Predicting Dynamic Process Limits in Progressive Die Sheet Metal Forming

D Budnick[1], A Ghannoum[1], F Steinlehner[2], A Weinschenk[3], W Volk[2], S Huhn[3], W Melek[1] and M Worswick[1]

[1] Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Canada
[2] Chair of Metal Forming and Casting, Technical University of Munich, Garching, Germany
[3] Forming Technologies, Hexagon Manufacturing Intelligence, Oakville, Canada

dbudnick@uwaterloo.ca

Abstract. Tool makers have a limited selection of tools and are afforded limited flexibility during progressive die try-outs when attempting to identify suitable process control parameters and optimize throughput. The performance of a given tooling design hinges on selecting a suitable stroke rate for the press. Cost efficiencies are realized when operating a press at higher stroke rates, but risk subjecting the sheet metal strip to larger, uncontrolled oscillations, which can lead to collisions and strip-misalignment during strip progression. Introducing active control to the strip feeder and lifters can offer increased flexibility to tool makers by allowing the strip progression to be fine-tuned to reduce strip oscillations at higher stroke rates. To alleviate uncertainties and assist in fine-tuning the process control parameters, machine learning models, such as an artificial neural network, are constructed to predict whether a given set of process parameters will lead to a collision or strip-misalignment during the strip progression. The machine learning models are trained using a dataset of FEA simulations which model the same progressive die operation using different process control inputs for the feeder, lifter and press. The machine learning models are shown to be capable of predicting the outcome of a given process permutation with a classification accuracy of about 87 % and assist in identifying the dynamic process limits in the progressive die operation.

1. Introduction

With the advancement of technologies and concepts that embody ‘Industry 4.0’, most industries have seen some semblance of impact from these technologies with sheet metal forming being no exception. It only takes a glance at literature review papers or conferences on sheet metal forming to notice the increase in papers that leverage advances in machine learning, sensing technology, and, advanced control. Work done by Jamli and Farid [1] reviews the use of Artificial Neural Network (ANN) to assist in predicting sheet metal spring back. They note that one benefit of using ANN is that it overcomes the complexities in constructing constitutive models that incorporate spring-back behavior and can learn directly from experimental data. Their review extends to the topic of aligning the benefits of Finite Element Analysis (FEA) and ANN to simultaneously reduce experimental costs and enhance prediction accuracy. Often these approaches focus on an inverse analysis and use the ANN trained on Finite Element (FE) simulations to more rapidly model a process with the goal of optimizing a set of process

parameters. Manoochehri and Kolahan [2] generated data from FEA to train an ANN which was used to find optimal deep drawing process parameters such as blank holder force by using simulated annealing in conjunction with the ANN to minimize thinning. With respect to sensing and control technology in sheet metal forming, a review by Polyblank *et al*. [3] provides a sense for the breadth of more recent advances. They point out that while most equipment used in sheet metal forming uses closed-loop control, current implementations seen in industry rarely account for the behavior of the product and instead focus on precisely following tool-paths. More recent approaches focus on incorporating the product properties into the control loop. One such approach implemented by Endelt *et al*. [4] controlled material flow in a deep-drawing process by measuring flange draw-in with laser displacement sensors and adapting binder pressure in a closed-loop control system. However as pointed out by Allwood *et al*. [5], most of the applications for novel control implementation in sheet metal forming focus on controlling product geometry or preventing failure.

There has been limited research, which investigates control strategies that seek to minimize the probability of collisions between the work piece and tooling, particularly during progressive die operations in which a sheet metal strip is fed into a series of forming stations (tooling). Examples of such research pertain exclusively to sheet metal forming processes which seek to find an optimal collision-free tool path by using mechanical grippers to transfer the sheet metal product between operations [6,7]. In some cases, the compliance of the sheet metal is even considered in the trajectory optimization problem [8,9]. However, in the case of progressive die operations, passive spring-operated lifters raise the strip and a strip-feeder, which is in-sync with the press stroke rate, is used to progress the product between die operations. With the exception of the press stroke rate, tool operators have limited control over progressive die operations. Without additional forms of control, progressive die operations have an inherent process speed limit for a given design since increasing the stroke rate will result in the strip exhibiting a larger dynamic response which eventually may result in a collision or reduced part quality [10]. By introducing adjustable inputs to the strip feeder and lifters, excessive strip dynamics can potentially be reduced at higher stroke rates, increasing the overall maximum process speed. However, the strip layout, and in particular, the stretch-web connector selection, will have a large influence on the dynamic response of the strip. Stretch webs must be designed with the necessary compliance to permit material flow in deep-drawing operations. If the stretch web is design with too much compliance, the strip may exhibit a larger dynamic response during strip progression, increasing the chances of a collision or poor placement of the product on the subsequent die.

This paper presents a systematic approach to predicting whether a collision will occur during a progressive die operation between the tooling and sheet metal strip. This approach generates data using FE simulations, which model a progressive die operation with varying inputs to the strip feeder, lifters, and press stroke rate. Data was generated for two strip layouts, which utilize different stretch-web connectors. The generated data is subsequently used to train a machine learning model, such as an ANN, which predicts whether a given process permutation will result in a collision between the sheet metal strip and the tooling or misalignment with the locating pins due to excessive strip dynamics. The goal of this work is to create a framework for identifying the dynamical process limit of a given progressive die operation and the limitation invoked by the stretch-web selection.

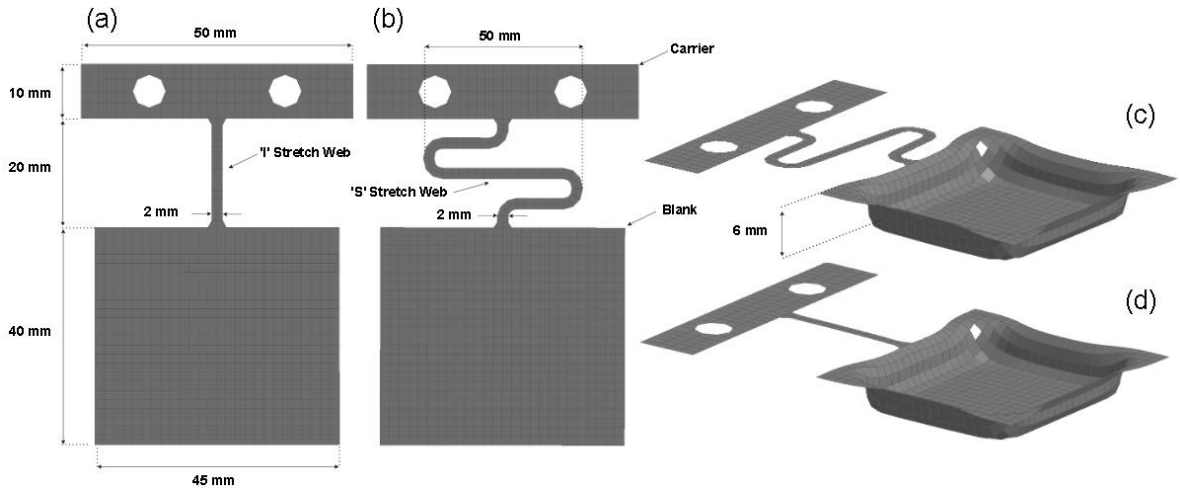

Figure 1: Mesh and dimensions for (a) 'I' stretch web strip (b) 'S' stretch web strip (c) 'S' stretch web with formed cup, (d) 'I' stretch web with formed cup.

2. Methodology

There are three distinct phases to this research, each of which are described in the following: (1) FE Modeling, (2) Data Generation, (3) Machine Learning. The physical problem being modelled comprises the transfer process and two different strip layouts are modelled using the same progressive die tooling and transfer system. The difference between these two strip layouts is limited to the stretch-web selection. As seen in Figure 1, the stretch-web in the strip layout is either selected as an 'I' or an 'S' web. A simple square-shaped cup is produced in the progressive die operation.

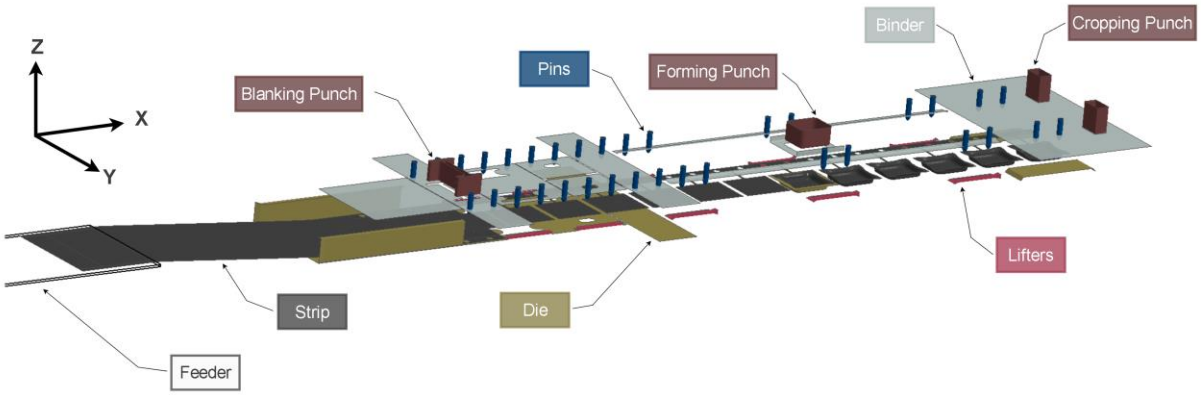

Figure 2: Simulation with labelled components. The strip is in its initial configuration, but, the binder, pins, and punches are lifted for visualization.

2.1. Finite Element Model

The FE simulation can be seen in Figure 2. All the simulations were completed using the LS-DYNA explicit dynamic solver R9.3 MPP with 16 cores on an Intel Xeon 8160 cluster. The progressive die operation consists of three fundamental operations: blanking, forming, and cropping. The current paper is focused on the dynamic response of the strip; consequently, the actual forming and trimming operations were not modelled and the part geometry was assigned as shown in Figure 1. The blanks ahead of the forming station were initialized using the flat blank geometry, while those after the forming station were assigned the formed part geometry. This simplification was adopted due to the long run-times of FE simulations, which stem from the requirement to model the dynamic response in real time without the use of mass scaling. The tooling elements were modelled as rigid bodies and penalty

function-based contact treatment was prescribed to capture collisions with the oscillating strip during the strip lift and advance motion. The simulation begins as the upper tooling is rising and ends once the binder clamps the strip at the next die station for a total of 256° of the crank angle. The progressive die operation was designed in a way to intentionally expose the strip to more risk of increased dynamic response and, as result, may not reflect design best-practices seen in industry. The reader is referred to the paper by Budnick *et al.* [10] in which the deformation of the part being formed is considered.

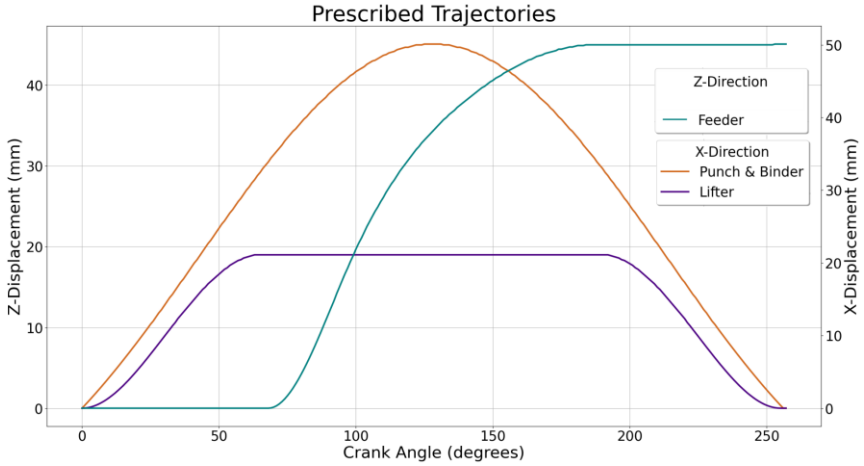

Figure 3: Prescribed trajectory for the binder, punch, lifters, and feeder.

The generated FE model is similar to that implemented in previous work by the current authors [10] with some notable differences. All motion of the punch, binder, feeder, lifters, and locating pins follow a prescribed motion. Additionally, the lifter actuation also takes place over a longer portion of the press crank angle then is possible with passive-actuation. All of the prescribed boundary conditions can be seen in Figure 3. The number of through-thickness integration points used for the stretch web was 5 and 3 for the rest of the strip. Since the focus of these simulations were to capture the dynamic response, which is largely an elastic behavior, the reduction of through-thickness integration points was justified for the advantage of shorter run-times. No mass scaling was applied in any of the simulations in order to capture the true dynamic response of the strip.

The blanks are modelled as rigid bodies to reduce the number of deformable elements and reduce the run-time. Fully-integrated shell elements are used for the rest of the strip. The strip is a 1 mm thick sheet of A5182 and modelled using a von Mises plasticity model with a yield strength of 121.4 MPa and isotropic strain hardening. Frequency independent damping was applied using 0.75 % critical damping over the first five natural frequencies. Penalty-based contact definitions were applied between the strip and the die, punch, pins, lifters, and binder for a total of five contacts.

Finally, only the first advance of the strip was modelled to simulate the initial dynamic oscillation. (In on-going work, multiple stroke simulations are being modelled to capture the steady-state response.) This initial response was used to train the machine learning models and a total of 1000 simulations were run, 500 for each strip layout.

2.2. Data Generation

The input permutations for the feeder and lifter motion used for each FE simulation were randomly sampled from a set of possible stroke rates and trajectories for the lifter and feeder. The permissible stroke rates varied between 60 and 360 strokes per minute (SPM).

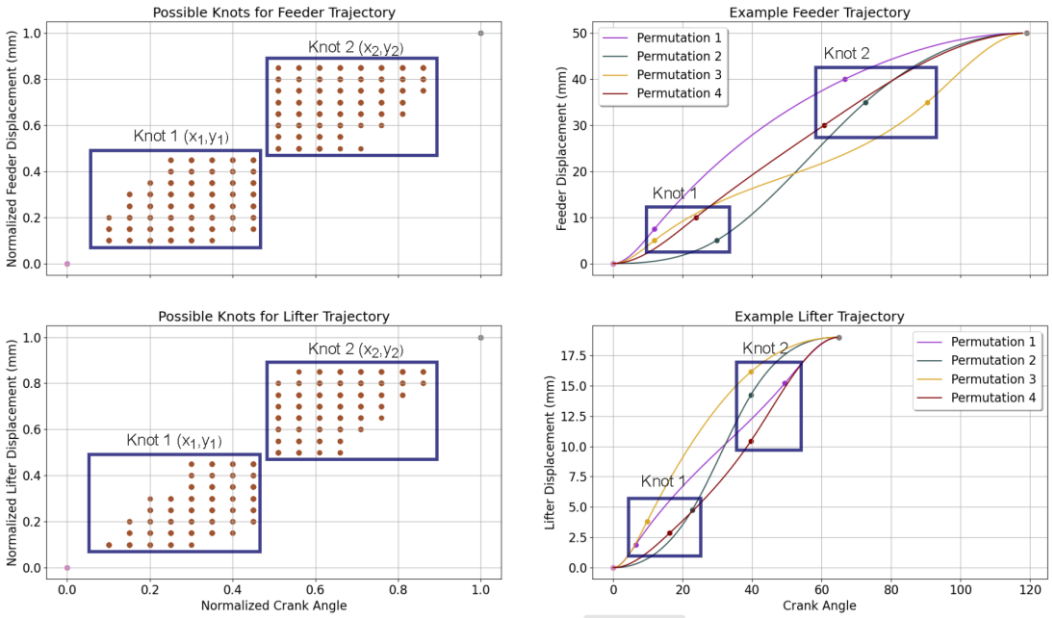

Figure 4: Possible knots for lifter and feeder trajectories (left) and examples of feeder and lifter permutations with the selected 'Knot 1' and 'Knot 2' indicated by the box (right).

2.2.1. Trajectory Selection. The set of possible trajectories for a given permutation were nearly identical for both the lifter and feeder. Cubic splines were used to define each trajectory using a total of four knots with boundary conditions that prescribe an initial and final velocity to be set to zero. Examples of possible trajectories can be seen in Figure 4. The first and last knots are fixed to the end points of the curve to define start and end points with only the inner-knots ('Knot 1' and 'Knot 2') being varied between permutations. To reduce the complexity of the solution-space to ensure that the ANN would have a reasonable chance to predict the behaviour of the system, further restrictions were applied to limit the size of the permutation space (Figure 4, left). All splines were required to be monotonically increasing to prevent the feeder or lifters from retracing their displacement and inverting their motion. As well, the velocity was restricted to avoid saddle points and intermittent periods of near zero velocity. More specifically, the normalized velocity was required to be above 0.25 between the inner two knots. Furthermore, the inner knots were restricted to a sampling area such that the X and Y normalized coordinates of 'Knot 1' are bounded by [0.1, 0.5] and 'Knot 2' is bounded by [0.5, 0.9]. Finally, the lifter trajectory was restricted to not rise faster than the upper tooling since this would cause a tooling collision.

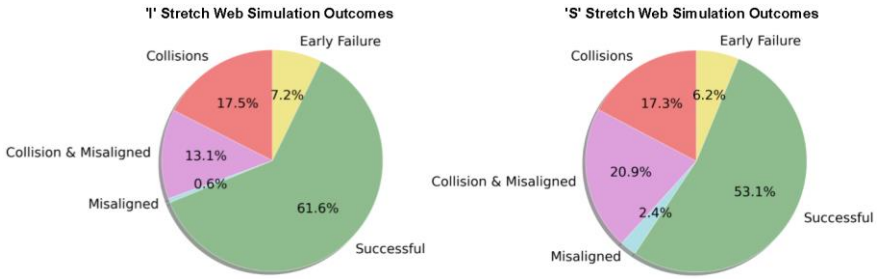

Figure 5: Simulation outcome for (left) 'I' stretch web connector and (right) 'S' stretch web connect.

2.2.2. Data Labelling. A critical component to constructing an effective machine learning model is first ensuring that the data being used is properly labelled and reflects the system that is being modelled. Referring to Figure 5, four possible outcomes were identified and labelled for each simulation:

'collision', 'strip misaligned', 'early failures', and 'successful'. (1) A simulation is labelled to have a 'collision' if there is a non-zero contact force with the tooling during the strip progression. This detection omits the initial strip lifting period since there can often be chatter between the binder and strip during this phase. (2) Referring to Figure 6, a 'misalignment' is labelled when the locating pins fail to locate the strip. Determining whether the pins accurately located the strip is done by measuring the position of a node at each locating hole and evaluating whether it is aligned with the pin within the tolerance allowed by the geometry, which is taken as 0.1 mm in this case. (3) 'Early failures' are simulations which showed erratic behaviour as a result of errors in the FE simulation setup. (4) 'Successful' simulations are any permutation that did not experience a collision, strip misalignment or early failure for which the strip was properly placed and located at the next die station.

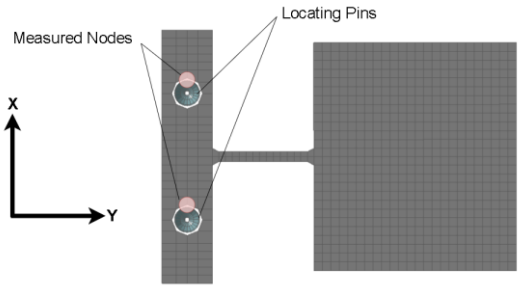

Figure 6: Single section of strip with labels indicating nodes measured for strip alignment.

With the outcome of each simulation labelled, the data is then prepared for the dataset used in the machine learning models. All of the simulations which resulted in early failures were removed from the dataset. The goal of the machine learning model is to predict whether a given set of process inputs will cause excessive strip dynamics and result in a poor outcome, or, in other words, a collision or strip misalignment. Any permutation that results in a collision or strip misalignment are classified as an unsuccessful permutation with all remaining permutations being classified as successful in this binary classification problem. Therefore, the inputs to the network are the feeder and lifter inputs and the stroke rate with the output being whether the simulation was 'successful' or 'unsuccessful'. The inputs and outputs of the network are summarized in Table 1**:** Labelled inputs and outputs for generated data. The data was split into a training and test set using an 80/20 split. The input variables for the training set were normalized and the same normalization scale was applied to the inputs of the test set.

Table 1: Labelled inputs and outputs for generated data.

	Inputs									Outputs
Variable	Feeder Knots				Lifter Knots				Press (SPM)	Succesful / Unsuccesful
	X_1	Y_1	X_2	Y_2	X_1	Y_1	X_2	Y_2	Stroke Rate	
Value	Normalized [0,1]									Binary {0,1}

2.3. Machine Learning

Two types of machine learning architectures were implemented and compared: (1) Artificial Neural Network, and (2) XGBoost. Both machine learning architecture were trained and tested using Python 3.8.3 on an Intel Core i9-9900K CPU. Separate models were implemented for each strip layout for a total of four machine learning models (two for each strip layout). After removing the simulations that resulted in 'early failures', the 'I' and 'S' stretch webs had dataset sizes of 461 and 466 samples respectively. ANN's have been shown to perform well on a wide variety of non-linear modeling problems and have been used widely in the literature. XGBoost was also chosen to be used as a comparison against the ANN as it has shown to perform well on a wide variety of problems and handle smaller datasets well, even with a larger feature size. XGBoost is a decision-tree based model, which uses an ensemble learning method based on the random forest approach to design a classifier using gradient boosting [11].

To select the hyperparameters used for each model, two rounds of hyperparameter grid-searches were performed on both models using the training data. To evaluate the performance of the hyperparameter selection for each model, K-fold cross validation (KCV) was used with 5 folds to assess the generalizability of each model by measuring the variance and average test accuracy. KCV is implemented by randomly splitting the dataset into K-folds of even size and training the model on K-1 folds and testing on the remaining fold [12]. Once hyperparameters are selected, the model is trained using the entire training data and tested on the test set to evaluate the generalizability of the selected models from the KCV hyperparameter selection procedure. The same architecture and hyperparameters were found to work well for both stretch webs and therefore the parameters were not changed between datasets. The architecture of the selected ANN has a single hidden layer with 8 neurons using a 'tanh' activation function and a single network output with a 'sigmoid' activation function and was implement using TensorFlow 2.6.0. The network was trained using small-batch gradient descent algorithm with 50 epochs, a batch-size of 32 and a learning rate of 0.005 using the Adam optimizer. The two ANN models were trained on FE simulation data for the 'I' and 'S' stretch web data respectively and used binary cross-entropy loss which can be seen in equation (1):

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (1)$$

where N is the number of data samples, y_i is the 'true' label of a data point, and $\hat{y}_i$ is the output of the machine learning model. The selected XGBoost algorithm was implemented with XGBoost version 1.4.2 for Python and uses a learning rate of 0.1, a max tree-depth of 2 and the regularization term, gamma, set to 0.1. The two XGBoost models were trained on the data collected for the 'I' and 'S' stretch web from the FE models and used a binary cross-entropy loss function.

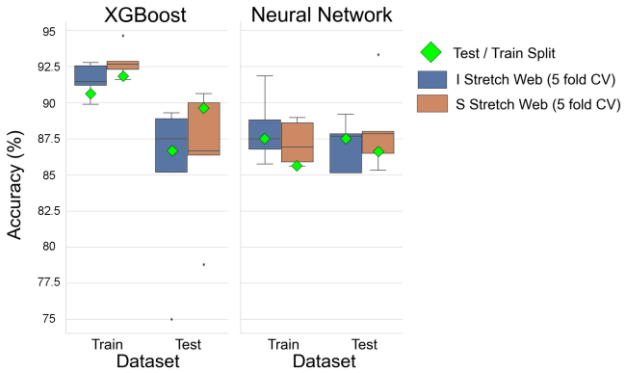

Figure 7: 5-Fold cross-validation accuracy, and test and training accuracy for each model.

3. Results & Discussion

To measure the performance of the machine learning models, prediction accuracy is assessed in terms of the number of correctly predicted simulation outcomes divided by the total number of simulations. The training and test results for the ANN and XGBoost algorithm for both stretch webs can be seen in Figure 7. The KCV results are shown using the box plots while performance of the selected models trained on the entire training data are indicated by the triangles. In general, all models performed reasonably well with an average test accuracy of about 87 %. As well, both the XGBoost and ANN models generally performed comparatively on the 'I' stretch web data and the 'S' stretch web data for both test and training sets indicating the data for both strip layouts is equally separable. Furthermore, the XGBoost models had a larger discrepancy between the test accuracy and training accuracy, as compared to the ANN, implying that the XGBoost models were subject to more overfitting. The discrepancy between prediction accuracy on the datasets reflects the hyperparameter tuning process for the XGBoost model, which was highly sensitive to overfitting the training data. Finally, both models appeared to generalize well as they achieved similar accuracy on the test set as compared to the KCV results.

To demonstrate the predictive capability of the machine learning models, the FE simulation predictions of the minimum distance measured along the vertical (press stroke) direction between the strip and binder during the strip progression are plotted as a function of stroke rate and coloured by the prediction of the XGBoost classifiers for both stretch webs overlaid (Figure 8). The predicted distance in the FE models provides a quantitative measure of how close the control inputs of the feeder and lifter were to promoting a collision. If the minimum distance is small or negative, it indicates that portions of the strip overshot the position of the upper tooling, which makes it highly susceptible to a collision. The minimum distance between the binder and strip is calculated by measuring the center-of-mass of the nearest blank with respect to the binder and therefore there will be cases for which there is a collision when there is still a positive distance. The scatter plot has large regions of correctly predicted 'successful' permutations in the upper left corner where the stroke rate tended to be low and there was a large distance between the binder and strip. Inversely, there was a large collection of conditions correctly predicted as 'failures' in the bottom right corner of the plot where the stroke rate tended to be higher, and the distance was minimal. It should be noted that when the minimum distance is negative, the strip has overshot the binder, which is likely around the forming station where there is a gap in the binder.

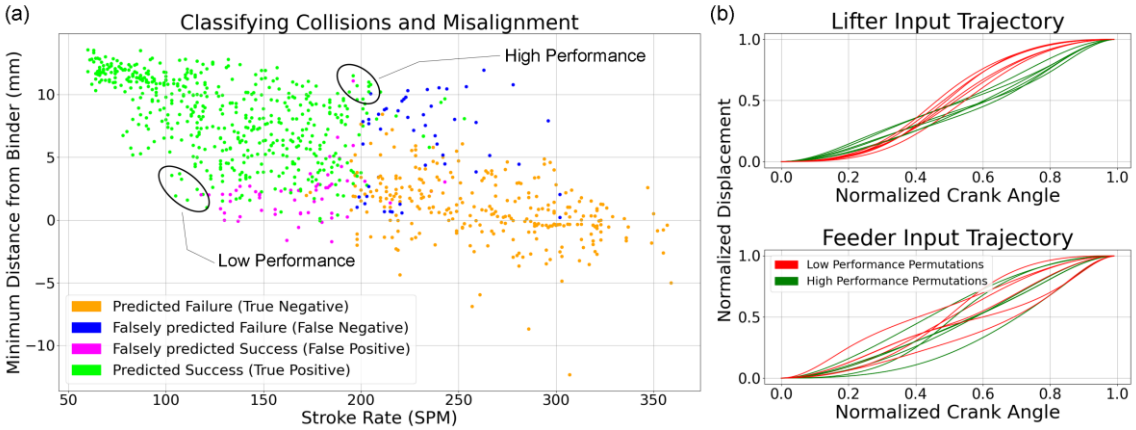

Figure 8: (a) Plot of stroke rate *vs.* the minimum vertical distance along the press stroke direction between the strip and binder as predicted by the FE models. The symbols are coloured based on the prediction of the XGBoost classifiers trained on the entire training set and evaluated on the test and training sets for each stretch web. (b) Feeder and lifter trajectories for three of each stretch web types which are 'low performance' (red) and 'high performance' (green) permutations, as indicated by the ellipses in (a).

Orange and green samples are correctly predicted 'failures' and 'successful' respectively (true negative and true positive). The pink and blue samples are permutations that were incorrectly predicted as 'successful' or as 'failures' respectively (false positive and false negatives). These misclassified permutations are the result of the given permutations outperforming the trend (false negatives) by avoiding failures at higher strokes and underperforming (false positives) by failing at relatively lower stroke rates.

Figure 8 (b) shows examples of feeder and lifter input trajectories for select permutations that were either 'low performing' and had minimal distance at lower stroke rates or were 'high performing' and achieved high stroke rates with relatively larger distance between the strip and binder. The selected permutations were comprised of three of each stretch web type for a total of six 'high performing' and six 'low performing'. The 'high performing' permutations follow a more linear path for the lifter input whereas the 'low performing' lifter inputs followed a path similar to a sigmoid function. However, in the case of feeder inputs, the trends are less evident, with a variety of inputs shapes comprising both the high and low performance permutations. This lack of trend is explained by the fact that the minimum

distance between the binder and strip is related to how much the blank overshoots the nominal lifting value and is directly influence by the lifter trajectory and not the feeder. The feeder trajectory has a larger impact in causing the strip to be misaligned with respect to the locating pins as well as how the blank is placed on the next tooling station.

In both cases, the excessive dynamic response of the blank is the source of the collision or strip misalignment and is initiated during the strip lifting. Once there is an excessive dynamic response of the strip, then regardless of the feeder trajectory, a collision or strip misalignment is highly probable. Therefore, since the lifter trajectory is responsible for causing excessive excitations to the blank which can result in collision or strip misalignment, there is a clear trend with respect to the lifter inputs and the minimum distance measured between the strip and binder in the FE simulations.

Ultimately, Figure 8 sheds light as to where the machine learning models struggled, and the data becomes difficult to predict. The models perform well at higher and lower stroke rates since regardless of the feeder and lifter inputs, the model can interpret based on the SPM whether or not a failure is likely. In the midrange of stroke rates where the feeder and lifter inputs tend to play a more integral role, the model begins to struggle at classifying all the results properly. The reduced classification accuracy in the midrange of stroke rates is due in part from the limited samples in both datasets and the size of the permutation space that the models are trying to learn. As well, there is a clear correlation between the stroke rate and failure and all the machine learning models will naturally assign larger weights to the input of the stroke rate and lower weights to the feeder and lifter inputs. Machine learning, and in particular, ANN provide the flexibility to employ techniques such as data augmentation, to generate additional data points, and transfer learning, to retrain the ANN models, to enhance the classification accuracy where there is opportunity for improvement, such as the midrange of SPM. Nonetheless, the current machine learning models can identify with reasonable accuracy the dynamical process limit of the progressive die operations by testing new feeder and lifter inputs and identifying the maximum stroke rate at which the model predicts a ‘successful’ outcome.

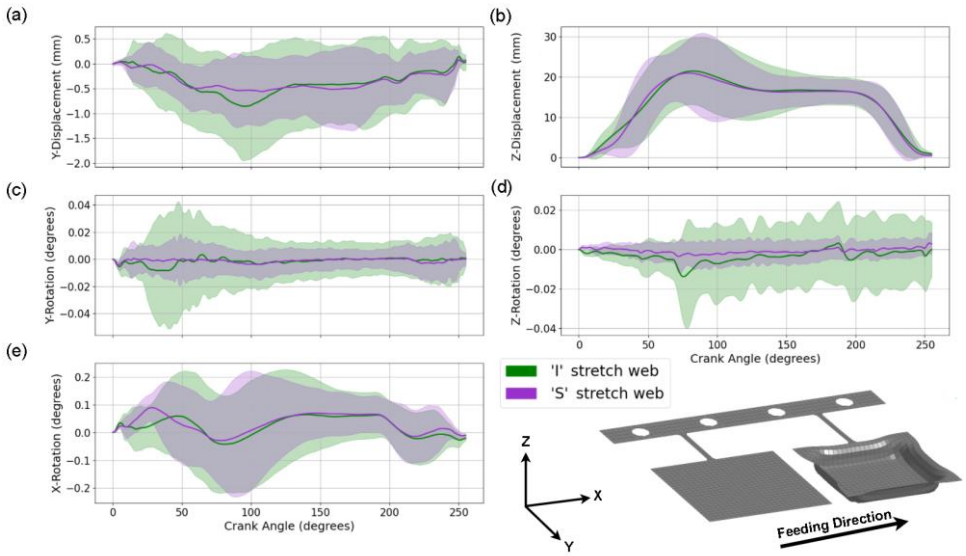

Figure 9: Dynamic response of the blank measured for ‘successful’ permutations in (a) y-displacement, (b) z-displacement, (c) y-rotation, (d) z-rotation, (e) x-rotation. The thin lines represent the mean and the coloured regions bound the mean by two standard deviations.

While both stretch webs appear to show a similar trend between the minimum distance from the binder and the stroke rate (Figure 8), the differences between the stretch webs are clearer when directly measuring the dynamic response of a given blank measured in the FE models (Figure 9). More specifically, by plotting two standard deviations from the mean dynamic response for each stretch web, the general behavior is more readily compared. In general, the ‘S’ stretch web experiences larger

variability in the dynamic response, particularly when comparing the rotational response in the 'y' and 'z' orientations. The response in the z-direction is of particular importance since any rotation about this axis will influence the rotational position of the blank once it is placed on the die face (xy-plane) and will influence the final formed geometry of the blank. The large variance in the dynamic response of the 'S' stretch web can be attributed to its more compliant structure which will permit material flow during deep drawing operations but is more susceptible to an excessive dynamic response.

4. Conclusion

The machine learning models presented in this work are capable of predicting whether a collision or strip misalignment will occur after a single stroke with a classification accuracy of about 87 %. By adjusting the inputs to the feeder and lifter, a wide range of outcomes are observed with certain permutations underperforming and resulting in failures at lower stroke rates and others outperforming with no failure at higher stroke rates. By utilizing a collision prediction model in conjunction with active control to the feeder and lifters, tool makers can potentially realize higher stroke rates for a given strip layout and stretch web selection or predict the limiting stroke rate prior to die try-outs.

Acknowledgements

The authors appreciate and acknowledge the financial support from the Ontario Advanced Manufacturing Consortium, Natural Sciences and Engineering Research Council of Canada, the Industrial Research Assistance Program (IRAP), the Canadian National Research Council (NRC), the German Aerospace Center (DLR) and the German Federal Ministry of Education and Research (BMBF).

References

[1] Jamli M R and Farid N M 2019 The sustainability of neural network applications within finite element analysis in sheet metal forming: A review *J. Int. Measur. Confed.* **138** 446–60
[2] Manoochehri M and Kolahan F 2014 Integration of artificial neural network and simulated annealing algorithm to optimize deep drawing process *Int. J. Adv. Manuf. Technol.***73** 241-9
[3] Polyblank J A, Allwood J M, and Duncan S R 2014 Closed-loop control of product properties in metal forming: A review and prospectus *J. of Mater. Process. Technol.* **214** 2333–48
[4] Endelt B, Tommerup S, and Danckert J 2013 A novel feedback control system - Controlling the material flow in deep drawing using distributed blank-holder force *J. of Mater. Process. Technol.* **213** 36–50
[5] JAllwood J M *et al.* 2016 Closed-loop control of product properties in metal forming *CIRP Ann – Manuf. Technol.* **65** 573–96
[6] Han G 2016 *Transfer Die System Timing and Parameter Optimization According to an Obstacle Map* Master's theses (Vancouver, BC: University of British Columbia)
[7] Liao X and Wang G G 2003 Evolutionary path planning for robot assisted part handling in sheet metal bending *Robot. and Comp.-Integ. Manuf.* **19** 425–30
[8] Glorieux E 2017 *Multi-Robot Motion Planning Optimisation for Handling Sheet Metal Parts* Doctoral dissertation (Trollhättan, Sweden: University West)
[9] Li H and Celglarek D 2002 Optimal Trajectory Planning For Material Handling of Compliant Sheet Metal Parts *J. Mech. Design* **124** 213-22
[10] Budnick D *et al.* 2021 Simulation of Dynamic Effects in Progressive Die Operation and Control *IOP Conf. Series: Material Science and Engineering (Stuttgart)* vol. 1157, no. 1
[11] Chen T and Guestrin C 2016 XGBoost: A scalable tree boosting system *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (San Francisco)* vol. 13-17 (New York, NY, USA: ACM) pp 785–94
[12] Hastie T, Tibshirani R, and Friedman J 2017 *The Elements of Statistical Learning Data Mining, Inference, and Prediction* vol 2 (New York: Springer) chapter 7 pp 241-7