Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

DISCRETE EVENT SIMULATION USING DISTRIBUTIONAL RANDOM FORESTS TO MODEL EVENT OUTCOMES

Sean Reed
Magnus Löfstrand

Örebro University
School of Science and Technology
Örebro, 70182, SWEDEN

ABSTRACT

In discrete event simulation (DES), the events are random (aleatory) and typically represented by a probability distribution that fits the real phenomena that is studied. The true distributions of event outcomes, which may be multivariate, are often dependent on the values of covariates and this relationship may be complex. Due to difficulties in representing the influence of covariates within DES models, often only the averaged distribution or expected value of the conditional distribution is used. However, this can reduce modelling accuracy and prevent the model from being used to study the influence of covariates. Distributional random forests (DRF) are a machine learning technique for predicting the multivariate conditional distribution of an outcome from the values of covariates using an ensemble of decision trees. In this paper, the benefits of utilizing DRF in DES are explored through comparison with alternative approaches in a model of a powder coating industrial process.

1 INTRODUCTION

Discrete event simulation (DES) is an operational research method for modelling stochastic, dynamic systems where the system evolves through a sequence of events and each event occurs at a precise instant in (simulated) time and results in a change to the system state. It is used for modelling queuing systems, where the system is represented as entities flowing from one activity to another (often after a time delay), and the activities are separated by queues (Robinson 2014). For example, in the DES model of a manufacturing system, the entities might be parts and workers, the activities might be the processing steps that the parts undergo and the tasks that the workers perform, and the queue might be the buffers where parts awaiting processing are stored. In DES, the delays between events and changes in system state resulting from events are random (aleatory), usually represented by probability distributions that fit the real phenomena of the modelled system. For example, the delay between orders arriving might vary according to an exponential distribution with a certain rate parameter, whilst the accuracy of a hole drilled in a part might vary according to a normal distribution with certain mean and standard deviation parameters. In practice, situations are often encountered where these distributions are not constant but depend on the values of certain covariates. For example, the rate at which orders arrive might depend on the time of day and season, whilst the accuracy of a drilled hole might depend on the time since the drilling machine was last calibrated, the experience of the operator, and the rate at which parts are currently arriving for drilling. If the influence of covariates are ignored within a DES model such that the unconditional distribution for the outcome of the event is sampled from, it could result in an inaccurate model, invalid conclusions, and prevent any analysis of the influence of covariates on the behavior of the modelled system.

Let $\boldsymbol{Y} = (Y_1, Y_2, \dots, Y_d) \in \mathbb{R}^d$ be a multivariate random variable representing the outcomes of an event and $\boldsymbol{X} = \left(X_1, X_2, \dots, X_p\right) \in \mathbb{R}^p$ be a set of covariates on which the joint distribution of $\mathbf{Y}$ depends (note:

here and throughout the paper, bold font is used to signify a vector value). The conditional distribution for the event outcome for a certain covariate value vector **x** is then denoted $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}) = \mathbb{P}\left(\boldsymbol{Y}|X_1=x_1, X_2=x_2, \ldots, X_p=x_p\right)$. The relationship between the conditional distribution of random variable **Y** and covariates **X** may be complex and not understood. However, it may be possible to make a set of observations $\{(x_i, y_i)\}_{i=1}^{n}$, where x_i and y_i are the covariate value vector and outcome value vector, respectively, for the ith of n observations. In the manufacturing scenario, for example, the accuracy of drilled holes in parts could be measured and recorded along with the time since the drilling machine was last calibrated, the operator experience, and the current arrival rate of parts. These observations could potentially be used to inform the DES model, so that event outcomes are sampled from more accurate distributions that are generated based on the values of covariates in the model at the time an event occurs. In this paper, an approach is proposed that uses a machine learning method called distributional random forests (DRF) (Michel and Cevid 2021) within a DES, enabling the simulation to utilize covariate values from the model state to sample event outcomes from more accurate distributions. The DRF is trained on example covariate and outcome vector pairs to learn the relationship between them. Alternative methods for modelling the influence of covariates on outcome distributions, both traditional statistical approaches like proportional hazards (Leemis 1990) and machine learning methods like mixture density networks (Reed et al. 2021). However, DRF offers a unique approach that is not based on maximizing the log-likelihood of the training data and has several advantages over the alternatives (see Section 3.1 for more details). The main contribution of this paper is to demonstrate how DRF can be integrated in a DES model and evaluate its advantages in this application. The approach is demonstrated for a DES model of a simple powder coating industrial process and the modelling accuracy compared to two alternative approaches.

The remainder of this paper is organized as follows: Section 2 gives some background to the proposed approach and related works, Section 3 describes the proposed approach for combining DES with DRF, Section 4 describes the case study used to evaluate the proposed approach, Section 5 gives the results of the evaluation, and Section 6 gives some conclusions.

2 BACKGROUND AND RELATED WORK

2.1 Decision Trees and Random Forests

Decision tree learning is one of the most popular types of supervised machine learning algorithm used for predicting an output variable value (target) of an item from input variable values (features) of that item (Wu et al. 2008). After training on example data (i.e., a set of input and output value pairs), a decision tree structure is formed known as a classification tree, if the predicted outcome is from a discrete set, or a regression tree, if the predicted outcome is a real number. Each internal (non-leaf) node of a decision tree is labelled with an input variable and has (usually two) output branches that partition the possible values of the input variable, therefore representing a decision rule. Each leaf node is associated with a value for the output variable. A prediction for the value of the output value of an item corresponds to the value associated with the leaf node that is reached by following a path from the root node of the decision tree along the branches that correspond to the values of its input variables. During the training procedure, the splits in the tree and decision rules are formed to minimize the predictive error on the training data set according to a specified loss function e.g., the least sum of squared error for a regression tree. This is done by ensuring branches partition the data into groups with similar expected values. Decision trees are easy to interpret, can handle both real and discrete valued variables, perform well on large datasets, and give high predictive performance for relatively small computational effort (Rokach and Maimon 2014). However, they are also prone to overfitting, such that predictive performance is high for the examples from the training set but poor for new items. Random forests (Ho 1995) were introduced to correct for the problem of overfitting by decision trees. They are a widely used approach, where a set of decision trees are constructed during training and the predicted output value for an item is the most common output value, for classification, or the average output value, for regression, from that set (Breiman 2001). Each decision tree from the random forest is

trained on a bootstrapped subset of the training data and utilizes a subset of the input variables in its decision rules, reducing overfitting through a "wisdom of the crowds" approach (Mannes et al. 2014). Decision trees and random forests can be used with DES to predict the expected outcome of events during simulation based on covariate values, i.e., $\mathbb{E}[\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}]$ where $\boldsymbol{Y}$ is a random variable representing the event outcome, $\boldsymbol{X}$ is a set of covariates on which the conditional distribution of $\boldsymbol{Y}$ depends, and $\boldsymbol{x}$ is the values of covariates according to the simulation model state when the event outcome is generated.

2.2 Applications of Machine Learning Techniques to Predict Event Outcomes in DES

Traditional approaches to modelling the conditional distributions of event outcomes based on covariate values, such as proportional hazards and accelerated life models (Leemis 1990), require restrictive assumptions such as covariates having a multiplicative effect on hazard rate. Therefore, researchers have investigated how supervised machine learning techniques that are able to learn these relationships from sets of training examples can be utilised within DES models to model events more accurately.

Bergmann et al. (Bergmann et al. 2017) used decision trees to determine dispatching rules for jobs in a DES model of a production system based on properties of the job, such as due date and processing time. Artificial neural networks (ANN), another type of supervised machine learning algorithm, have also been used to predict expected event outcomes in DES. This includes job scheduling decisions and dispatching rules in DES models of manufacturing systems (Bergmann et al. 2016; Bergmann et al. 2014), the treatment time of patients in a DES model of a dental clinic (Chang and Chang 2018), and the acceptance or rejection of loan applications in a DES model of a bank credit approval process (De la Fuente et al. 2018). A limitation of these approaches is that the predictions provided by the decision trees or ANNs are the expected outcomes based on the values of covariates, rather than its conditional distribution. Therefore, these approaches do not allow the variation in possible outcomes for a given covariate vector to be simulated and modelling this is a major reason for choosing the DES modelling approach over alternatives (Ross 2012).

Reed et al. (Reed et al. 2021) utilized a particular type of ANN, known as mixture density networks (Bishop 1994), to predict the conditional distribution of event outcomes within a DES model of load-haul-dump vehicle operations at a mining site. Mixture density networks are able to learn complex, non-linear relations between the conditional distribution of a random variable and the values of covariates, output arbitrary distributions that are not constrained to a particular form, train efficiency on large data sets, and update when new data becomes available. They output the conditional distribution in the form of a weighted mixture of normally distributed components, where the weights, means, and standard deviations are determined by the ANN. The ANN learns to predict mixture distributions from covariate vectors that maximize the log-likelihood of the training data. A downside of ANNs is that achieving good performance usually requires lots of training data and optimized choices of hyperparameters, such as the learning rate and number of layers (Chollet 2017). This motivates the need to explore other machine learning techniques to predict conditional distributions in DES, especially since studies have shown the performance of different learning algorithms can vary significantly across different problems and datasets (Caruana and Niculescu-Mizil 2006).

3 PROPOSED APPROACH

This paper proposes an approach for enhancing the ability of a DES to model a real system by using distributional random forests (DRF), trained on data on events from the real system, to predict the conditional distribution of event outcomes during simulation.

3.1 Distributional Random Forests

DRF were introduced by Ćevid et al. (Ćevid et al. 2020) as a novel random forest (Ho 1995) construction for predicting the multivariate conditional distribution $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x})$. Similar to the random forest algorithm, the branches of the decision trees perform splits based on the values of input variable values. However, whilst the decisions trees from random forests split the data into groups with similar expected values, those from distributional random forests split the data into groups with similar distributions. It constructs a forest from decision trees based on sequential multivariate two-sample test statistics. Any test statistic that can detect a wide variety of distributional changes can be used, with the Maximal Mean Discrepancy statistic (Gretton et al. 2007) proposed as the default choice by Ćevid et al. (Ćevid et al. 2020). At each step of the tree construction process, the data is split into two groups based on a covariate X_j so that the distribution of the output $\boldsymbol{Y}$ for the subset of data for which $X_j \leq l$, for some value l, differs most in comparison to the distribution of $\boldsymbol{Y}$ for the subset for which $X_j > l$, according to the chosen statistical test. This results in a forest comprised from decision trees that each partition the data such that the distribution of the mutlivariate output $\boldsymbol{Y}$ in the leaves is as homegeneous as possible. As with standard random forests, the individual trees are trained on randomly chosen bootstrapped samples from the training data and subsets of the input variables from $\boldsymbol{X}$. For a given test point $\mathbf{x}$, a weighting can be induced on each example from the training data $w_{\boldsymbol{x}}(x_i)$ based on the relative frequency with which that example is encountered in the leaf nodes from a traversal of the DRF with the test point. Thus, the relevance of each training data point x_i to a given test point $\mathbf{x}$ can be quantified and used to estimate the conditional distribution $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x})$ as the empircal distribution determined by these weights. For full details, see Ćevid et al. (Ćevid et al. 2020).

A key advantage of DRF is that it does not depend on a particular estimation target, such as the expected value given by standard random forests. Additionally, it is completely non-parametric, does not require additional user input such as the log-likelihoods, can be used for complicated targets for which there is no obvious forest construction, and only requires a single forest fit for producing estimates of many different targets that are additionally compatible with each other. This contrasts with the mixture density network approach that was previously used to predict conditional distributions in a DES by Reed et al. (Reed et al. 2021), which assumed a parametric form and requires specification of the log-likelihood function.

3.2 Integration of DRF with DES

The overall approach for integrating DRF within a DES model is described by the flowchart shown in Figure 1. For every event in the DES model for which the conditional distribution will be predicted by a DRF, the covariate variables that the conditional distribution depends upon must be identified. The DES model should be designed such that the values of these covariates can be obtained from the model state during simulation. A set of training data of pairs comprising of the multivariate event outcome and the vector of covariate values (e.g., obtained from the real system that will be modelled or a separate, perhaps physics based, simulation model) are then used to train a DRF to predict the conditional distribution $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x})$ for any covariate vector $\boldsymbol{x}$, including those not seen previously in the training data. This step is performed offline, prior to running the simulation. Each time the event occurs during a simulation run, the outcome is sampled using the following procedure:

1. Obtain the values of the covariates from the model state and form them into a vector, where the covariate variable indices match those used in the training data.
2. Obtain weights of the training example from trained DRF that correspond to the covariate vector.
3. Randomly sample an example from the training data, where the probability of selecting each is proportional to the weight assigned to it by the DRF in the previous step.
4. The outcome of the event is then the outcome from the sampled training example.
5. Update the model state according to this event outcome and continue the simulation.

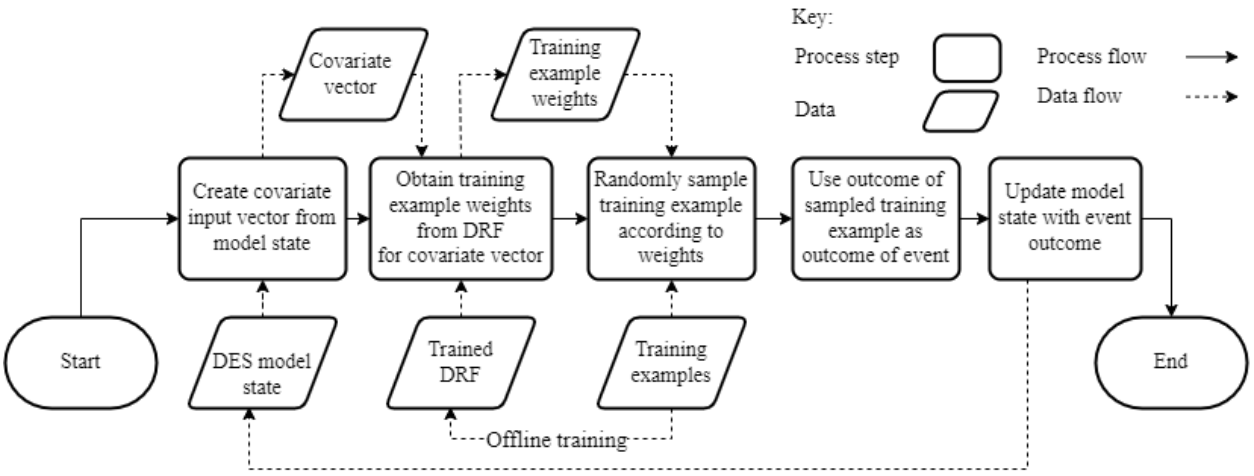

Figure 1: Flowchart of the steps involved in sampling the outcome of an event within a DES when using a DRF to predict the conditional distribution.

4 CASE STUDY

To evaluate the potential benefits of the proposed approach, it was compared to alternative approaches to modelling event outcomes in a DES model of an industrial process. The test case and methodology for this comparison is described in this section.

4.1 Test Case: Powder Coating Process

To evaluate the potential benefits of using DRF to model outcomes within a DES model, a simple industrial powder coating process, shown in Figure 2, is used as the test case that will be modelled. At the start of each day, a set of parts to be powder coated that day are placed into the "Parts In" buffer of the process, which is a first-in-first-out (FIFO) queue of infinite capacity. Each part has a specification that can be summarized in terms of two parameters: the material of the part, given as a value from the set {Steel, Aluminum, Brass, Copper}, and the size of the part, given as a real-valued number between 1 (smallest) and 5 (largest).

Each part is first cleaned at the "Cleaning Station" by a worker and is then placed in the "PC" buffer, which is a FIFO queue of capacity 2. If the "Parts Buffer" is full, then the part waits at the "Cleaning Station" until space becomes available. The processing time for a part at the "Cleaning Station" follows a uniform distribution with a minimum time of 4 minutes and a maximum time of 6 minutes. The cleanliness of a part after cleaning is graded as a real-valued number and follows a triangular distribution with a minimum value of 1 (least clean), modal value of 4, and maximum value of 5 (perfectly clean). Note that both the processing time and cleanliness grade distributions are independent of the part specification.

A worker removes a cleaned part from the "PC" buffer and performs powder coating at the "Powder Coating Station" and inspects the quality of the powder coating application at the "Inspection Station". The processing time, in minutes, for this sub-process (i.e., powder coating and inspection) follows a normal distribution with mean given by Equation (1) and standard deviation given by Equation (2), where a and b are constants that depend on the material from which the part is made (see Table 1) and s is the size parameter value of the part.

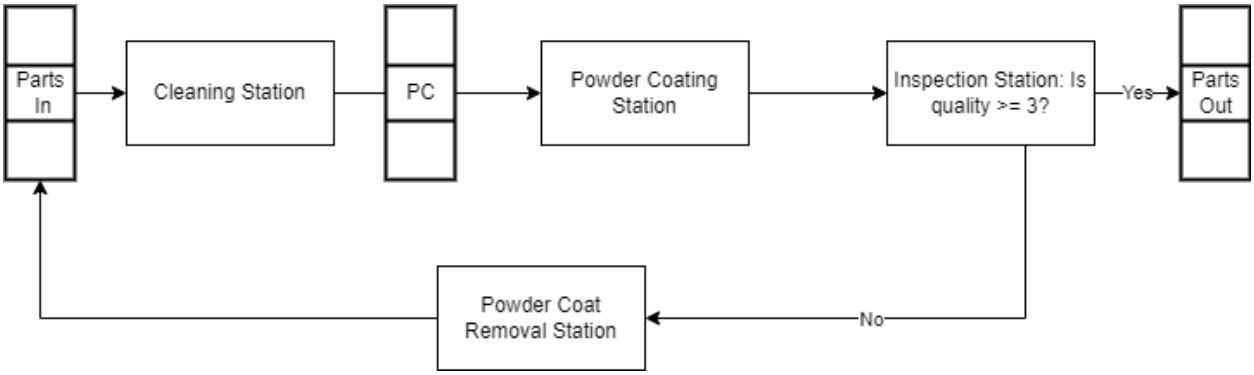

Figure 2: The Powder coating process.

$$a + 1.5s + 4.5 \tag{1}$$

$$\sqrt{(b^2 + (2.5 + 0.4s)^2)} \tag{2}$$

The quality of the powder coating application is graded by the worker during inspection as a real-valued number between 1 (worst quality) and 5 (best quality). The quality value for the powder coating of a part follows a normal distribution with mean given by $max(1, 5 - 0.2(5 - c) - 1.4d)$ and standard deviation of 0.8, where c is the cleanliness value of the part and d is the deviation of the processing time given by $|\mu - t|/\sigma$, where μ and σ are the mean and standard deviation of the processing time for the part based on its specification (given by Equations (1) and (2), respectively) and t is the actual processing time for the part. Poor cleanliness of a part results in poorer quality of powder coating applications because it lessens the ability of the coating to adhere to the part's surface and can cause visual defects such as specks, bumps, or even complete delamination (Liberto 1994). The quality of powder coating application also tends to deteriorate with processing times that deviate below or above the average as this is correlated with under and over curing, respectively, which can lead to brittleness, reduced resistance to chemicals or UV light, and visual defects such as discoloration (Liberto 1994). This also means that the distribution of the quality outcome is not only dependent on the covariates, but also on the processing time outcome.

Table 1: Constants in equations for mean and standard deviation of total processing time of a part at the "Powder Coating Station" and "Inspection Station" steps of the powder coating process for each part material type.

Part Material	a	b
Steel	0	0
Aluminum	0	0
Brass	3	1.2
Copper	3	1.2

If the quality of the powder coating applied to a part is assessed to be less than 3, then that part has the powder coating removed by the worker at the "Powder Coat Removal Station", before being placed at the back of the queue within the "Parts In" buffer at the start of the process. The processing time, in minutes, for powder coating removal follows a uniform distribution with minimum value of 3 and maximum value of 5, independent of the specification of the part. Otherwise, the part passes the inspection and is placed in the "Parts Out" buffer which stores the parts that have completed the process successfully. There is one worker that performs the cleaning step, such that a maximum of one part can be at the "Cleaning Station" at any time, and two workers that perform the other process steps, such that a maximum of two parts can be undergoing this sub-process at any time (each of these workers stays with a part from removal at the "PC" buffer until placing it in the "Parts Out" or "Parts In" buffers).

4.2 Discrete Event Simulation Models

A DES model of the test case process was constructed that was a perfect representation of the true process as described above. The purpose of this first model is to provide benchmark "ground truth" results against which the accuracy of alternative models could be validated and compared. Three alternative DES models were then constructed, which differed only in how the "Powder Coating Station" was modelled, whilst the modelling of all other parts of the process remained perfect. To construct these alternative models, first observations for 1000 parts being powder coating were generated from the true model of the process step, where the input values of material, size, and cleanliness were recorded for each part along with the output values of processing time and assessed quality of powder coating. The inputs for the 1000 parts were generated randomly by sampling each input parameter value from the uniform distribution over the range of possible values. This represents a common scenario when constructing DES models of a real system or process, where the real phenomena in the system or process being modelled is unknown but data on its behavior either exists or can be collected from which a model can be formed. Each of the three models used this generated training data in a different way, as described below :

1. Bootstrapping: Sampling with replacement from the full set of training data. During simulation, when a new part is processed at the "Powder Coating Station", a sample from the training data is sampled at random (uniform distribution) and its outcome used as the event outcome.
2. Random forest: A regression random forest was trained on the set of training data. During simulation, when a new part is processed at the "Powder Coating Station", the expected outcome of the event predicted by the random forest from the covariate vector is used as the event outcome. The Ranger package (Wright and Ziegler 2017) for the R programming language was used as the implementation of the random forest.
3. Distributional random forest: A DRF was trained on the set of training data. The DRF package (Michel and Cevid 2021) for the R programming language was used as the implementation of DRF. During simulation, when a new part is processed at the "Powder Coating Station", the DRF was used to predict the conditional distribution of the outcome using the approach described in Section 3.

The default values within the R packages were used for the training and construction of the random forest and DRF. These DES models were constructed using the Simmer package (Ucar et al. 2019) for the R programming language. This package is a process-orientated library for DES that provides facilities for specifying a model in the form of trajectories (a chained set of activities that form a process) through which entities of a certain type (e.g., parts) flow, as well as automatic monitoring capabilities. The DES models were configured such that a set of parts to be powder coated were inserted into the "Parts In" buffer of the process at the start of a simulation (i.e., simulation clock time of 0) and the process then simulated until all parts had been successfully powder coated.

4.3 Performance Metric, Scenarios, and Model Accuracy Evaluation

The accuracy of the three alternative DES models was evaluated by comparing the values of a key performance indicator (KPI) output metric against those from the benchmark model resulting from the simulation of two different scenarios. The chosen KPI was the total time to successfully powder coat all 100 parts. For each scenario, the specifications of 100 parts to be powder coated were generated. For the first scenario, the size and material properties were generated from the uniform distribution (i.e., the same distribution as the training data used to train the DRF and random forests). For the second scenario, the material properties were generated from a uniform distribution and the size properties were generated from a triangular distribution with minimum of 1, maximum of 5, and mode of 2. This specification of the parts from this second set therefore differs from the training data by having a smaller size on average. For each DES model, the powder coating of the parts from each scenario was simulated 100 times, where each repetition used a different random seed value, producing 100 values for the KPI of the total time to powder coat all parts. The distribution of this KPI for each alternative model was then compared to that from the benchmark model, with greater similarity indicating higher modelling accuracy. This comparison was performed visually through the use of box plots and quantile-quantile (Q-Q) plots. A box plot summarizes the minimum, first quartile, median, third quartile, and maximum of the distribution, whereas a Q-Q plot is a scatterplot of the quantiles from one data set (in this case, the "ground truth" model) against another (in this case, the three alternative modelling approaches).

5 RESULTS

The box plots and Q-Q plots are given in this section comparing the distributions of the total time to powder coat 100 parts output by the three alternative DES models, that differed only in the approach used to model the "Powder Coating Station" step of the manufacturing process, to the "ground truth" distribution output by the benchmark model. The box plot and Q-Q plots comparing the KPI distribution in Scenario 1 output by the benchmark and alternative models are shown in Figure 3 and Figure 4, respectively. The box plot and Q-Q plots comparing the distributions of the KPI in Scenario 2 output by the benchmark and alternative models are shown in Figure 5 and Figure 6, respectively. The solid black line shown in each Q-Q plot is a reference line representing perfect accuracy, i.e., where all quantiles have equal value.

6 DISCUSSION AND CONCLUSIONS

An approach for integrating DRF within DES models to predict the conditional distribution of multivariate event outcomes was proposed and tested in two scenarios for a model of a powder coating process. In each scenario, a DES model utilizing DRF was compared to approaches that instead used bootstrapped samples and random forests to determine outcomes of a key event in the modelled process. This compared the accuracy with which the DES models predicted the distribution of a key performance indicator (KPI) for the process, the total time to powder coat 100 parts. The box plot and QQ-plot for Scenario 1, show that the DRF approach is able to predict the distribution of the KPI with better accuracy than the other approaches.

The box plot shows that the median and other key quartiles, with the exception of overestimating the third and maximum, closely match those from the benchmark. In the QQ-plot, it is also evident that the quantiles follow the benchmark well as the deviations from the reference line are small overall. The bootstrap approach also does reasonably well. This could be expected since the distribution of the covariate values (i.e., the size and material for the parts) in this scenario is the same as in the training data that the bootstrapped samples are taken from. However, it cannot match the performance of the DRF approach which adjust the conditional distributions of the event outcome (i.e., the processing time and quality for the powder coating step of the process) for each individual part in the generated set of 100 parts. The random forest approach, which predicts and utilizes the expected outcomes for the event, performs badly. It severely underestimates the mean time to complete the processing of all 100 parts and also gives much lower variance for that KPI.

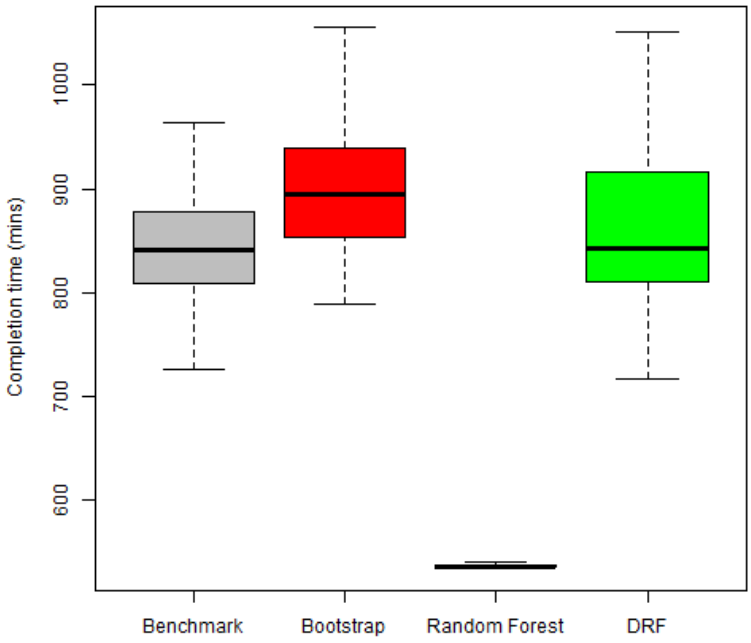

Figure 3: Box plot comparing the distribution of the total time for 100 parts to complete the powder coating process in Scenario 1 and the three alternative modelling approaches.

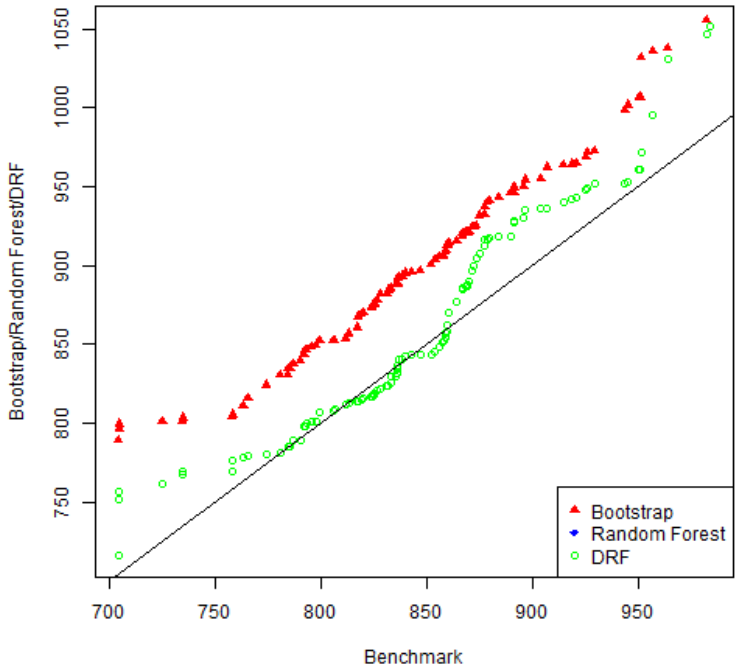

Figure 4: Q-Q plot comparing the distribution of the total time for 100 parts to complete the powder coating process in Scenario 1 and the three alternate modelling approaches (note: plot points for random forest approach are outside axis range shown).

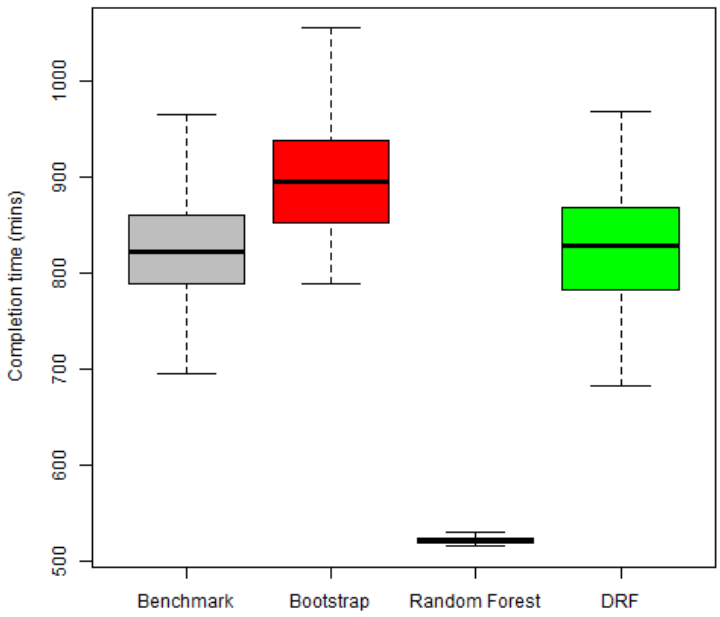

Figure 5: Box plot comparing the distribution of the total time for 100 parts to complete the powder coating process in Scenario 2 and the three alternate modelling approaches.

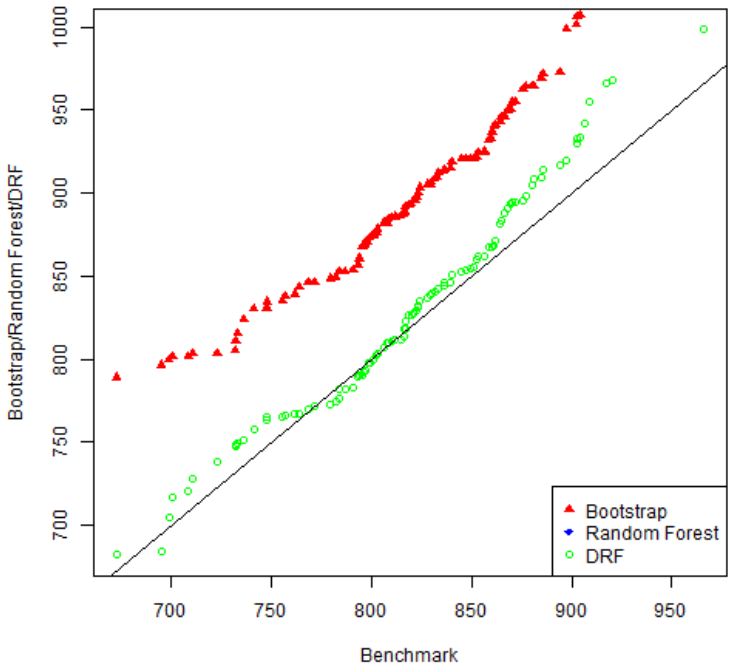

Figure 6: Q-Q plot comparing the distribution of the total time for 100 parts to complete the powder coating process in Scenario 1 and the three alternate modelling approaches (note: plot points for random forest approach are outside axis range shown).

The main reason for this is that the expected quality level of powder coating for parts of any specification is such that it passes the inspection, therefore this model never simulates parts having to have the powder coating removed and re-applied due to poorer than expected quality. In Scenario 2, the plots show that the DRF based approach is able to achieve remarkably accurate results that are very close to the true distribution from the benchmark model, performing at least as well as it did in Scenario 1 . In contrast, the bootstrap approach has lower accuracy than it did in Scenario 1. This is to be expected, since it uses the event outcome distribution from the training data and, unlike the DRF approach, is unable to adjust for the values of the covariates, which have a different distribution in this scenario. The random forest approach again performs poorly, for the same reasons as discussed for Scenario 1.

These results appear to indicate that using DRF to predict the conditional distribution of event outcomes can increase the accuracy of DES models. This includes situations where event outcomes have a complex relationship with covariate values and are multivariate with dependencies between the individual outcome variables. The results also demonstrated a weakness in machine learning approaches which have been used in the past with DES, such as random forests, that predict only the expected value of event outcomes from covariate values. In DES models of many systems, such as the example process modelled in this paper, it is important that the variation in event outcomes is modelled to obtain accurate and useful predictions. Whilst promising, the statistical significance of these initial results require verification through future work that extends to a wider range of scenarios. The proposed DRF approach has potential Industry 4.0 (Lu 2017) and digital twin (Boschert and Rosen 2016) applications, where data collected from a production process can inform DES models used to predict and optimize its performance. By learning from past event outcomes and covariate values, such DES models could improve in accuracy over time and adapt to the changing circumstances of the real systems they represent. An area for further work is to compare the use of DRF to predict the conditional distribution of event outcomes to competing approaches, such as the ANN based approach described by Reed et al. (Reed et al. 2021), for models of a variety of different systems and scenarios.

REFERENCES

Bergmann, S., S. Stelzer, and S. Strassburger. 2014. "On the Use of Artificial Neural Networks in Simulation-Based Manufacturing Control". *Journal of Simulation* 8(1):76–90.

Bergmann, S., N. Feldkamp, and S. Strassburger. 2016. "Approximation of Dispatching Rules for Manufacturing Simulation Using Data Mining Methods". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. Roeder, P. Frazier, R. Szechtman, E. Zhou, T. Huschka and S. Chick, 2329–40. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bergmann, S., N. Feldkamp, and S. Strassburger. 2017. "Emulation of Control Strategies through Machine Learning in Manufacturing Simulations". *Journal of Simulation* 11(1):38–50.

Bishop, C. M. 1994. "Mixture Density Networks". Aston University, Birmingham, UK.

Boschert, S. and R. Rosen. 2016. "Digital Twin - The Simulation Aspect". In *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, edited by P. Hehenberger and D. Bradley. 59–74. Zurich: Springer.

Breiman, L. 2001. 'Random Forests'. *Machine Learning* 45(1):5–32.

Caruana, R. and A. Niculescu-Mizil. 2006. "An Empirical Comparison of Supervised Learning Algorithms". In *ACM International Conference Proceeding Series* 148, June 25th-29th, New York, 161–68.

Ćevid, D., L. Michel, J. Näf, N. Meinshausen, and P. Bühlmann. 2020. "Distributional Random Forests: Heterogeneity Adjustment and Multivariate Distributional Regression". arXiv:2005.14458 [stat.ML].

Chang, W., and Y. Chang. 2018. "Design of a Patient-Centered Appointment Scheduling with Artificial Neural Network and Discrete Event Simulation". *Journal of Service Science and Management* 11(1):71–82.

Chollet, F. 2017. *Deep Learning with Python*. 1st ed. New York: Manning Publications.

Gretton, A., K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. 2007. "A Kernel Method for the Two-Sample-Problem". *Advances in Neural Information Processing Systems* 19(1): 513–20.

De la Fuente, R., R. L. Smith III, and I. Erazo. 2018. "Enabling Intelligent Processes in Simulation Utilizing the TensorFlow Deep Learning Resources". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain and B. Johansson, 1108–19. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Leemis, L. 1990. "Variate Generation for Accelerated Life and Proportional Hazards Models with Time Dependent Covariates". *Statistics & Probability Letters* 10(4):335–39.
Liberto, N. 1994. *Powder Coating. The Complete Finisher's Handbook*. 3rd ed. Independence, Missouri: Powder Coating Institute.
Lu, Y. 2017. "Industry 4.0: A Survey on Technologies, Applications and Open Research Issues". *Journal of Industrial Information Integration* 6(1):1–10.
Mannes, A. E., J. B. Soll, and R. P. Larrick. 2014. "The Wisdom of Select Crowds." *Journal of Personality and Social Psychology* 107(2):276–99.
Michel, L., and D. Cevid. 2021. "Drf: Distributional Random Forests". https://cran.r-project.org/package=drf, accessed 15th June.
Reed, S., M. Löfstrand, and J. Andrews. 2021. "Modelling Stochastic Behaviour in Simulation Digital Twins through Neural Nets". *Journal of Simulation* 16(5): 512–525.
Robinson, S. 2014. *Simulation: The Practice of Model Development and Use*. 2nd ed. London: Red Globe Press.
Rokach, L., and O. Maimon. 2014. *Data Mining with Decision Trees: Theory and Applications*. 2nd ed. Singapore: World Scientific.
Ross, S. 2012. *Simulation*. 5th ed. San Diego: Academic Press.
Ho, T. K. 1995. "Random Decision Forests". In *Proceedings of 3rd International Conference on Document Analysis and Recognition,* 14th – 16th August, Montreal, Canada, 278–82.
Ucar, I., B. Smeets, and A. Azcorra. 2019. "Simmer: Discrete-Event Simulation for R". *Journal of Statistical Software* 90(2): 1 - 30.
Wright, M. N., and A. Ziegler. 2017. "Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". *Journal of Statistical Software* 77(1): 1 – 17.
Wu, X., V. Kumar, Q. J. Ross, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. 2008. "Top 10 Algorithms in Data Mining". *Knowledge and Information Systems* 14(1): 1 – 37.

AUTHOR BIOGRAPHIES

SEAN REED is a Senior Lecturer of mechanical engineering in the School of Science and Technology at Örebro University. He earned his Ph.D. in the Department of Aeronautical and Automotive Engineering at Loughborough University, UK. His research interests include the application of discrete event simulation to predictive digital twins, maintenance optimisation, and reliability modelling of complex systems. His e-mail address is sean.reed@oru.se.

MAGNUS LÖFSTRAND is a Professor of mechanical engineering in the School of Science and Technology at Örebro University. He earned his Ph.D in the Computer Aided Design deparment at Luleå University of Technology. His research is mostly focused on approaches to support availability, costs, and new business models in the process and production industry through techniques such as data analytics and simulation. His email address is magnus.lofstrand@oru.se.