Spring Back Prediction Using Gated Recurrent Unit and Data Augmentation

Du Chen[1], Frans Coenen[1], Yang Hai[2], Mariluz Penalva Oscoz[3], and Anh Nguyen[1]

[1] Department of Computer Science, The University of Liverpool, Liverpool, L693BX, UK,
[2] EITRI (Nanjing Eurosmart Intelligent Technology Research Institute), 9 Chengxin Avenue, Jiangning Development Zone, 21000, Nanjing, China,
[3] TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Cientıfico, Parque Cientıfico y Tecnologico de Gipuzkoa, 20009 Donostia-San Sebastian, Spain

Abstract. Artificial Intelligence (AI) has been widely used in manufacturing, healthcare, sports, finance, and other fields to model nonlinearities and make reliable predictions. In manufacturing, AI has been applied to improve processes, reduce costs and increase reliability. A new manufacturing process enhanced by AI is Single Point Incremental Forming (SPIF), a technique that uses a computer numerical control (CNC) machine to incrementally feed a metal sheet or polymer blank. However, achieving the geometric accuracy of the process is still of primary challenge due due to the impact of spring back. One of the most common solutions is toolpath correction. In this paper, we proposed a mechanism to capture local geometry using a novel point series representation, which then forms a general global geometry information. Each point series can then be associated with a predicted spring back value and learn using deep learning. In particular, this article proposes the use of data augmentation to solve the problem of insufficient data and enable deep learning models to achieve the better performance. Intensive experimental results show that we achieved the best R2 or "coefficient of determination" of 0.9228 compared to recent methods. We show that the proposed method provides a realistic solution to the current limitations of SPIF.

Keywords: Single Point Incremental Forming, GRU, Data Augmentation, Spring Back Prediction, Deep Learning

1 Introduction

Single Point Incremental Forming (SPIF) [10] is a sheet metal cold forming technique in which a metal sheet is clamped along its perimeter and then shaped incrementally using a computer numerical control (CNC) machine which controls a tool head that follows a predicted tool path to incrementally push out the desired shape as shown in Figure 1. The sheet metal is held in place, and the tool head is moved according to the planned "tool path" to extrude the desired shape (the example in the Figure 1 is a flat-topped pyramid). Compared

with traditional forming methods, SPIF has the advantages of low cost, short production cycle and flexible process. The tool path is determined according to the desired shape. The advantage of cold forming is that the sheet metal does not need to be heated before forming takes place, as in the catalogue "hot forming". Cold forming is therefore cheaper and more environmentally friendly than hot forming. The disadvantage of SPIF cold forming is that the process features spring back, a property of metal material whereby the material tries to return to its original shape after bending. This means that parts evaluated using SPIF are not exactly the shape intended. This is significant concerning all industries, significant in industries where part accuracy is important, for example, the aeroplane manufacturing industry. If the spring back problem can be addressed this will provide a significant advantage to industries such as the airplane manufacturing industry.

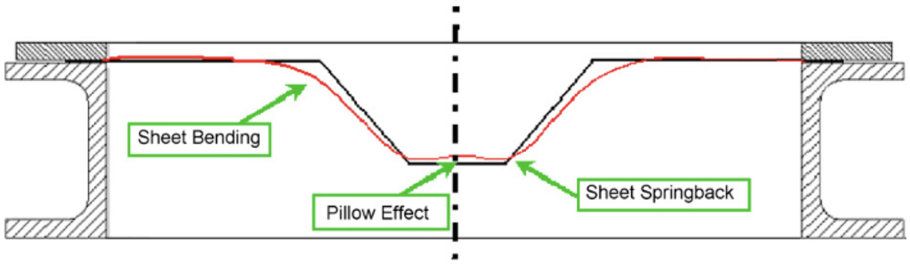

Fig. 1. Typical geometric errors in SPIF

Artificial intelligence applications that forecast SPIF geometric deviation (spring back) have drawn a lot of attention recently. Artificial neural networks (ANNs), deep and shallow back propagation neural networks, regression and other AI techniques have all been tested by researchers [1][11][15]. In [7][8], an intriguing strategy was put up in which the local geometry is represented as a series of points. The goal is to create a collection of local geometries that is a superset of all possible component local geometries for any produced part. The research provided in [7][8] uses the k-nearest neighbour (kNN) classification technique, which is a well-established machine learning classification technique, but defines rebound in terms of a set of classification labels.

In this work, our goal is to predict the spring back error of Single Point Incremental Forming from the data of CNC machine using deep learning. We aim to facilitate the application of the resulting spring back error values to the CNC machine when spring back predictions are obtained. As such, we proposed the fundamental idea is to use tools and techniques of machine learning (GRU [3]) to predict spring back so that a correction can be applied to the desired shape specification that will take spring back into the account. More specifically, the idea is to train a machine learning model to predict spring back from historical data representation.

The contribution of this paper mainly focuses on the effective data augmentation of the original data set, which significantly increases the size of the data set without changing the order of the original data set. The most obvious is that when grid size d = 50mm and step value = 1, the size of the data set is expanded from the original 16 to 55571, which is 3473 times larger. This will significantly improve the training set and saturation of deep learning, and improve the robustness of the model. At the same time, the LSTM used by previous researchers was replaced with a simpler and easier-to-use GRU model. The evaluation model used is R2, and the experimental results show that the best result of the model has increased by 1%, from 0.9140 to 0.9228.

The rest of this paper is organized as follows. Section 2 presents the related work underpinning the work of this report. Section 3 presents the methodology of the overall experimental model. The obtained evaluation results are presented and discussed in Section 4. The report in Section 5 is the summary of this experiment and the goals for future work.

2 Related Work

The potential of SPIF has sparked a significant quantity of published research that aims to address the difficulties presented by the technology. The spring back error of the SPIF has been researched by numerous academics in the past using 19 different regressions, including multiple linear, regression trees, SVM, Gaussian process, and ensemble learning [14]. Some researchers [2][9] use genetic algorithm and other optimization algorithms, combined with methods such as finite element simulation, to optimize the forming parameters, so that the degree of spring back of the formed parts is reduced. For example, the forming parameters are optimized by genetic algorithm, and the formed parts with a small degree of spring back are obtained and some researchers [13] also optimize the forming parameters and sheet shape by combining finite element simulation and machine learning methods, so that the degree of spring back of formed parts is reduced. For example, using finite element simulation and neural network-based optimization algorithms, formed parts with less spring back are obtained. The report's distinction is that it replaces multiple machine learning technologies while also improving the data set before adding it to the GRU model for training.

The work of [4][5][8][14] is noteworthy. In [4], Local Geometry Matrix (LGM) and Local Distance Measure (LDM) are proposed, using the grid form to represent the local geometric shape. In [5], a new data representation — point series is proposed. These studies all advocate the use of various deep learning models to predict spring back based on the local geometry represented as a series of points, and then consider applying the determined spring back values in reverse to correct the tool path.

The point series in this research were generated by dividing the shape to be manufactured into a set of equal-sized grid squares. Each grid square was defined by a set of [x, y, z] coordinates that were referenced to a specific origin. The desired point series was generated by examining the neighbouring grid

squares of each current grid square and subsequently determining the difference in the z value between them. This approach was demonstrated to be effective, and a similar method is employed in the research presented in this paper. However, there are a few key distinctions between this work and previous studies [7][8][4][6]. In earlier studies, the difference in z value was expressed as a set of nominal values, while in this study, spring back prediction was conducted through classification, which was necessitated by the limited processing power available to the researchers at the time. Classification is a type of supervised learning that involves training a classifier using a pre-labelled data set. Using a set of pre-labelled data, a classifier is trained. For comparing the similarity of time series, the k Nearest Neighbour (kNN) model in combination with Dynamic Time Warping [12] was employed as the classification model. In [14], the proposed idea is to use LSTM to predict the spring back error, because the spring back occurs sequentially during the board pressing process, and secondly, the previous shape also influences the subsequent shape.

3 Methodology

3.1 Background

To generate the model training data was required; a data set $D_{train} = \{T_1,T_2,...\}$ where each $T_i \in D_{train}$ is a fragment of shape geometry that has been labeled with an identified spring back value e. There are variety of ways in which fragments of geometry can be represented. As noted earlier, in [5] and [4] point series were used. Thus each $T_i \in D_{train}$ comprises a tuple of the form $\langle$P,e$\rangle$, where P is a point series P = $[p_1,p_2,...]$ and p_j is a point value in the point series, and e is a spring back value.

The input comprises two point clouds F_{in} and F_{out} such that: F_{in} describes the piece to be manufactured, generated from the Computer Aided Design (CAD) software package originally used to specify the piece to be manufactured; and F_{out} describes the resulting manufactured piece, generated using 3-D optical photogrammetry.

Each $g_i \in G_in$ was represented as a labeled point series P = $\langle$P,e$\rangle$ = $\langle[p_1, p_2, ..., p_n]$, e$\rangle$ where p_j is the difference in z value between grid square g_i and its neighbour g_j, and e is the corresponding spring back value in E.

Once the set of point series $\{p_1, p_2, ...\}$, and the set of errors $\{e_1, e_2,...\}$ has been calculated the point series and errors can be incorporated into the training set D_{train} =$\{T_1, T_2, ...\}$ where each $T_i = \langle P_i, e_i\rangle$.

3.2 Data Augmentation

Data sets with various grid sizes are shown in the Table 1 below both before and after data augmentation. The size of the data set is not even 100 when the grid size is 30mm, 40mm, or 50mm. An intricate model like deep learning cannot be trained with such a meagre amount of data. Underfitting is a simple consequence

of training in this situation. This explains why there is a difference between the spring back prediction results and other grid sizes in the research [14] when the grid size is 50mm. I considered using the data augmentation approach to solve this issue, and because the data's order cannot be changed, I decided to employ overlapping.

The Table 1 three right-hand columns display the size of the data set after it has been overlapping using various step sizes. There are 1, 3, and 5 pixels of offsets, respectively. We can observe that using overlapping considerably increases the data set size for all different grid sizes. When overlapping is employed, the values are reduced to 1.9 times, 1.1 times, and 1.8 times, respectively, from the original data's value of the data set with grid size 5mm being 265 times more than that of the data set with grid size 50mm. And when overlapping is used with a step value of 1, the values for the dat aset with grid size 50mm are increased by 3743, 375 and 143 times from the value of the original data, respectively. When the size of the effective data set is increased, it can fully demonstrate the advantages of deep learning.

Table 1. Data Augmentation

Grid Size	Original	Overlapping step_1	Overlapping Step_3	Overlapping step_5
5	4253	107313	7020	4253
10	1006	101922	10340	4098
15	396	96125	10697	3891
20	223	90116	9055	3667
30	81	78043	8718	3195
40	36	66507	7359	2731
50	16	55571	6004	2291

3.3 Spring back Prediction with GRU

In [4], S. EI-Salhi et al. proposed the generic Representation And Springback Prediction (RASP) framework used to generate spring back classifiers. The RASP framework on which the proposed surface representation techniques are founded is comprised of three processes: (i) data preprocessing and error calculation, (ii) surface representation and (iii) classifier generation. The Figure 2 shows process of RASP.

The first phase is concerned with generating the advocated grid representation with associated error values. During the second data representation phase the grid representation is translated into the proposed surface representation techniques — Point Series. The third phase comprises classifier generation and evaluation. Once an appropriate classifier has been generated it can be used to predict grid labels associated with unseen grids. In the case of sheet metal form-

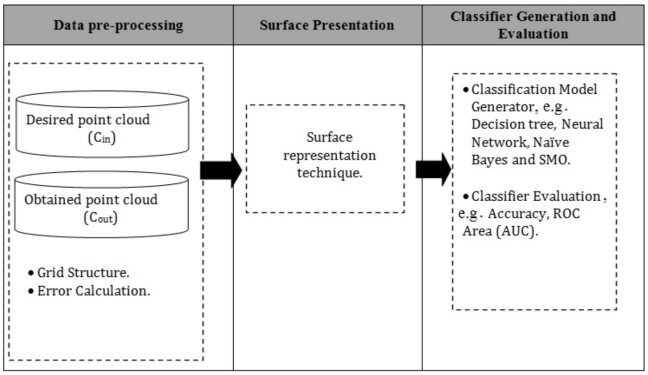

Fig. 2. Representation And Springback Prediction

ing applications, these will be the spring back errors associated with new shapes to be manufactured.

This report also adopts this framework to achieve the initial goals. Base the spring back prediction using point sequences and deep learning [14]. In this work, the data sets with grid sizes of 30mm, 40mm, and 50mm are very small and not enough for deep learning models to train well. In response to this problem, a data augmentation method is proposed to increase the size of the training data set, because model shaping is a sequential process, so overlapping is a very good choice compared with data augmentation methods such as rotation, overlapping is still a sequential process. And I chose a model that is simpler than LSTM - Gated Recurrent Unit (GRU).

When generating a GRU model, for each time period t, the GRU gets one input x_t, and computes a new cell state ct and a hidden state ht based on the new input and memory from the previous time period as shown in Equation 1 where h_t is the hidden state at time t, x_t is the input at time t, h_{t-1} is the hidden state of the layer at time t-1 or the initial hidden state at time o and r refers to the reset gate, z stands for the update gate, n is the new gate; σ stands for sigmoid function, and * stands for Hadamard product.

$$\begin{aligned} r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \\ z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \\ n_t &= tanh(W_{in}x_t + b_{in} + r_t * (W_{hn}h_{(t-1)} + b_{hn})) \\ h_t &= (1 - z_t) * n_t + z_t * h_{t-1} \end{aligned} \tag{1}$$

3.4 Implementation Detail

Figure 5 provides a summary of the RGU method that was used.The operations of the GRU model was compared with an SVM Regression model and LSTM-MLP model [14]. SVM and LSTM-MLP were implemented in Python using the

Keras Python package while GRU was implemented in Python using the Pytorch Python package. As noted earlier the GRU architecture comprised: (i) a GRU layer with 128 internal units, (ii) a dense (fully connected) layer with 256 nodes, (iii) another dense layer with 64 nodes, and (iv) a final dense layer with a single node to predict the spring back error. To accelerate the training of the predictor a momentum of m = 0.01 was used; together with a learning rate of $\alpha = 0.001$ and an epoch size of e = 1000. The input data set was split into training and validation subsets using a ratio of 8: 2. Also, to further accelerate the training process, the data were grouped into batches and the batches size is 8.

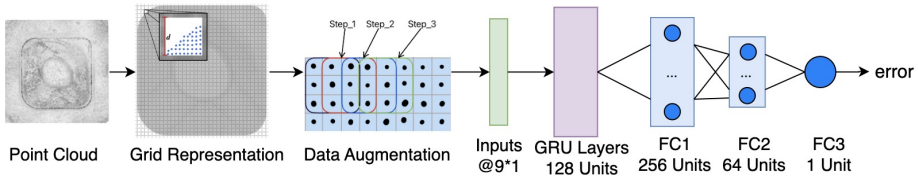

Fig. 3. Fix this figure GRU Network Architecture

4 Evaluation

4.1 Experiment Result

As mentioned in the previous section, to address the issue of insufficient original data when the grid size is 30mm,40mm and 50mm. First attempt uses overlapping to increase the original data. The combined data volume of all original data sets may rise due to overlap. Expanding the data set can help to lower mistake rates. Based on this, the GRU (Gated Recurrent Unit) model, a gated RNN, is adopted to greatly alleviate solve the gradient disappearance problem of RNNs through the gating mechanism, while making the structure simpler and maintaining the effect of LSTM (another widely used gated RNN).

The objectives of the evaluation are as follows:

1. What does the proposed system compare with the most recent previous work [14]?
2. What is the best step value?
3. What is the best grid size?

Table 2 shows the results of the current study compared with earlier studies [14]. The first two columns of SVM and LSTM list methods and results from earlier studies [14]. The third column uses the original data set and the GRU method. The fourth, fifth, and sixth columns still use the GRU technique, but with different overlapping data sets. Since each column uses a unique offset

Table 2. Compare with Previous Work

	Grid size	SVM	LSTM	GRU(New)	GRU(ours) Step_1	GRU(New) Step_3	GRU(New) Step_5
MAE	5	0.4569	0.3067	0.4729	0.2566	0.4737	0.4055
	10	0.3592	**0.3002**	0.3726	0.2573	0.4737	0.4219
	15	0.3239	0.3129	0.4804	0.2613	0.3458	0.4834
	20	0.2925	0.3028	0.5180	0.2957	0.4728	0.4208
	30	**0.2877**	0.3274	0.3705	0.2988	0.3009	0.4035
	40	0.3058	0.3070	0.3463	**0.2354**	0.3565	0.2986
	50	0.3907	0.4423	**0.2843**	0.3367	**0.2904**	**0.2539**
MSE	5	0.3637	0.1580	0.3585	0.1544	0.4086	0.2821
	10	0.2206	**0.1596**	0.2630	0.1476	0.3720	0.3702
	15	0.1784	0.1664	0.4409	0.1455	0.2312	0.4609
	20	0.1452	0.1640	0.4906	0.1747	0.3835	0.3432
	30	**0.1366**	0.2313	0.2876	0.1634	0.1760	0.3635
	40	0.2164	0.2318	0.2246	**0.1046**	0.2481	0.1706
	50	0.3529	0.4391	**0.1547**	0.2313	**0.1567**	**0.1287**
RMSE	5	0.6005	**0.3693**	0.5988	0.3930	0.6392	0.5312
	10	0.4688	0.3981	0.5128	0.3842	0.6099	0.6085
	15	0.4215	0.4071	0.6640	0.3814	0.4808	0.6789
	20	0.3796	0.4043	0.7004	0.4179	0.6193	0.5858
	30	**0.3668**	0.4807	0.5363	0.4043	0.4195	0.6029
	40	0.4598	0.4703	0.4739	**0.3235**	0.4981	0.4131
	50	0.5934	0.6579	**0.3933**	0.4809	**0.3959**	**0.3587**
R2	5	0.8036	**0.9140**	0.8116	**0.9228**	0.7744	**0.8518**
	10	0.8518	0.8921	**0.8463**	0.9123	0.7883	0.7838
	15	0.8668	0.8750	0.7098	0.9015	**0.8350**	0.6967
	20	0.8899	0.8751	0.6262	0.8609	0.6919	0.7385
	30	**0.8997**	0.8286	0.6990	0.8362	0.8245	0.6196
	40	0.8265	0.8147	0.7079	0.8729	0.6889	0.7781
	50	0.7526	0.6940	0.7778	0.6841	0.7837	0.8152

pixel, the amount of data obtained will vary. Experimental results show that we obtain the best results when we use GRU and use a data set with an offset of 1. Compared with earlier results, there is a certain improvement in the prediction accuracy of spring back error. When the offset pixels are 3 and 5, although a small part of the data can be enhanced compared with the previous results, the overall effect is not perfect.

4.2 Evaluation Metrics

We compare our model's projected spring back values to those that are already known as part of the evaluation process. In other words, we wanted to find the relationship between the n predicted values$E = \{e_1, e_2, \ldots, e_n\}$, en and the n expected values $\hat{E} = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$, en after applying the proposed spring back prediction model. The Mean Absolute Error (MAE), Mean Square Error

(MSE), Root Mean Square Error (RMSE), and R2 performance metrics were also employed.

The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the data set. It measures the average of the residuals in the data set.

$$e_{mae} = \frac{\sum_{i=1}^{i=n}(|e_i - \hat{e}_i|)}{n} \tag{2}$$

Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$e_{mse} = \frac{\sum_{i=1}^{i=n}(e_i - \hat{e}_i)^2}{n} \tag{3}$$

Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals

$$e_e = \sqrt{P\frac{\sum_{i=1}^{i=n}(e_i - \hat{e}_i)^2}{n}} \tag{4}$$

The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$e_{r2} = 1 - \frac{\sum_{i=1}^{i=n}(e_i - \hat{e}_i)}{\sum_{i=1}^{i=n}(e_1 - \bar{\hat{e}})} \tag{5}$$

4.3 What does the proposed system compare with the previous work?

The following 4 figures shows the results of the method proposed in this paper compared with the results of previous research [14]. The dots represent the results of previous research, and the 'x' represent the results of this study. It can be seen from these 4 figures that when overlapping is used and the step size is 1, it has a very good performance. The best R2, or "Coefficient of Determination" of 0.9228 was found, demonstrating that the proposed methods offered a workable remedy to the current SPIF constraints.

4.4 What is the best step value?

The comparison of the new data set size obtained after using different offset pixel amounts with the original data set size is shown in Table 1. Step 1 represents an offset of 1 pixel, and the same step 3 and step 5 represent an offset of 3 pixels and 5 pixels. It can be seen that the data sets with different grid sizes have been

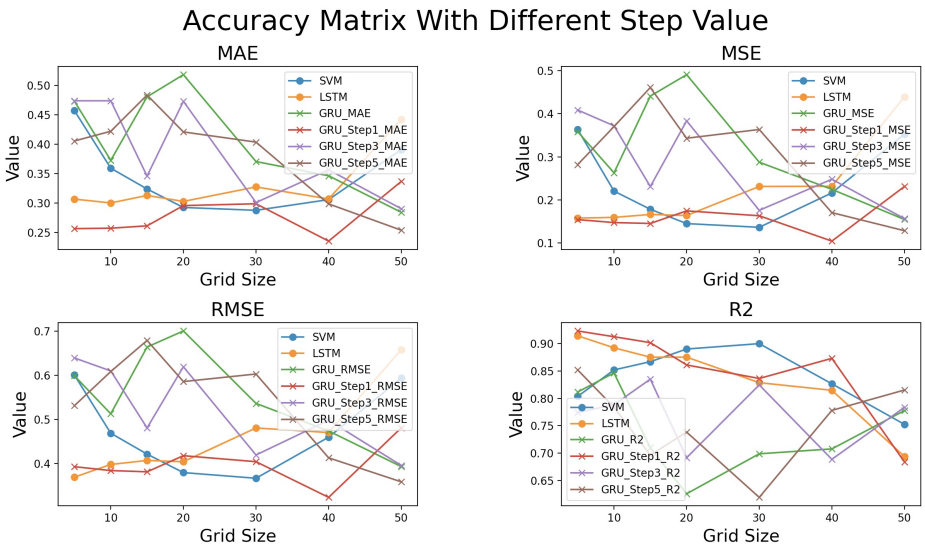

Fig. 4. Compare Result by Accuracy Matrix with Different Step Value

significantly improved. In Table 2, the right three columns use the data set after data augmentation. When the step value is 1, MAE, MSE and RMSE have the best performance at d = 40mm, and when we move away from d = 40mm grid size, in any direction, the performance drops rapidly. When the step value is 3 and 5, MAE, MSE and RMSE have the best performance at d = 50mm. The best performance of R2 is when d = 5mm. Therefore, it can be seen that when the step value is 1, the overall prediction performance is effectively improved.

4.5 What is the best grid size?

It can be seen from the Table 2 that shows the results obtained from the experiments for a range of different values of d. In the study [14], for the LSTM-MLP method, in the SVM regression method, the best results were obtained at low grid size (d less than 10mm) and d = 30mm, while using the original data concentration, GRU obtains the best results at low grid size (d less than 10mm), and when using the data set after data augmentation, the experimental results of all grid sizes have been improved, and when d = 5mm has the best performance. However, there is a trade off between accuracy and model generation time.

5 Conclusion

Based on [14], this paper introduces a data augmentation method for the original data set to solve the shortcomings of insufficient data sets and uses 3 different

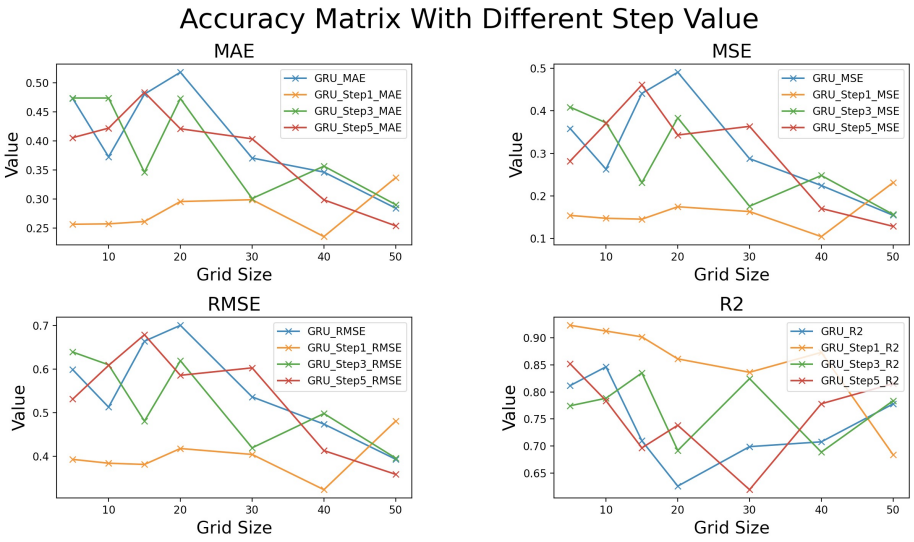

Fig. 5. Accuracy Matrix with Different Step Value

step values to obtain 3 large groups of data sets of different sizes. The use of this method can maximize the advantages of the deep learning model. The approach in [14] is to use a new type of local geometric representation of point series conceptualized based on a grid defined for a given grid size d. The intuition is that local geometries can be combined in many ways to describe any part to be manufactured. The proposed point sequence representation builds on that used in [7][8], but uses actual point values instead of classified point values. The proposed representation is used to generate a spring back prediction model. A simpler mechanism for generating such models was considered: the GRU. The reported evaluation first shows that, for GRU model generation, a batch size of 8 and a step value equal to 1 is the best trade-off between training time and accuracy. The optimum value for d was found to be 40mm. Overall, the work presented in this paper demonstrates the basic idea of representing local geometries as time series, local geometries can be combined according to any shape, and can then have predicted spring back values associated with them, using appropriate deep learning methods to generate or machine learning models that may help make SPIF available in a wider commercial context.

Future work will focus more on the relationship between local and global geometry and consider using graph neural networks or adding attention mechanisms to existing models. In particular, attention should be paid to how to better identify and represent edge corners and their relationship to global geometry. And how to better realize spring back prediction through these relationships, reduce spring back errors and how to better apply these errors to practical applications.

References

1. S. Akrichi, A. Abbassi, S. Abid, and N. Ben Yahia. Roundness and positioning deviation prediction in single point incremental forming using deep learning approaches. *Advances in Mechanical Engineering*, 11(7):1687814019864465, 2019.
2. R. Bahloul, H. Arfa, and H. B. Salah. Application of response surface analysis and genetic algorithm for the optimization of single point incremental forming process. In *Key Engineering Materials*, volume 554, pages 1265–1272. Trans Tech Publ, 2013.
3. K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
4. S. El Salhi. *Identification of correlation between 3D surfaces using data mining techniques: a case study of predicting springback in sheet metal forming*. The University of Liverpool (United Kingdom), 2014.
5. S. El-Salhi, F. Coenen, C. Dixon, and M. S. Khan. Predicting features in complex 3d surfaces using a point series representation: A case study in sheet metal forming. In *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, December 14-16, 2013, Proceedings, Part I 9*, pages 505–516. Springer, 2013.
6. M. S. Khan, F. Coenen, C. Dixon, and S. El-Salhi. Classification based 3-d surface analysis: predicting springback in sheet metal forming. *Journal of Theoretical and Applied Computer Science*, 6(2):45–59, 2012.
7. M. S. Khan, F. Coenen, C. Dixon, S. El-Salhi, M. Penalva, and A. Rivero. An intelligent process model: predicting springback in single point incremental forming. *Journal of Expert Systems With Applications*, 76:79–93, 2014.
8. M. S. Khan, F. Coenen, C. Dixon, S. El-Salhi, M. Penalva, and A. Rivero. An intelligent process model: predicting springback in single point incremental forming. *The International Journal of Advanced Manufacturing Technology*, 76:1–12, 2015.
9. K. Maji and G. Kumar. Inverse analysis and multi-objective optimization of single-point incremental forming of aa5083 aluminum alloy sheet. *Soft Computing*, 24(6):4505–4521, 2020.
10. P. Martins, N. Bay, M. Skjødt, and M. Silva. Theory of single point incremental forming. *CIRP annals*, 57(1):247–252, 2008.
11. D. Möllensiep, P. Kulessa, L. Thyssen, and B. Kuhlenkötter. Regression-based compensation of part inaccuracies in incremental sheet forming at elevated temperatures. *The International Journal of Advanced Manufacturing Technology*, 109:1917–1928, 2020.
12. H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.
13. M. Sbayti, R. Bahloul, and H. Belhadjsalah. Efficiency of optimization algorithms on the adjustment of process parameters for geometric accuracy enhancement of denture plate in single point incremental sheet forming. *Neural Computing and Applications*, 32:8829–8846, 2020.
14. H. Y. M. P. O. M. O. F. C. A. N. Yang Bingqian, Yuanyi Zeng. *Spring Back Prediction Using Point Series and Deep Learning*. 2022.
15. M. Zwierzycki, P. Nicholas, and M. Ramsgaard Thomsen. Localised and learnt applications of machine learning for robotic incremental sheet forming. In *Humanizing digital reality: Design modelling symposium Paris 2017*, pages 373–382. Springer, 2018.