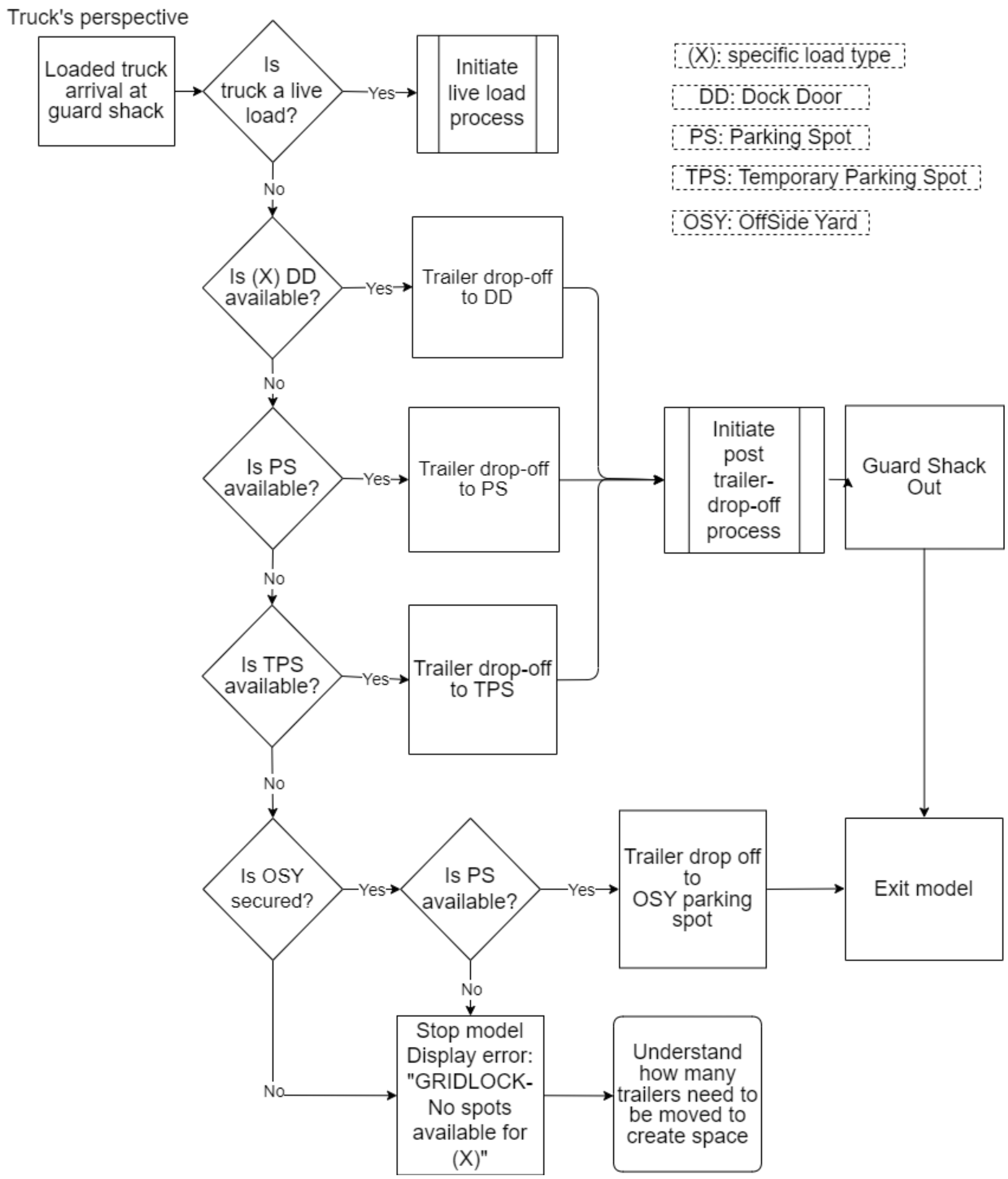

Figure 2: Inbound truck process flowchart.

2.4 Database Connection

All the data tables mentioned in Section 2.3 refresh on a daily cadence and reside on Amazon Redshift servers. Amazon Redshift is a data warehouse product which forms part of the larger cloud-computing platform Amazon Web Services (AWS) (Amazon Web Services, 2023). To enable automation for data recovery, we used the database connectivity module of AnyLogic Professional software. We queried the data tables from Amazon Redshift into the yard simulation model, as shown in Figure 3.

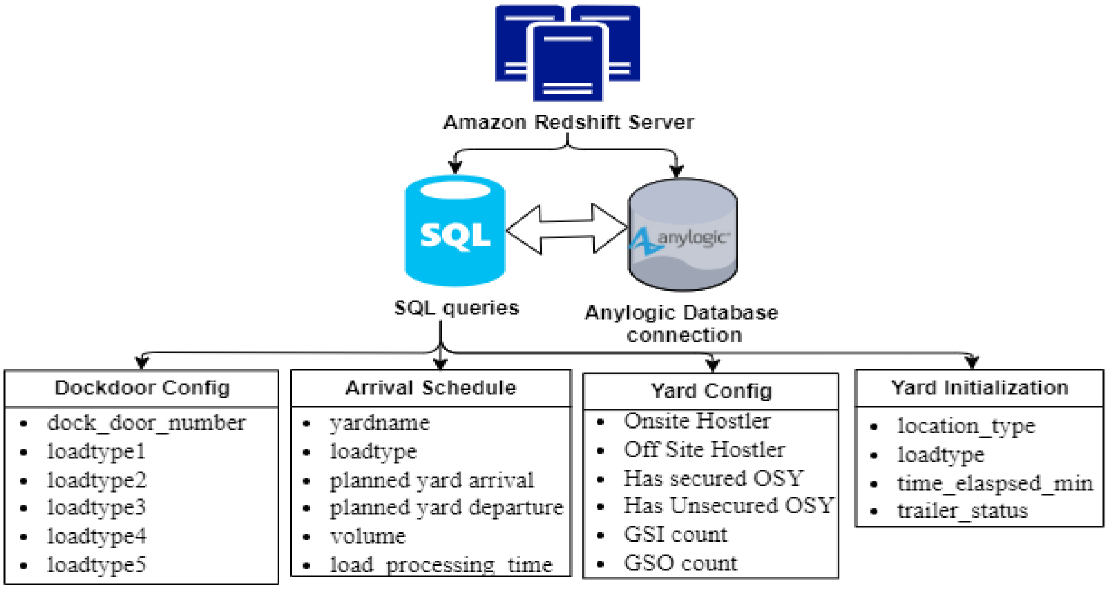

Figure 3: Database connectivity.

2.5 Self-Service Cloud Interface

The yard simulation model runs strategic analyses and answers questions such as - What is the breaking point of a yard with current resources/schedules? What is the optimal amount of hostler trucks required with increased volume? Will a certain yard see gate queue issues for a forecasted volume schedule?

The aim of the cloud platform was to provide a scalable solution that hundreds of users can access at a time. Cloud offers self-service capability that allows users to run yard simulation experiments. It enables them to make data backed decisions within a matter of minutes instead of relying on a simulation modeler. It also offers functionality to allow users to programmatically schedule simulations on a regular cadence, allowing them to focus on analyzing the results and solving problems.

2.5.1 Cloud Framework

Amazon's AWS tool Elastic Compute Cloud (EC2) offers a broad set of choices from processing speed, storage, and operation system to host applications on the cloud (Amazon Web Services, 2023). We installed a private instance of AnyLogic Enterprise Cloud on EC2 providing a secure and private cloud interface for users, as shown in Figure 4. Like the off-web model, the yard simulation model on the cloud can also connect to Redshift servers for data. We provide users with a webpage link and login credentials to access the cloud interface. Because of the end user's needs, there was a deficiency on the default AnyLogic cloud User Interface (UI) that hinders interpretability and ease of use. To overcome this, a custom UI which is flexible and meets the end user's needs was implemented. We built the custom UI using HTML, JavaScript, Custom CSS, and Bootstrap. The custom UI API was uploaded to the existing AnyLogic Cloud UI.

2.5.2 Experimentation and Model Run

Users can run experiments on the cloud by entering information such as yard name, start and end date for historical run. They can also provide additional experimental inputs to run personalized experiments such as increasing volume, increasing hostler count, changing check-in/check-out processing times and many more. Once each experiment run completes, the cloud platform stores simulation output for easy retrieval.

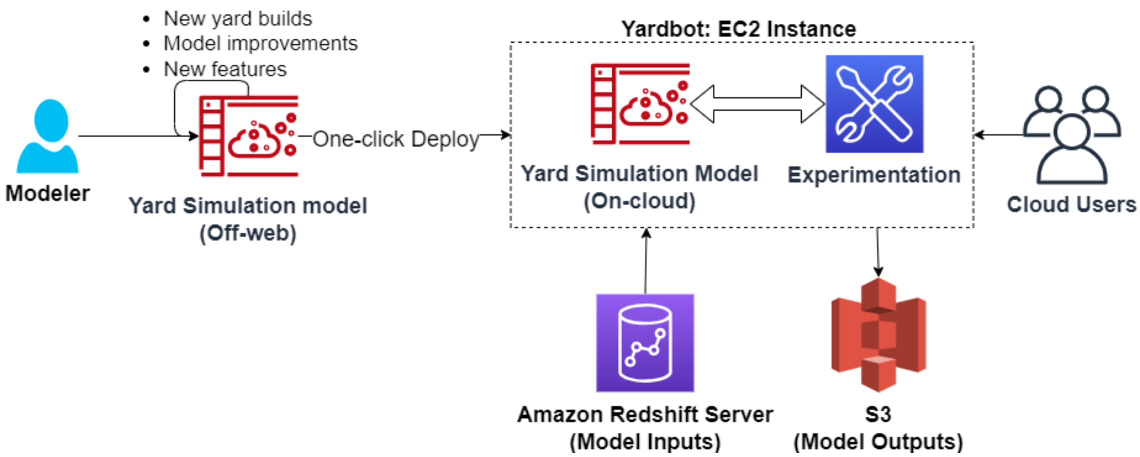

Figure 4: Cloud framework.

We also save the outputs on the Amazon's Simple Storage Service called S3 (Amazon Web Services, 2023) for quick reports. Users also can see the simulation run with all the yard elements to get a better understanding of yard performance, as shown in Figure 5.

Figure 5: Simulation run of a yard showing yard elements

2.6 Output Metrics

We leverage the yard simulation model to reduce operational yard flow risk and costs. Yard utilization provides an overall fullness of the yard. It is measured as occupied parking spots by overall parking capacity. If yard utilization exceeds 90 %, users would consider it as yard risk. Gate queue metric provides how many trucks are waiting to check-in and measures the time trucks wait before checking in. If the gate queue is longer than a set threshold for that yard, the trucks might hit the public road and cause road safety issues. Hostler utilization measures the fullness of hostler trucks in the yard. High hostler utilization can reflect sub-optimal hostler moves or shortage of hostlers for that site. Low hostler utilization could mean wastage of resources. Total daily volume provides the daily volume for the site and dock door utilization provides dock door fullness. The model also displays heatmap functionality to detect congestion in the yard and can provide direction for mitigation strategies.

3 MODEL VALIDATION

With multiple users using simulation modeling analyses for real-world operational decisions, there would always be trust concerns regarding its validity of the results. One of the most common approaches for validating simulation models against actual historical data is to apply statistical testing such as t-tests, chi-square, Kolmogrov-simirnov. However, the statistical tests assume IID data (Independent and identically distributed random variables) whereas in most cases, real-life applications are non-stationary (i.e., possess time-series nature) (Law and Kelton 2000). Applying time-independent statistical tests would invalidate the validation process. In this paper, we attempt to combine a few methodologies to get a robust two-phase mechanism to validate time-series models.

Law and Kelton (2000) proposed a technique to satisfy the IID (Independent and identically distributed random variables) assumptions when using t-test for time series data. According to them, the approach is to run *n* number of independent replications (preferable *n* >30) and compare that with *n* number of independent observations from the real-world system. Table 1 shows the required data to satisfy IID assumptions for the paper's use case. Method 1 considers different yards run across a single time period and Method 2 considers validation for a single yard (Yard *x*) across different schedules.

Table 1: Input data structure.

Method 1: Different Yard	**Method 2: Different Schedule (Yard *x*)**	**Simulation output**	**Historical output**
Yard 1	Schedule 1: May 1 - May 7	x_1	y_1
Yard 2	Schedule 2: May 8 - May 14	x_2	y_2
…	…	..	..
Yard 30	Schedule 30: July 15 - July 22	x_{30}	y_3

3.1 Phase 1: Overall Model Validation Using Confidence-Interval

In Phase 1, we apply a two-sample t-test with a 95 % confidence level for four hourly yard metrics: mean yard utilization, number of overall check-ins, number of overall check-outs and number of overall hostlers moves. Phase 1 provides a quick overview of the state of validation for the yard simulation model. The test was performed on a sample of 101 yards. We applied Method 1 from Table 1 and ran 101 yards once over a 7-day period (15th–22nd July 2022). The results showed that the output metrics "mean yard utilization", "total number of check-ins" and "total number of check-outs" passed the t-test. On the contrary, the test failed for "total number of hostler moves", as shown in Figure 6 and requires investigation. We can therefore conclude for now that the simulation model can predict the first three output metrics accurately.

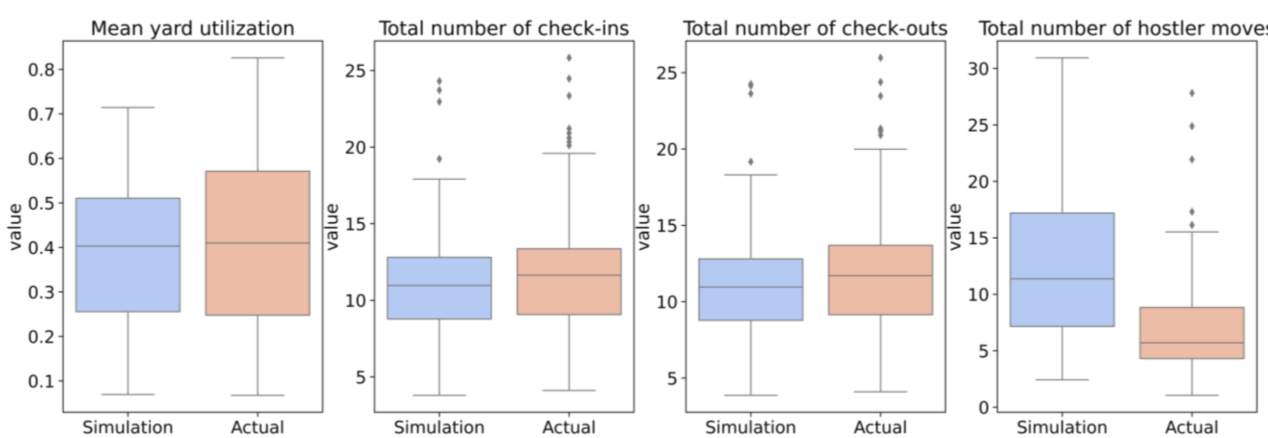

Figure 6: Box plots for the four output yard metrics using a representative sample of 101 yards. The figure testifies that the model is accurate for 3 out of 4 yard metrics.

3.2 Phase 2: Accuracy Score per Yard using AHP

For Phase 2, we recommend applying multiple statistical tests for higher reliability when calculating accuracy, as each test has different pros and cons. Accuracy tests are also sensitive to the output metric scales in data; data normalization helps prevent this issue. We propose applying a min-max method for data normalization as x–min (x) / (max (x)–min (x)), where x is a single observation. Phase 2 recommends using three accuracy tests that work well with standardized time series data: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Dynamic Time Warping (DTW). A common issue with using multiple tests and multiple metrics is the multi criteria decision problem. An end user would not be confident in looking at different results to make a fair decision. To overcome this issue, Analytic Hierarchy Process (AHP) helps simplify the problem by breaking it down into a hierarchical system and providing a single accuracy score for all output metrics and tests (Saaty, 1980). Figure 7 shows the AHP formulation for the accuracy score for the paper's use case. AHP also helps to provide a weighted pair-wise comparison between the output metrics and the accuracy tests used to select a model variant that achieves the highest accuracy score.

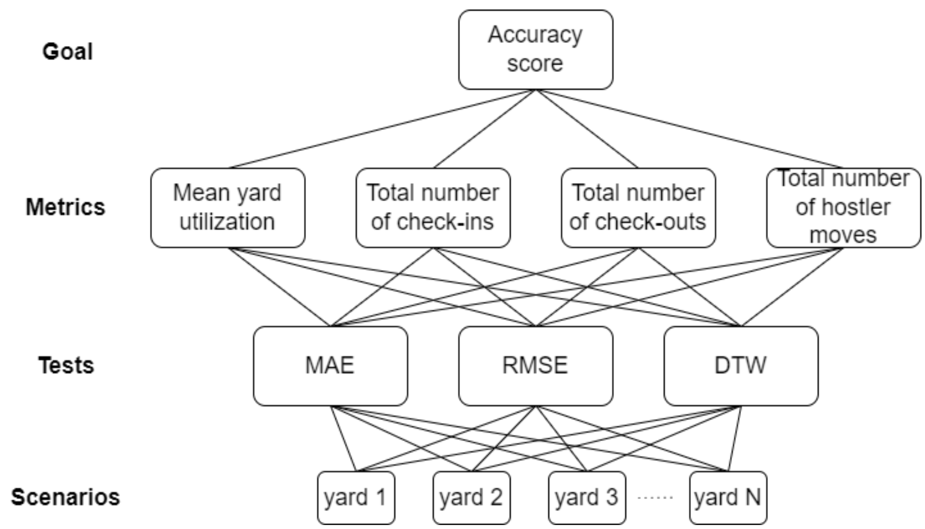

Figure 7: AHP formulation for yard simulation.

3.2.1 Setting Initial Thresholds

One drawback of using time-series data is that there are no well-established thresholds and these thresholds can be problem-dependent. For instance, predicting arrival time of trucks may have different thresholds than predicting forecasting weather data. The goal is to find an acceptable level of randomness when running multiple replications. In our use case, 3 replications of randomly selected simulation runs were run with different random seeds to create initial threshold levels for the accuracy tests. The error level resulted by the same model, but different runs should be an acceptable range to allow for model variability. We noticed that the initial thresholds for our use case were: DTW ranges between 6–12; MAE ranges between 2 %–4 %, and RMSE ranges between 2 %–5 %.

3.2.2 Validation Results

For yard level statistics, we use the AHP model to provide us with an accuracy score for each yard. We weighted each accuracy test equally, and a user can change that based on their use case and preference. The accuracy score has a scale between 0–1, 1 being the highest. Figure 8 shows a ranking of the accuracy for the tested 101 yards. We can further investigate yards with an accuracy ranking lower than 0.8 to determine the cause of discrepancy. We can assess individual yard output metrics to provide a holistic view of the situation. Figure 8 also shows a comparison between yards with the best and worst accuracy scores (Yard1 has the highest accuracy score of 1.0 whereas Yard101 has the lowest accuracy score of 0.68). To maintain validity, we should periodically perform this analysis to preserve the accuracy level of the simulation model.

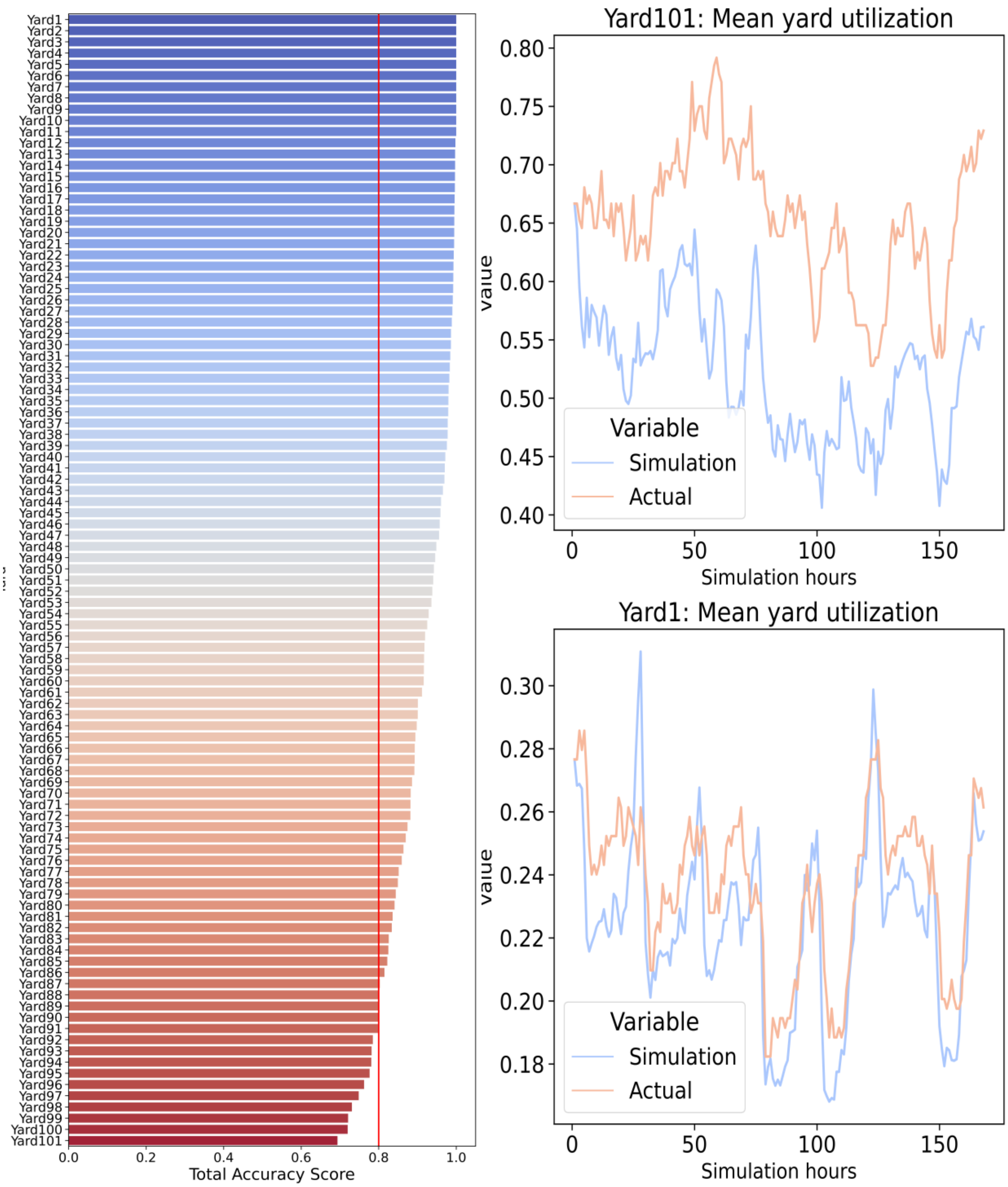

Figure 8: Phase 2 results displaying a summary of accuracy scores and yard with highest and lowest accuracy scores.

3.3 Accuracy Model Integration in Yard Simulation Model

We developed the accuracy model in Python. We developed the yard simulation model in AnyLogic software, which uses Java in its development environment. Because of the specific packages used in Python for model validation, it is hardly workable to translate it into Java to integrate it into the simulation model. Therefore, we imported the python code as an external package in AnyLogic. The code automatically runs at the end of the simulation run to get the validation and accuracy metrics.

4 CASE STUDIES

4.1 Case 1: Right-Sizing for Yard Capacity

Frequent high yard utilization leads to yard management teams requesting leadership to expand the yard by adding additional parking spots or providing off-site parking areas for trailers. An analytical justification for right-sizing prevents an unsuccessful venture and might prevent enormous costs to the organization. The cloud-based yard simulation model can assess the state of the yard with/without the expansion of parking spots. To determine the right-sizing for yard capacity, we ran a cloud experiment with increased parking capacity. To our surprise, we noticed even with increased parking spots, the yard still gridlocked as shown in the mean yard utilization graph of Figure 9. After further investigation, we observed the bottleneck for the yard was dock doors. High dock door utilization (<0.90) resulted in incoming trailers to fill up parking spots and waiting for a dock door to free up. We could pin-point the load type that was a constraint and informed yard management to optimize the dock door configuration to accommodate it. The analysis resulted in a no-go to expand the yard, preventing construction costs for the organization.

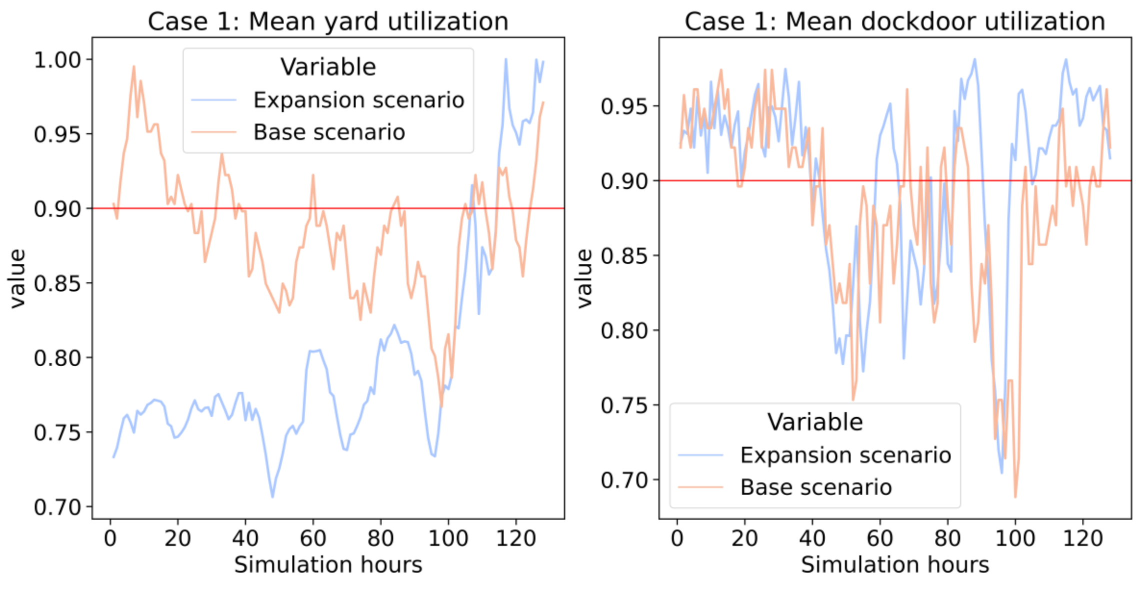

Figure 9: Case 1 results showing dock doors as a bottleneck for the yard.

4.2 Case 2: Cloud-Based Optimal Hostler Count

Resource and labor planning play an important role in yard operations. Insufficient resources such as hostlers can lead to delays in trailers meeting their deadlines for processing. An abundance of hostlers on site would mean an inefficient use of hostlers and labor. Requirements of these resources may also fluctuate depending on the time of the year or day (holiday season). Prior planning for such events prevents these issues. Yard management can use the cloud platform in advance to determine the optimal hostlers needed for each hour. Users can choose different hostler configurations to run the simulation experiment and can also perform a comparison between runs directly on the cloud. Users would look at the hostler utilization and yard utilization output metric to determine the optimal number of hostlers needed. As shown in Figure 10, for a specific yard, the yard would gridlock in the middle of the run when there is only one hostler on site. Even with two hostlers, though the model runs to completion, there is a risk for yard gridlock in the future, as seen in the mean yard utilization graph. For this yard, the user would choose three hostlers as an optimal solution for the scenario.

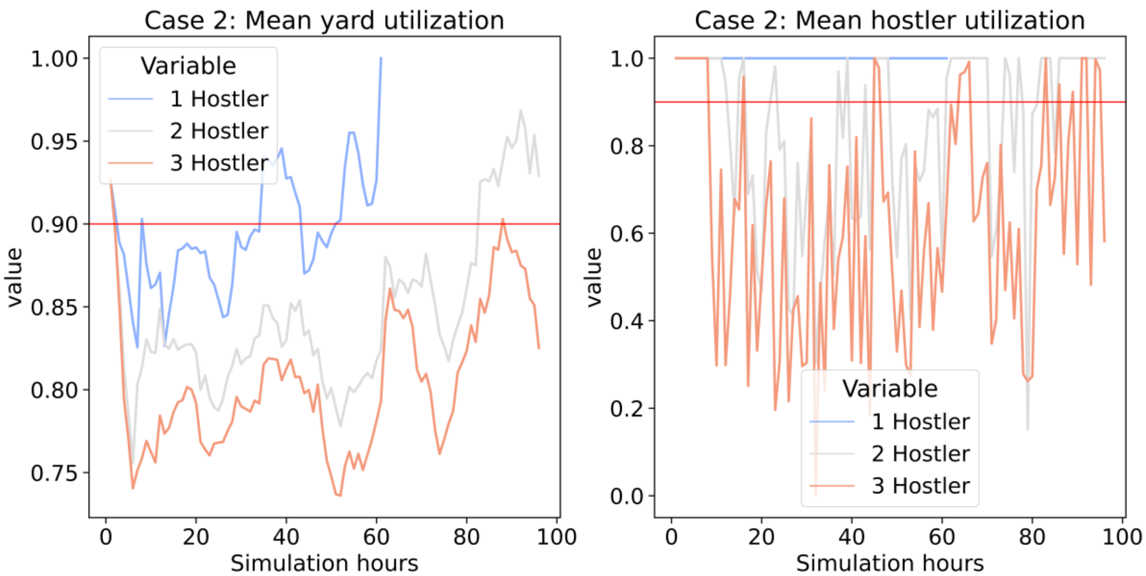

Figure 10: Case 2 results showing three hostlers as an optimal solution.

5 CONCLUSION

This paper shows how a cloud-based yard simulation modeling is used to predict and forecast gridlock risk in truck yards. The yard simulation model is a hybrid simulation model (Discrete-event and Agent-based) and acts as a base framework model to scale multiple different yards. We highlighted how database connectivity in simulation models automate inputting data and speeding up new yard scale time. A self-service cloud platform acts as a scalable solution, allowing multiple users without modeling expertise to run simulation experiments and make data driven results. We also covered the validation of the yard simulation model by showcasing a new two-phase approach to get a single accuracy score using AHP. The two-phase method provides ease in pin-pointing discrepancies in data and model errors and provides modelers a benchmark score for building more accurate models. This validation technique can also apply to other simulation models that use historical time series data. Finally, we displayed two use cases of the cloud-based yard simulation model that helped an organization prevent costs in new equipment and expansion projects.

In future scope, the cloud-based yard simulation model has a capability to incorporate machine learning expertise to automate the optimization process. User would click a button to automatically determine the optimal resources needed for a scenario. Machine learning can also generate schedules for new yards without historical data, forecast volume, predict load processing times, and optimize max arrivals thresholds for yards at any hour.

ACKNOWLEDGMENTS

The model build was facilitated by SimWell members Kevin Perez and Sebastian Mihalko. A special thanks to the AnyLogic software team and Amazon's AWS team for assisting on private cloud computing framework.

REFERENCES

Amazon Web Services 2023. "Amazon Redshift". https://aws.amazon.com/redshift/, accessed 16th February 2023.

Amazon Web Services 2023. "EC2". https://aws.amazon.com/ec2/, accessed 16th February 2023.

Amazon Web Services 2023. "S3". https://aws.amazon.com/s3/, accessed 16th February 2023.

Beaudoin, D., L. LeBel, and M. A. Soussi. 2012. "Discrete Event Simulation to Improve Log Yard Operations". *INFOR: Information Systems and Operational Research* 50(4):175–185.

Brailsford, S. C., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2019. "Hybrid Simulation Modelling in Operational Research: A State-of-the-Art Review". *European Journal of Operational Research* 278(3):721–737.

Cayirci, E. 2013. "Modeling and Simulation as a Cloud Service: A Survey." In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 389–400. Piscataway, New Jersey: IEEE.

Farhan, M., C. Krejci, M. Olsen, and M. S. Raunak. 2021. "Metamorphic Testing for Hybrid Simulation Validation". In *Proceedings of the Annual Modeling and Simulation Conference (ANNSIM)*, July 19th –2nd, Virginia, USA, 1–12.

Guo, S., F. Bai, and X. Hu. 2011. "Simulation Software as a Service and Service-Oriented Simulation Experiment". In *Proceedings of the 2011 IEEE International Conference on Information Reuse and Integration*, 113–116. Piscataway, New Jersey: IEEE.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling & Analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Lin, E., and C. Cheng. 2009. "YardSim: A Rail Yard Simulation Framework and its Implementation in a Major Railroad in the U.S.,". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls. 2532–2541. Piscataway, New Jersey: IEEE.

Oliveira, J. B., M. Jin, R.S. Lima, J.E. Kobza, and J.A.B. Montevechi. 2019. "The Role of Simulation and Optimization Methods in Supply Chain Risk Management: Performance and Review Standpoints". *Simulation Modelling Practice and Theory* 92:17–44.

Saaty, T.L. 1980. *The Analytic Hierarchy Process*. New York: Mc-Graw-Hill

Taylor, S., A. Anagnostou, T. Kiss, G. Terstyanszky, P. Kacsuk, and N. Fantini. 2014. "A Tutorial on Cloud Computing for Agent-Based Modeling and Simulation with Repast". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 192–206. Piscataway, New Jersey: IEEE.

Weisbrod, G., and S. Fitzroy. 2011. "Traffic Congestion Effects on Supply Chains: Accounting for Behavioral Elements in Planning and Economic Impact Models". In *Supply Chain Management – New Perspectives*, edited by S. Renko, 337–354. London: InTech.

AUTHOR BIOGRAPHIES

MOHAMMED FARHAN is a Simulation Data Scientist at Amazon's Global Engineering Services Support team. He received his Ph.D. in Industrial Engineering from The University of Texas at Arlington. His research interests include Hybrid simulation, human behavior modeling, reinforcement learning and supply chain management. His email address is frhann@amazon.com.

PASCALIN NGOKO is a Simulation Data Scientist at Amazon's Global Engineering Services Support team. He received his Master of Applied Science (MASc) in Industrial Engineering from Polytechnique Montréal Canada in 2019. His research interests are discrete-event and agent-based simulation, optimization, and machine learning. His email address is npascaln@amazon.com.

FAROUQ HALAWA is a Simulation Scientist at Amazon's Global Engineering Services Support team. He received his Ph.D. in Systems Sciences and Industrial Engineering from Binghamton University in 2020. His research interests are in simulation modeling, optimization, and machine learning. His email address is halawafh@amazon.com.

RAASHID MOHAMMED is a Senior Simulation Data Science Manager for Amazon's Global Engineering Services Support team. He obtained his Master's in Industrial and Systems Engineering from Northern Illinois University in 2010. He has over 12 years of industry experience with 4+ years in Caterpillar as an Industrial Research Engineer and 8 years at Amazon in Data Science and Managerial roles. His research interests are simulation optimization, robotics, machine learning and IIOT hardware development. His email address is raashidm@amazon.com.