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 AND MACHINE LEARNING FOR PROTOTYPE COMPOSITES MANUFACTURE LEAD TIME PREDICTIONS

Jamie Karl Smith
Calum Dickinson

Advanced Manufacturing Research Centre
AMRC, University of Sheffield,
Rotherham, S60 5TZ, UK

ABSTRACT

The article looks to generate synthetic data for machine learning algorithms using discrete-event simulation (DES). The case study used for the DES model was the Composite Centre at the AMRC, where prototype composites products are manufactured. The machine learning algorithm was used to predict the lead times of composite products based on the current state of the system. The machine learning algorithm can calculate the lead times much faster than a simulation model and does not require the expertise of a simulation engineer to execute. Three different types of composites materials and their manufacturing process were initially modelled: dry fiber, prepreg and thermoplastic. The accuracies of three machine learning algorithms were compared. The algorithms chosen were: Artificial Neural Network (ANN), Recurrent Neural Network (RNN) and linear regression. It was found that the RNN provided the most accurate predictions and the linear regression algorithm was the worst performing algorithm.

1 INTRODUCTION

Discrete Event Simulation (DES) is a tried and tested simulation methodology that has been around for decades. The methodology is widely used in manufacturing applications and can provide insights into key performance indicators, such as: throughput, machine utilization, staff utilization and manufacturing lead times. The main advantage of using DES is that the modeler can simulate any number of scenarios without having to interfere with real-world production. DES can be utilized to determine the right quantity of equipment and workers required to meet production volumes while keeping the utilization of the resources high. The problems with using DES typically arise from the accuracy of the data. How the data is collected can have an impact on this accuracy. The model also requires a subject matter expert to build and maintain the running of the simulation model(s) making it difficult for planners to come up with strategies without additional support.

A trend to improve and update how DES models are implemented has become popular in the manufacturing sector due to the emergence of Industry 4.0. Techniques such as digital twinning, where live real-time factory data is linked to the simulation model (Beregi et al. 2018, Eyre et al. 2019, Grube et al. 2019) and the use of artificial intelligence have come to the fore. Simulation is increasingly being used to generate synthetic datasets to feed artificial intelligence models. These simulated datasets can be large and the data labelled (Behl et al. 2020). Behl et al. (2020) focused on the simulation of images to generate models for computer vision. The majority of synthetic data generation has been demonstrated for computer vision, although other fields are starting to adopt this approach (Nikolenko 2021). Research into how artificial intelligence and machine learning (ML) can best service DES is still in its infancy and may yet be specific to certain case studies. Greasley and Edwards (2019) published a PRISMA systematic literature review of how discrete-event simulation has been enhanced with big data analytics. The review found a total of eight papers that look at using machine learning with DES. Three of the eight papers use a combination of Siemens Tecnomatix Plant Simulation for the modelling work and MATLAB as the machine learning tool, with two of these papers using the ANN ML method (Bergmann et al. 2014, Bergmann et al. 2017). Both of these papers look to use ANNs for simulation manufacturing control and look to approximate dispatching rules. Another use case for ANNs and DES has been to use ANNs to determine the current distribution of each event outcome, enabling a more realistic and accurate modelling of stochastic behavior (Reed et al. 2021). Although various avenues for coupling DES and AI have been explored, the main research trend found in the literature has focused on production scheduling/ planning (Priore et al. 2018, Waschneck et al. 2018, Heger et al. 2020, Woo et al. 2020, Krause 2020, Turgut et al. 2020). None of these papers deal with the impact of a prototyping manufacturing line. With prototyping manufacture the planning and scheduling of production runs is more sporadic than mass manufacturing. As such the lead times for a given part to be produced will vary drastically depending on the state of the system.

In this paper, we introduce a method to predict lead times of a composite part using a combination of DES and ML. The purpose of the method is to allow planners to use the predictive ML model to easily calculate lead times without having to set all the boundary conditions within a simulation. One advantage of using simulation to generate synthetic data is the quantity that can be generated in a relatively short period of time, i.e. five years' worth of production data in a matter of hours and minutes. Another advantage is that the data is labelled and structured, simplifying the data preparation and extraction phases (Lopez-Rojas et al. 2012). The impact of the ML algorithm choice to predict lead times is also assessed, with three different algorithms selected. The three chosen algorithms were: linear regression, ANN and a Long Short Term Memory (LSTM) RNN. The impact of the number of hidden neurons on RMSE, for two algorithms in the ANN model was also assessed. Currently, to the best of the author's knowledge, LSTM RNNs have not been used in conjunction with DES models.

Section 2 of this paper introduces the composites manufacturing process that has been utilized in this case study, detailing the three composite material types and their respective process flows. The methodology to build the discrete event simulation model, extract the structured data and then fed it into one of three machine learning algorithms is detailed in Section3. Simulation model validation and the results from the machine learning predictions are presented in Section 4. Conclusions are presented in Section 5.

2 COMPOSITES MANUFACTURING: CASE STUDY DESCRIPTION

The University of Sheffield Advanced Manufacturing Research Centre (AMRC) is a network of research and innovation centers working with advanced manufacturing companies around the globe. The AMRC Composite Centre, which typically manufactures prototype composites parts, was selected as a case study due to the variability in parts that are manufactured. This variability makes production planning difficult. Processing of composites encompasses a wide range of material formats and manufacturing methods. Different processes may have significant differences in level of automation and thus the cycle time and labor resources required. The following three processes and materials have been considered for DES modelling:

- Thermoplastic Automated Fiber Placement (AFP)
- Dry fiber and liquid resin molding in a hydraulic press tool
- Prepreg layup for Double Diaphragm Forming (DDF) using a hydraulic press

2.1 Automated Fiber Placement

The AMRC's AFP cell uses a 6-axis robot with external mandrel axis and various fiber placement heads depending on the material format to be used. The composite laminate is constructed layer by layer as the machine deposits tape material onto a solid mandrel using a compaction roller and heat source. The mandrel can take the final shape of the component, so no separate forming operation is required to create the 3D shape. For this case study, a carbon-fiber tape pre-impregnated with a thermoplastic matrix is used. This head only deposits a single tape at a time and the speed must be reduced where the mold surface has tight radii or double curvature, which results in a cycle time that is highly dependent on component size, thickness, and complexity. A generic composite center shop floor process flow was produced for manufacturing a composite component using the AFP process.

The automated fiber placement process time uses a time multiplier that accounts for the dimensions and complexity of the thermoplastic part. The baseline time of 25 minutes and 5 seconds is multiplied by the factor shown in Table 1.

Table 1: Baseline time multipliers for AFP cell.

Material Attribute	Low	Medium	High
Complexity	0.325	1	2.6
Thickness	10	25	60
Size	1	2	4

2.2 Dry Fiber Liquid Molding Process

The process begins by processing the dry carbon-fiber fabric and later infuses this with the polymer matrix in a liquid molding process. In this case the reinforcement fabric is cut into the required size pieces by a CNC ply cutting machine. These fabric plies are manually laid into a mold tool and a hydraulic press holds the matched mold closed during liquid molding. A Resin Transfer Molding (RTM) machine injects thermosetting resin into the mold tool to infuse the dry reinforcement fabric with the polymer matrix. The polymer cures whilst the tool is still closed to mold the composite component to the desired shape.

2.3 Prepreg Double Diaphragm Forming

The process starts by cutting to size a carbon-fiber fabric pre-impregnated with thermoset polymer matrix, also known as prepreg. The Double Diaphragm Forming (DDF) method is used to form the prepreg to a 3D geometry and cure the polymer matrix. The prepreg is held between two diaphragm films under vacuum and a hydraulic press is used with matched molds to form the geometry. Other methods of forming the geometry can be used but, in this case, the use of a press tool is a variation on DDF also known as Double Diaphragm Compression Molding. The cutting time for the prepreg and dry fiber products depends upon the material size, thickness and complexity. A reference part that is low in size, thickness and complexity has a cutting time of 30 minutes. The time multiplier for more complex and larger parts is presented in Table 2.

Table 2: Baseline time multipliers for cutting dry fiber and prepreg parts.

Material Attribute	Low	Medium	High
Size	1	2	4
Thickness	1	1.12	1.24
Complexity	1	1.33	N/A

3 METHOD

3.1 DES Model Build & Data Capture

The methodology described in this section addresses the process of building the simulation model, the input data requirements and how the simulation data could then be exported into the ML model. DES model building usually follows a set process to go from defining the problem to obtaining results. The first step in the model building is to define the problem we are trying to solve. In this case we are aiming to model the manufacturing process of the AMRC Composite Centre. The second stage of the model build process looks at gathering input data for the simulation. The input data gathered for the DES model can be grouped into various categories, such as: process and time information, resource information and demand information. The demand information splits the product types by percentage with 67% of the orders being prepreg products, 27% dry fiber products and 10% being thermoplastic. Once the order has been generated it is assigned a dimension and complexity. The dimension and complexity of the product to be manufactured has a direct impact on manufacturing time. The prepreg and dry fiber product attributes are grouped into small, medium, and large, for area and thickness and simple or complex for the complexity. The thermoplastic products have the same grouping of attributes with different timings for the dimensions it the complexity is split into small, medium, and high levels of complexity. The prepreg and dry fiber products could also form part of a batch, with a maximum of up to three identical products entering the system at the same time. The batch size like the dimensions and complexity attributes were generated randomly. The random generator determines a number in a uniform distribution. The integer that this number is closest to is then used to assign a specific attribute e.g. if the value =1 then the attribute = small. Using a uniform distribution makes the likelihood of any product attributes or batch size equally likely.

The baseline simulation model was constructed in 3D using Siemens Tecnomatix Plant Simulation software using a CAD layout of the composites center and press building. The 3D model enabled subject matter experts to easily identify the processes and assess the accuracy of the material flow. Presented in Figure 1 is a screenshot of an area in the composites manufacturing center part of the model.

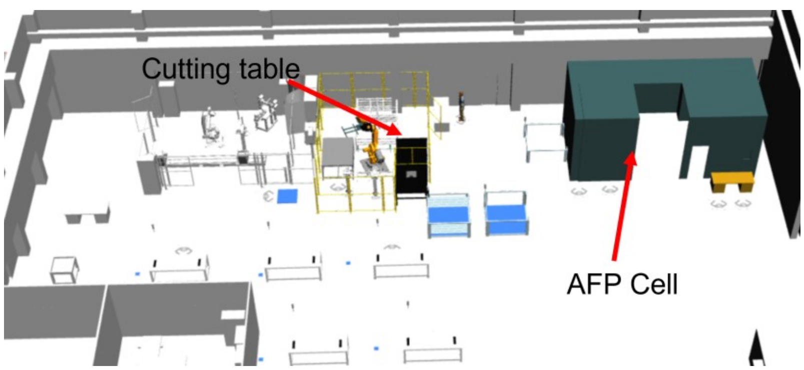

Figure 1: Annotated screenshot of an area of the Plant Simulation Model.

Once the baseline simulation model was built it was validated using two methods. The first validation method was to check the material flow animation in the simulation matched the process. The second validation method randomly generated different product types and to run them through the model with no other products. This was to check that lead time from the model corresponded with the timing values that had been provided. The quantitative results from the model validation are detailed in section 4.1 of this paper.

Several modelling assumptions about the process were made. These assumptions were agreed upon with subject matter experts within the AMRC Composite Centre. A full list of assumptions and statements about how the model operates is stated below:

- A single prepreg and dry fiber kit is cut at a time and no nesting of these kits occurs.
- Orders are prioritized based on when they entered the system. The oldest orders have the highest priority.
- One worker is dedicated to the transport of material to and from the composites center and press building.
- The maximum number of workers available for the model was 9.
- The maximum number of layup tables available was 5.
- The shift pattern was 8 hours/day Monday-Thursday with a 30 minute break and 5 hours on Fridays with no breaks.
- The maximum batch size for dry fiber and prepreg material was 3, thermoplastic was not batched.

The data was captured in the simulation model itself and stored in a table that was then exported to a comma separated values (csv) file once the simulation had stopped running. Because the aim of the AI model was to become a forecasting tool, data could not be captured about a specific product after it had entered the system. For example we could not use data on how many times a prepreg product type had needed freezing before it had been completed. Details about the facility such as work and details about the product, were all recorded as the product entered the system. The full list of input variables includes:

- Product specific process time
- Quantity in the queue for the cutting table
- WIP
- Product start day
- The number of parts waiting for transit to the press building
- Batch size
- The number the product was within that batch
- The minimum cutter waiting time

The product specific process time is derived from the material type, the dimensions, and the complexity of the part. The product start day was valued from 1-5 with Monday equal to 1 and Friday equal to 5. More granular information on the product start time was not required as the minimum lead time, after ramp up, was 23 hours. The input parameters and manufacturing lead times were stored within tables in Plant Simulation and exported to a comma separated values file using the ActiveX interface package, once the simulation run had finished. The csv data file could then be fed into one of three machine learning methods. The full process is displayed in Figure 2.

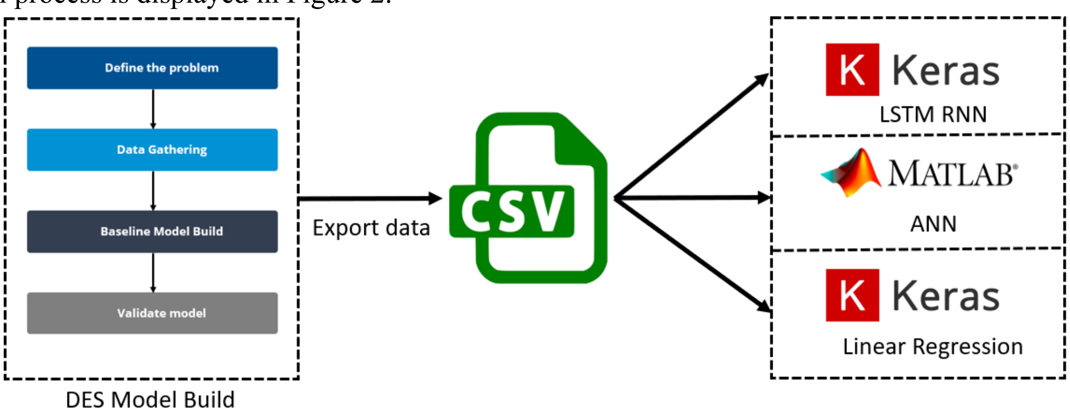

Figure 2: Overall process architecture.

3.2 Machine Learning Methodology

This section describes the various methodologies used when utilizing machine learning to predict manufacturing lead times. The measure of how well each of the algorithms performs was done by analyzing the RMSE.

3.2.1 ANN

Commercially available software was utilized to train and build the ANN. The software of choice was MATLAB due to it containing an easy-to-implement neural network wizard. The easy implementation enabled the assessment of whether good predictive results could be achieved by individuals with little to no machine learning experience. The wizard used in this case study was the neural net fitting wizard. The wizard allowed some flexibility in the design of the network but the model structure was not fully configurable. The number of hidden neurons could be specified by the user as could the data split between training, validation and testing. Three training algorithms were available for selection, these include: Levenberg-Marquardt, Bayesian regularization, and scaled conjugate gradient. The scaled conjugate gradient was not considered for this study as the memory required for the dataset size wouldn't be an issue for the other two algorithms. The number of hidden neurons was not a fixed parameter and was varied to find optimal results. A parameter fixed by the wizard is the number of layers in the model. There is one hidden layer and one output layer between the input and output values. The structure of the model is presented in Figure 3.

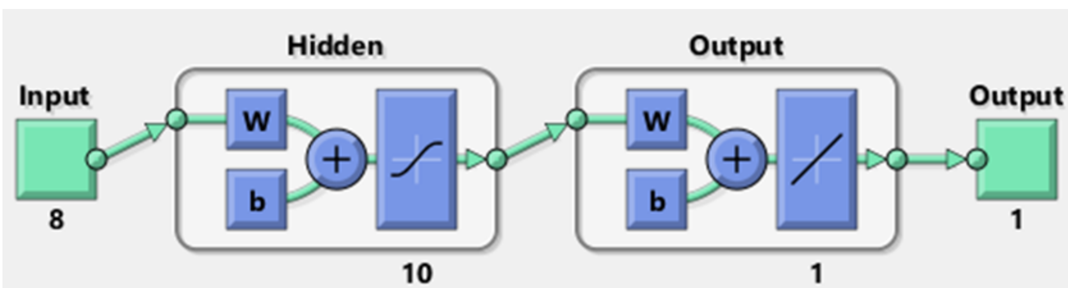

Figure 3: ANN structure for a network with 8 input features, 10 hidden neurons and a single output parameter.

Because the MATLAB neural net fitting wizard had no functions to normalize the data a preprocessing step of the data was conducted. This preprocessing step scaled the input data using a min-max normalization method. The data split for the ANN was 85% training data and 15% testing data.

3.2.2 RNN-LSTM

The RNN model was programmed in Python using open-source software and libraries. The key library used was TensorFlow which has been developed by Google. Keras is a deep learning API that runs on top of TensorFlow (Keras 2021). Once the data has been imported in from the comma-separated variable file it is scaled using a standard scaler from the scikit learn library. The data is then split into training and test data, with the split being 85% training data and 15% testing data. A sequence length is then defined to determine how many of the previous input values are also considered for this given data point. The model is sequential which means that the algorithm runs through several steps in sequence. In total 5 different layers are utilized in this model. The first is a Long Short-Term Memory (LSTM) layer, with units equal to 64. The number of units represents the dimensionality of the output space. For the first LSTM layer, the return sequences option was set to true. The second layer is a dropout layer which randomly sets input values to 0 with the rate at which it does this being an option within the model. Both dropout layers have a rate of 0.2 for the algorithm used in this model. The third layer is a bi-directional LSTM layer. The bi-directional LSTM layer doesn't not return the sequences and only 20 units are used for this LSTM layer. A second dropout layer then follows with the same attributes as the first. Finally, a dense layer is used to calculate the output values. The model is compiled so that the loss to be reduced is the mean squared error (MSE). The optimizer used was 'rmsprop'. The lead time is then inversely scaled so that the results have a meaningful value.

3.2.3 Linear Regression

The linear regression model was programmed in Python, using the TensorFlow library and Keras API. The importing, scaling and splitting of the data is the same as it was for the RNN LSTM model (see section 3.2.2). The linear regression model does not require a sequence length to be defined and the model sequence consist of only a single dense layer. The model is once again compiled to use MSE as the loss and the model optimizer is 'rmsprop'.

4 RESULTS

4.1 Model Validation

Validation of the simulation model was essential before any data could be gathered from the simulation. The validation was to ensure that the process being modelled was realistic and not a hypothetical scenario. The model logic not matching reality is unlikely to cause too much of a difference with what ML algorithm is required to solve the problem as any simulation logic will be captured in the ML model. A realistic simulation presents a stronger case study and demonstrates that we are not simply making a simulation scenario that works with ML but is never seen in manufacturing.

Model validation comprised of two steps, the first being to randomly generate different product types and to run them through the model with no other products. This was to check that lead time from the model corresponded with the timing values that had been provided. The results from comparing the expected lead time and the simulation lead time are presented in Table 3.

Table 3: Simulation Lead Times Compared to Timing Data Lead Times for multiple products

Material Type	Dimensions	Complexity	Expected Lead Time (h)	Simulation Lead Time (h)	Difference (%)
Prepreg	Small Area, Medium Thickness	Simple	6.74	6.84	1.36
Dry Fiber	Small Area, Medium Thickness	Complex	6.08	6.09	0.30
Thermoplastic	Small Area, Medium Thickness	Medium Complexity	49.29	50.30	2.05

The lead times calculated by the simulation are close to those when calculated by using the process timings. The difference is likely to be associated with the time taken to transport the parts. These results show that the model timings and process flow have been set up correctly within the simulation model for each of the individual material types. The simulation uses no probability distributions for the timings and the timings provided by the AMRC Composite Centre also have no variation for a specific part type and as such no uncertainties can be reported for either of the times stated.

The second validation method was qualitative and involved demonstrating the flow of the different product variants through the composites center process. Subject matter experts from the AMRC Composite Centre confirmed that the flow of material followed the real-world manufacturing method.

4.2 ANN Results

The results presented in this section use a standard ANN that uses one set of inputs for one output value. Initially the three material types were ran through the simulation and trained using the Bayesian Regularization algorithm in the MATLAB neural-net fitting wizard. The Bayesian Regularization runs for 1000 epochs, and this number is fixed within the software wizard. The number of hidden neurons for the ANN was set equal to 40. The simulation data was obtained from 35 simulation runs, where the number of workers and layup tables available were changed for each run. The simulation was run for 5 production years. The manufacturing lead time RMSE value was found to be 103 hours. This value is 4.34 times greater than the minimum manufacturing lead time found in the output data. It was therefore apparent that a single machine learning model would not be able to cover all three material types.

The simulation was then set up to focus solely on a single material type, with dry fiber being selected for several reasons, including: no freezing required unlike prepreg parts, relatively low lead times for complex large products compared to thermoplastics, and the smallest error value for the model validation phase. The dry fiber simulation duration was 5 years and only the one observation point was made for when the number of layup tables and number of workers were kept constant at values of 5 and 9 respectively. The collected data was then used to train the ANN. The ANN was trained using the both the Bayesian Regularization and Levenberg-Marquardt algorithm. The number of hidden neurons was varied to determine an optimum RMSE value for each algorithm, with the results presented in Figure 4.

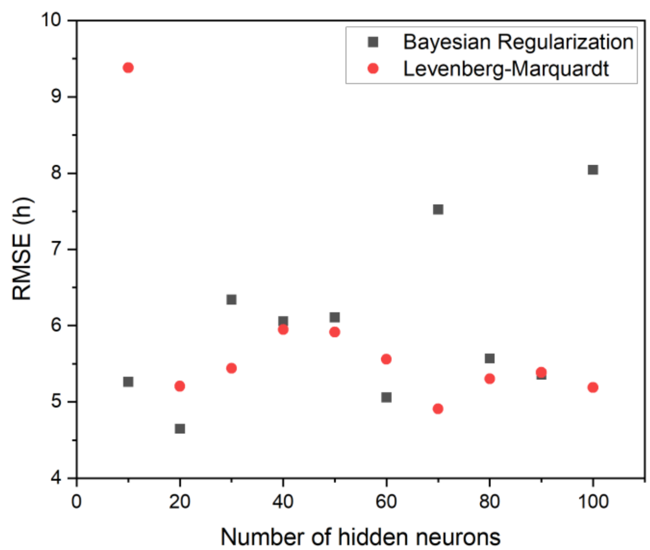

Figure 4: Plot of RMSE versus the number of hidden neurons for the Bayesian Regularization and Levenberg-Marquardt algorithms.

The lowest RMSE value is 4.65 hours which is approximately 22 times smaller than the RMSE when all three material types are running through the simulation and used to train the ANN. The Bayesian Regularization algorithm provides the lowest RMSE value of 4.65 hours with 20 hidden neurons, while the Levenberg-Marquardt best result of 4.91 hours is found when 70 hidden neurons have been used to train the network. The Levenberg-Marquardt has a lower mean RMSE of 5.82 hours compared to 6 hours for Bayesian Regularization. There is no correlation between the number of hidden neurons and the RMSE value. To give a visual indication of how the ANN model performs a plot of actual and predicted lead times, for the Bayesian Regularization algorithm with 20 hidden neurons, is presented in Figure 5. The predicted and actual values were selected randomly from 100 continuous data points from the testing data.

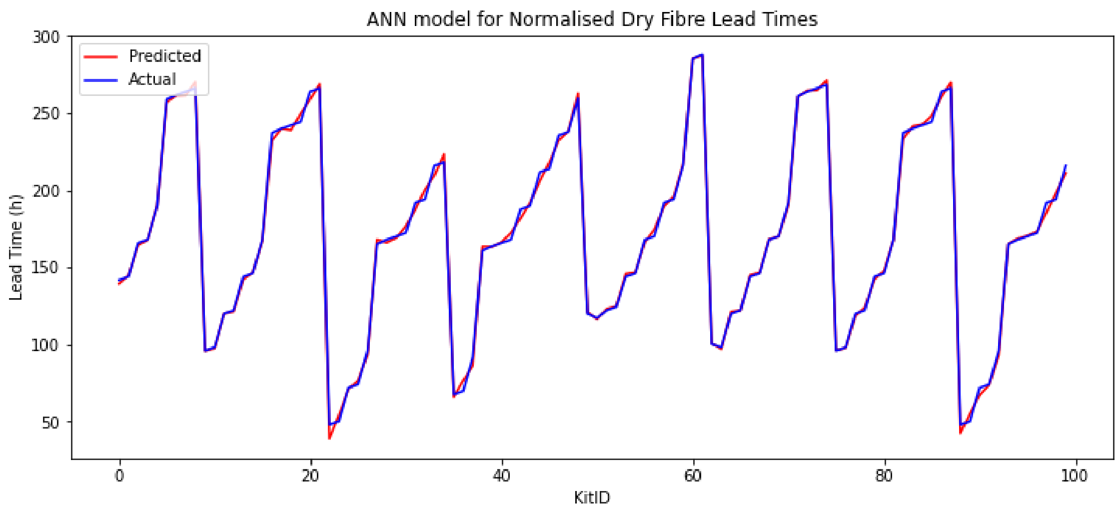

Figure 5: Lead time for dry fiber product types comparing the actual values (blue) and the predicted values (red) for a sample of 100 values.

4.3 RNN Results

The results presented in this section use the LSTM RNN described in section 3.2.2. Initially the number of epochs was set to 200. The RNN takes a sequence of inputs to calculate the output. The sequence length is variable and determined by the user. Three different sequence lengths were compared for the RNN model. The sequence lengths were: 5,10 and 15. One logical constraint for sequence size is that it should not be greater than the maximum WIP quantity in the system, as parts no longer in the system are unlikely to have an impact on new parts being produced. The training and testing loss values for a sequence length of 10 and 200 epochs is presented in Figure 6.

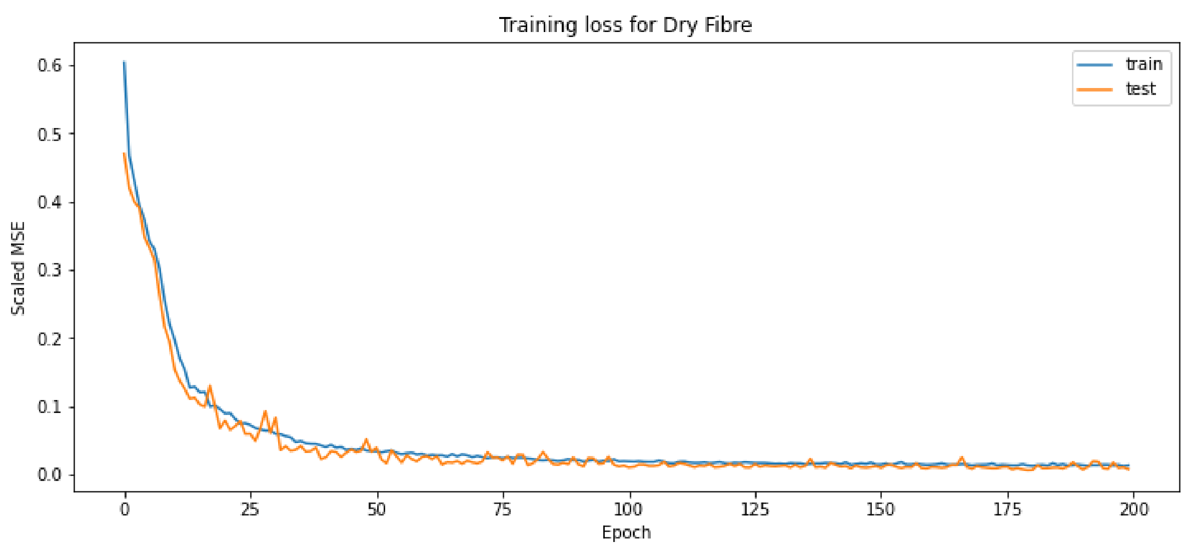

Figure 6: Training loss (blue) and testing loss (orange) plotted against the 200 epochs used in the LSTM RNN model.

Both the training loss and testing loss follows the same trend and no over-training of the model occurs. Presented in Table 4 are the RMSE values for the 3 sequence lengths tested using the LSTM RNN model.

Table 4: The impact of sequence length on RMSE for the LSTM RNN method.

Sequence Length	RMSE (h)
5	3.20
10	1.88
15	2.45

All three sequence lengths outperform the best performing ANN with an optimum sequence length of 10. This value was the closest to the mean WIP value. The RNN outperforming the ANN in this use case is unsurprising. Lead times are directly related to what is in the system with the sequence input of the RNN allowing for more detailed information about this to be used as an input. To give a visual indication of how the RNN model performs a plot of actual and predicted lead times is presented in Figure 7. The predicted and actual values were selected randomly from 100 continuous data points.

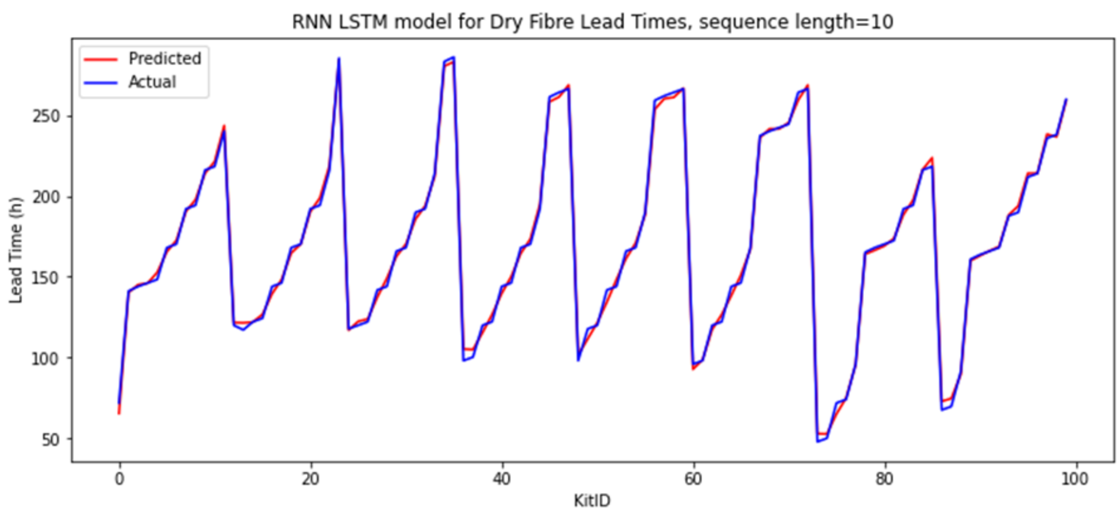

Figure 7: Lead time for dry fiber product types comparing the actual values (blue) and the predicted values (red) for a sample of 100 values. The predicted values used the RNN LSTM model where the sequence length = 10.

4.4 Linear Regression Results

Presented in this section are the linear regression results. The linear regression model was built to serve as a comparison to the more complex machine learning algorithms used previously. Presented in Figure 8 is a comparison of the actual scaled lead times for dry fiber products and the linear regression predicted values. Whilst the location of the peaks and troughs are captured reasonably well it is clear to see that the values of the peaks and troughs are not a good match. The linear regression model is also poor at predicting the lead time trend between the peaks and troughs. With the lead times between peaks and troughs being non-linear it is no surprise that this detail is not captured using this methodology. The RMSE value for the linear regression model is 5.99 hours which is more than three times the error of the linear regression and RNN LSTM model, with an RMSE equal to 1.88 hours (best case scenario). The relative error of the linear regression and the RNN LSTM model, for the mean lead time, was 3.5% and 1.1% respectively. The linear regression holds up better when compared to the ANN method. Half of the RMSE values using the Bayesian

Regularization algorithm performed worse than the linear regression model. Only 10% of the Levenberg-Marquardt results performed worse than the linear regression model.

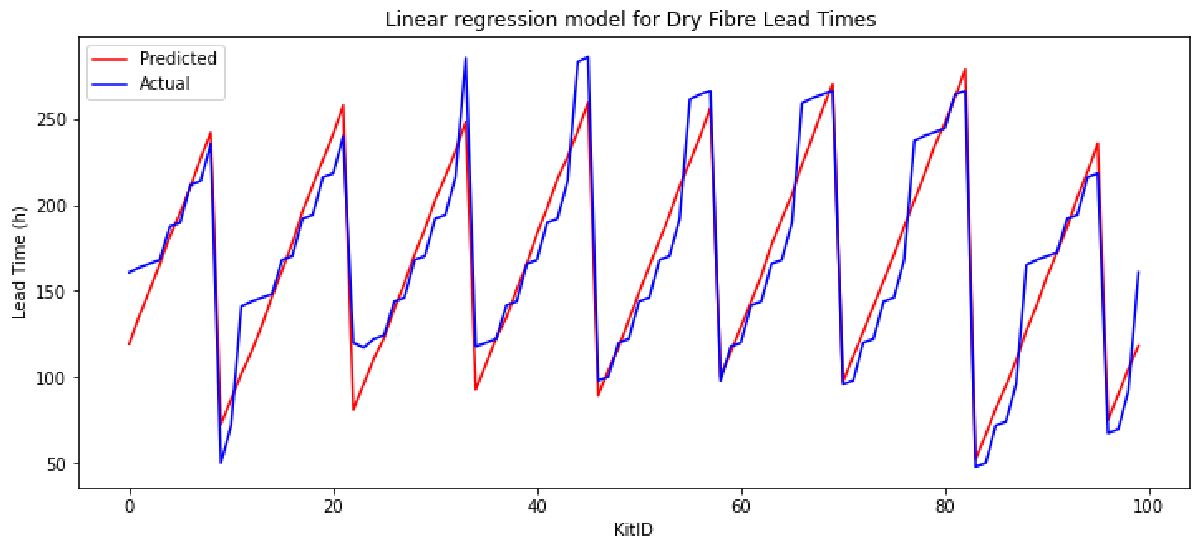

Figure 8: Lead times for dry fiber product types comparing the actual values (blue) and the predicted values (red) for a sample of 100 values. The predicted values use a simple linear regression model.

5 CONCLUSIONS

The article proposes a use case for artificial intelligence coupled with simulation for a manufacturing environment. Synthetic data was generated using discrete-event simulation of a composites prototype manufacturing center. The synthetic data was fed into a machine learning model that was used to predict manufacturing lead times. The purpose of the lead time calculator was to offer production planners an easy way to calculate the manufacturing lead time of a part based on the current state of the facility. The ML model required only a simple set of input data and did not need the services of a simulation and modelling expert. Initially three material types were selected for the simulation, with an artificial neural network and found the RMSE to be 103 hours. The poor prediction indicated that the model comprising of the three material types was too complex to be modelled by a single ML algorithm. The material types included were then reduced to look at the dry fiber process only. Three machine learning methods were compared. The best performing was the LSTM RNN with an RMSE value of 1.88 hours. Three sequence lengths of 5,10 and 15 were trialed with a sequence length of 10 demonstrating the best results. The sequence length of 10 was closest to the mean WIP value of 11. It can therefore be assumed that the sequence length should be approximately equal to the average WIP in the factory. The second-best performing was the ANN using a Bayesian Regularization algorithm, with 20 hidden neurons and a RMSE value of 4.65 hours. There was found to be no correlation between number of hidden neurons and RMSE value for both the Levenberg-Marquardt and Bayesian Regularization algorithms. The simple linear regression model was the worst performing method of the three, with a RMSE value of 5.99 hours. However, the linear regression did outperform 50% of the Bayesian Regularization and 10% of the Levenberg-Marquardt predictions.

ACKNOWLEDGMENTS

The authors would like to thank Matthew Williams for gathering data for the dry fiber and prepreg processes. Jamie Karl Smith would like to thank Dr Cansu Kandemir for taking the time to review the manuscript and for general discussions on AI and DES. Both authors would like to acknowledge the High Value Manufacturing Catapult for providing the funding to make this project possible.

REFERENCES

Behl, H.S., A.G. Baydin, R. Gal, P.H.S. Torr, V.Vineet. 2020. "AutoSimulate: (Quickly) Learning Synthetic Data Generation". In *Computer Vision – ECCV 2020*, edited by A.Vedaldi, H.Bischof, T.Brox and J.M. Frahm, 255-271. Cham, Switzerland: Springer.

Beregi, R., A. Szaller and B. Kadar. 2018. "Synergy of Multi-modelling for Process Control." *IFAC Papers online* 51(11): 1023-1028.

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.

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.

E.A. Lopez-Rojas, S. Axelsson. 2012. "Money Laundering Detection using Synthetic Data". In *The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS)*, edited by L.Karlsson and J.Bidot, 33-40. Linköping, Sweden: Linköping University Electronic Press.

Eyre, J. M., T. J. Dodd, C. Freeman, R. Lanyon-Hogg, A. J. Lockwood, R. W. Scott .2018. "Demonstration of an Industrial Framework for an Implementation of a Process Digital Twin". In *Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition*, November 9th-15th, Pittsburgh, USA, 1-9.

Greasley, A. and J. S. Edwards. 2019. "Enhancing Discrete-event Simulation with Big Data Analytics: A Review". *Journal of the Operational Research Society* 72(2): 247-267.

Grube, D., A. A. Malik and A. Bilberg. 2019. "SMEs Can Touch Industry 4.0 in The Smart Learning Factory". In *Procedia Manufacturing,* 31:219-224.

Heger, J., T. Voss. 2020. "Dynamically Changing Sequencing Rules with Reinforcement Learning in a Job Shop System with Stochastic Influences". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1608-1618. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hun Woo, J.,Y. In Cho, S. Hyeon Yu, S. Hyun Nam, H. Zhu, D. Hoon Kwak and N. Jong-Ho. 2020. "Machine Learning (Reinforcement Learning)-based Steel Stock Yard Planning Algorithm". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1560-1571. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Keras. 2021. "About Keras". https://keras.io/about/, accessed 21st December 2021.

Krause, T. 2020. "AI-Based Discrete-Event Simulations for Manufacturing Schedule Optimization". In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, January 6th –8th, Rabat, Morocco**,** 87-91.

Nikolenko,S. I. 2021. *Synthetic Data for Deep Learning*. 1st ed. Cham,Switzerland: Springer

Priore, P., B. Ponte, J. Puente and A. Gómez. 2018. "Learning-based Scheduling of Flexible Manufacturing Systems Using Ensemble Methods." *Computers & Industrial Engineering* 126: 282-291.

Reed, S., M. Löfstrand and J. Andrews. 2021. "Modelling Stochastic Behaviour in Simulation Digital Twins through Neural Nets." *Journal of Simulation*: 1-14.

Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp and A. Kyek. 2018. "Optimization of Global Production Scheduling with Deep Reinforcement Learning." *Procedia CIRP* 72: 1264-1269.

Turgut, Y., C. E. Bozdag. 2020. "Deep Q-Network Model For Dynamic Job Shop Scheduling Problem Based on Discrete Event Simulation". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1551-1559. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

JAMIE KARL SMITH is a simulation and modelling engineer at the University of Sheffield's Advanced Manufacturing Research Centre (AMRC). He earned his Ph.D. in Mechanical Engineering from University of Leeds where his thesis looked at simulating combustion strategies in spark ignition engines. He also earned a BSc in Physics from University of Leeds. His research interests include discrete-event simulation, artificial intelligence and digital twins. Particularly, he is focusing on how discrete-event simulation and Industry 4.0 techniques can be used in manufacturing applications. His email address is j.k.smith@amrc.co.uk.

CALUM DIKCINSON is a research engineer with 6 years experience in the composites industry. He graduated from University of Nottingham with a first class degree in Design Engineering. He has previously worked in a composite process development role at GKN Aerospace. His work now encompasses research surrounding automated composite forming technologies, robotics and process simulation. His email address is c.dickinson@amrc.co.uk.