Research Progress in Mechanical and Manufacturing Engineering Vol. 3 No. 1 (2022) 875-883
© Universiti Tun Hussein Onn Malaysia Publisher's Office

RPMME

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/rpmme
e-ISSN : 2773-4765

Application of Machine Learning Algorithm to Prediction of Thermal Spring Back of Hot Press Forming

Mah Wei Chun[1], Ng Chuan Huat[2]*, Ong Pauline[1]

[1]Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/rpmme.2022.03.01.093
Received 01 Dec 2021; Accepted 01 April 2022; Available online 30 July 2022

Abstract: Some steels are very difficult to fabricate using cold forming, which is a conventional sheet metal forming method. As a result, hot stamping is one of the ways utilized to manufacture components made of advanced high strength steel (AHSS). Although hot press sheet can form the high-strength steels, but it also can cause to thermal springback defects. In this paper, thermal spring back simulation data is used to examine and predict the springback condition by using machine learning algorithm. The focus of this research is to determine which machine learning model performs best for thermal springback predictions. To forecast thermal springback, three machine learning techniques were used in this paper: the KNN algorithm, the DT algorithm, and the SVR algorithm. Furthermore, the predicting errors of these three models are compared. The compared results indicate that the Decision Trees model can properly forecast and capture thermal springback variation trends.

Keywords: Hot Press Forming, Machine Learning, Thermal Springback, Supervised

1. Introduction

The Industry 4.0 is an important transformation by synthesize the digital and internet technologies with conventional industry. Increase the flexibility and resource efficiency through digitization is the aim of an industry to develop products faster. Cyber-physical systems monitor actual processes, create a digital replica of the actual environment, and make decentralised decisions in intelligent factories enabled by Industry 4.0. Also, they cooperate in real time with human resources and each other by using Internet of Things (IoT). The algorithms integrated with users monitored the Cyber-physical systems via Internet [1]. The transition to the Fourth Industrial Revolution, sometimes known as "Industry 4.0" is pervasive today and has had a significant impact on the manufacturing industry [2].

Hot stamping forming is a sheet metal forming technique that involves heating a blank before producing it. The young's modulus and yield limit are decreased by the elevated temperature while

**Corresponding author: ahuat@uthm.edu.my*

enhance ductility in comparison to the material's cold condition. The blank is heated to 900 °C during the heating state and held at this degree until it is totally austenite. For the utilised sheet thickness, it took at least 5 minutes at 900 °C. The Figure 1.1 shown that the blank is then moved and formed by a hydraulic press, which employs a die, a blank holder, and a punch. When the forming is completed, the component is hardened using contact pressure from cooler tools and air cool to room temperature [3].

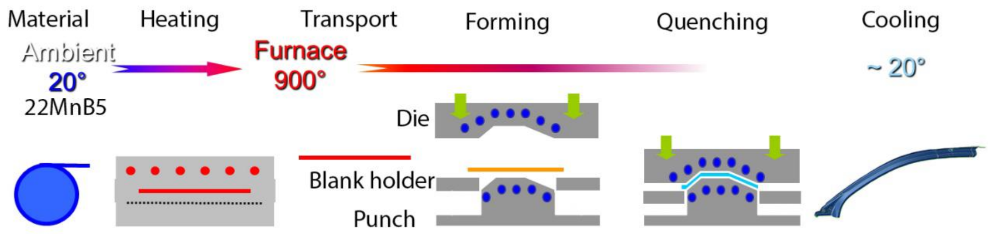

Figure 1.1: Hot stamping process

1.1 Problem Statement

Some steels are extremely hard to construct using cold forming, which is a standard procedure for sheet metal forming. Therefore, hot stamping is one of the methods used for producing parts of advanced high strength steel (AHSS). Although the hot stamping only has very small spring back. However, it can also be seen those non-symmetric sections with specialised tempering, such as A-pillars on an automobile, have a substantially bigger spring back during the cooling stage [3]. Because there is more springback following hot press sheet forming of high-strength steels, the evolution of sheet metal forming technologies is required in the future [4].

1.2 Objectives

1. To investigate the thermal spring back behaviour between the blank and dies under forming conditions.
2. To develop several machine learning model by using design of experiments with the forming parameters for extracts the thermal spring back image.
3. To compare the performance of machine learning models for monitors the thermal spring back in hot press forming process.

2. Materials and Methods

The materials and methods section, often known as methodology, covers all of the information required to produce the study's results.

2.1 Simulation

Deform software is used to create a simulation of hot press forming. The parameters utilised to run the simulation include preheating temperature, blank thickness, punch and die temperature, cooling water temperature, punch velocity, displacement between punch and blank, forming time, die holding time, and punch lifting time.

2.2 Data Preparation

This section will be focusing on handling the data quality issues, data filtration, feature extraction and etc, before they are ready to be feed into the machine. This is to make the quality of the data as high as possible and minimize the errors or inaccuracies that might occur later. The dataset after EDA is shown in **Figure 2.1**.

Blank thinkness (mm)	Ideal blank height (mm)	Temperature (°C)	Actual blank height (mm)	etween the ideal and actual blank	Springback condition
1	11	25	11.8954	0.8954	Serious SpringBack
1	11	50	11.8255	0.8255	Serious SpringBack
1	11	100	11.7292	0.7292	SpringBack
1	11	150	11.6088	0.6088	SpringBack
1	11	200	11.6981	0.6981	SpringBack
1	11	250	10.6254	-0.3746	Slightly SpringBack
1	11	300	11.3743	0.3743	Slightly SpringBack
1	11	350	10.8732	-0.1268	Normal
1	11	400	11.2666	0.2666	Slightly SpringBack
1	11	450	10.858	-0.142	Normal
1	11	500	11.2646	0.2646	Slightly SpringBack
1	11	550	11.1402	0.1402	Normal
2	12	25	13.0316	1.0316	Serious SpringBack
2	12	50	12.4721	0.4721	Slightly SpringBack
2	12	100	12.1846	0.1846	Normal
2	12	150	11.9096	-0.0904	Normal
2	12	200	12.1679	0.1679	Normal
2	12	250	11.9565	-0.0435	Normal
2	12	300	12.1246	0.1246	Normal
2	12	350	11.9383	-0.0617	Normal
2	12	400	12.1067	0.1067	Normal
2	12	450	11.9647	-0.0353	Normal
2	12	500	12.0948	0.0948	Normal
2	12	550	12.0861	0.0861	Normal
3	13	25	13.3344	0.3344	Slightly SpringBack
3	13	50	13.2569	0.2569	Slightly SpringBack
3	13	100	13.1868	0.1868	Normal
3	13	150	12.9568	-0.0432	Normal
3	13	200	13.0971	0.0971	Normal
3	13	250	12.9698	-0.0302	Normal
3	13	300	13.0721	0.0721	Normal
3	13	350	12.8742	-0.1258	Normal
3	13	400	13.0624	0.0624	Normal
3	13	450	13.052	0.052	Normal
3	13	500	13.0315	0.0315	Normal
3	13	550	13.0083	0.0083	Normal

Figure 2.1: Dataset after EDA

2.3 Machine Learning Models

In addition to the project implementation, the technique used for the three selected machine learning models will be described.

2.3.1 Model 1 (K-Nearest Neighbors)

The KNN algorithm presumes that comparable objects exist nearby. In other words, comparable objects are close together. KNN algorithm captures the concept of similarity (also known as distance, proximity, or closeness) with certain mathematics we may have learned in primary school, such as calculating the distance between points on a graph. There are various methods for determining distance, and depending on the task at hand, one method may be preferred. However, the straight-line distance (also known as the Euclidean distance) is a famous and well-known option.

2.3.2 Model 2 (Decision Trees)

Decision Tree is a Supervised learning approach that is being used to solve both classification and regression issues, however it is most commonly used to solve classification problems. Decision Trees categories examples by sorting them along the tree from the root to some leaf node, with the leaf node providing the example's categorization. This algorithm checks the values of the root property with the values of the record (actual dataset) attribute and then follows the branch and jumps to the next node depending on the comparison. After that, the algorithm will repeat the comparison of the attribute value and other sub-node and move further. This step will continue until the algorithm meets the leaf node (results).

2.3.3 Model 3 (Support Vector Regression)

The SVR operates on the same basis as the SVM. The aim of SVR is to locate the best fit line among all the provided data, or more specifically, the hyperplane with the greatest number of data points. Unlike other regression models that seek to minimize the difference between the real and predicted values, SVR simply tries to fit the best line within a threshold value, which is effectively the distance between the hyperplane and the boundary line. The complexity of SVR's fit time rises more than quadratically with the number of samples, making it difficult to scale to datasets with more than a few tens of thousands of samples.

3. Results and Discussion

3.1 K-Nearest Nrighbors model

MATLAB built-in function (fitcknn) is utilized. The optional parameters of command (fitcknn) "NumNeighbors" are then set to 3, and the variable 'cond' (condition of springback) in dataset is set as the output variable. Figure 3.1 shows the detail information of K-NN model. The accuracy of the KNN model is 90%, training error is 0.26923, and testing error is 0.096154 as shown in Figure 3.2.

```
Mdl =

  ClassificationKNN
           PredictorNames: {1×5 cell}
             ResponseName: 'cond'
    CategoricalPredictors: []
               ClassNames: {1×4 cell}
           ScoreTransform: 'none'
          NumObservations: 26
                 Distance: 'euclidean'
             NumNeighbors: 3


  Properties, Methods
```

Figure 3.1: Detail information of KNN model

```
accuracy =

    90

Training Error: 0.26923
Test Error: 0.096154
```

Figure 3.2: Accuracy, Training error and Testing error of KNN model

3.2 Decision Trees model

The decision tree model is created in MATLAB using the built-in command 'fitctree'. We don't use any parameters in this model, simply input and output variables to train it. The coding and detail information are shown in **Figure 3.3**. As shown in Figure 3.4, the DT model's accuracy is 90%, its training error is 0.076923, and its testing error is 0.086957. For Decision Trees algorithm, we can view the Decision Tree diagram as shown **Figure 3.6** in by using the coding in Figure 3.5. From the Tree Diagram, we can see that how the Decision Trees algorithm make decision and predict the outputs.

```
Mdl =

  ClassificationTree
           PredictorNames: {1×5 cell}
             ResponseName: 'cond'
    CategoricalPredictors: []
               ClassNames: {1×4 cell}
           ScoreTransform: 'none'
          NumObservations: 26


  Properties, Methods
```

Figure 3.3: Detail information of Decision Trees model

```
accuracy =

    90

Training Error: 0.076923
Test Error: 0.086957
```

Figure 3.4: Accuracy, Training error and Testing error of Decision Trees model

```
view(Mdl,'Mode','graph')
```

Figure 3.5: Coding to show the Decision Tree diagram in MATLAB

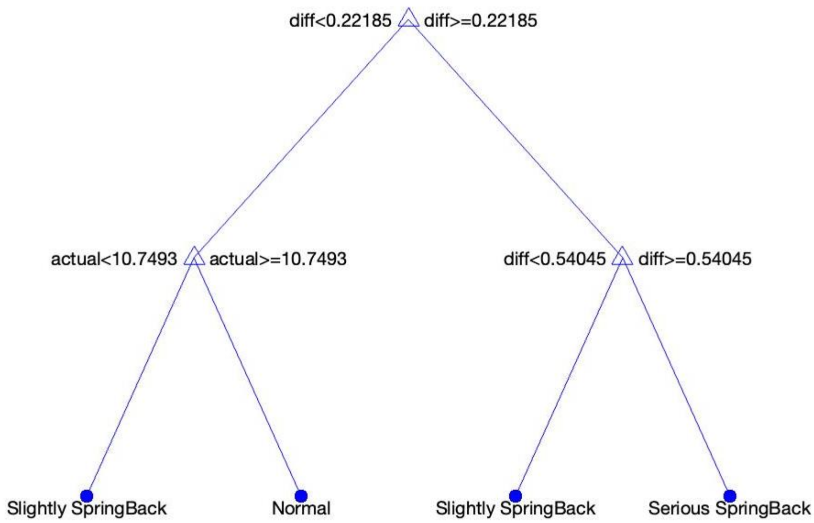

Figure 3.6: Tree diagram to predict the springback condition

3.3 Support Vector Regression model

In MATLAB, the SVR model is developed using the built-in programme 'fitcecoc.' The parameters of the SVR algorithm cannot modify directly. To do so, we must first construct a template with the updated parameters. Then, using the built-in parameters 'Learners,' assign the template just created as shown in Figure 3.7. The detail information of SVR model is shown in Figure 3.8. The SVR model's accuracy is 80%, its training error is 0, and its testing error is 0.21053 as shown in Figure 3.9.

```
template = templateSVM("KernelFunction","gaussian");
Mdl = fitcecoc(dataTrain,'cond',"Learners",template)
```

Figure 3.7: Coding to develop SVR model

```
Mdl =

  ClassificationECOC
           PredictorNames: {1×5 cell}
             ResponseName: 'cond'
    CategoricalPredictors: []
               ClassNames: {1×4 cell}
           ScoreTransform: 'none'
           BinaryLearners: {6×1 cell}
               CodingName: 'onevsone'


  Properties, Methods
```

Figure 3.8: Detail information of SVR model

```
accuracy =

    80

Training Error: 0
Test Error: 0.21053
```

Figure 3.9: Accuracy, Training error and Testing error of SVR model

3.4 Comparison between models

After finishing all three machine learning models, they were compared to see how accurate they were. Based on the results of the accuracy tests, one of the models must be chosen as the best match with the most dependable projected responses. The evaluation matrix for all three machine learning models is shown in Table 3.1.

Table 3.1: Evaluation matrix of all machine learning models

Name of Model	Accuracy (%)	Training Error	Testing Error	Rating

KNN	90	0.26923	0.09615	2
DT	90	0.07692	0.08696	1
SVR	80	0	0.21053	3

By observing Table 3.1, Decision Trees model (DT) was discovered to have the least level of Training and Testing error, followed by K-Nearest Neighbors model (KNN) to the Support Vector Regression model (SVR). As a consequence, the Decision Trees algorithm is regarded as the best model among all since it can produce the best predicted results that are the closest to the actual results.

4. Conclusion

This research aims at the use of machine learning in predicting springback in hot sheet metal forming. According to the research, Decision Trees might be a viable alternative to the K-NN and SVR algorithms for predicting springback errors. In the experiment, the Decision Trees technique predicted springback substantially better than the SVR algorithm and somewhat better than the KNN algorithm. The superior performance of Decision Trees over KNN and SVR can be due to the following factors:

1. When compared to the other two algorithms, the Decision Trees approach can be trained effectively even with a small amount of dataset. For Decision Trees algorithm, Once the variables have been created, there is less data cleansing necessary. So that, missing values and outliers have less impact on the data in the decision tree.

2. When compared to the K-NN and SVR algorithms, the Decision Trees algorithm has less parameters and even doesn't require them. The accuracy of the Decision Trees model is substantially better than the SVR model, as shown in the research, despite the fact that the SVR model contains more modified parameters.

3. The use of a training dataset to proceed with KNN and SVR training necessitates a high level of knowledge and care. As observed in the results, even though the SVR model's training error is zero, the predicted outcome is fully 'Normal,' showing that it is over-fitting to the training dataset.

We may deduce from the three factors mentioned above that incorrect parameter selection might result in either over-fitting or under-fitting of the training phase. Because the Decision Trees method has better ability of prediction of springback defects over the KNN and SVR algorithms, future research will look at improving the Decision Trees algorithm to obtain better generalization performance.

Acknowledgement

The authors would like to thank the Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia for the support in accomplishing this research.

References

[1] Stăncioiu, A., *The fourth industrial revolution „Industry 4.0”.* Fiabilitate Şi Durabilitate, 2017. **1**(19): p. 74-78.

[2] Ong, P., W.K. Lee, and R.J.H. Lau, *Tool condition monitoring in CNC end milling using wavelet neural network based on machine vision.* The International Journal of Advanced Manufacturing Technology, 2019. **104**(1): p. 1369-1379.

[3] Lugnberg, M. and T. Netz, *Investigation of thermal spring back of a hot formed 22MnB5 A-pillar with tailored properties*. 2016.

[4] Yanagimoto, J., K. Oyamada, and T. Nakagawa, *Springback of High-Strength Steel after Hot and Warm Sheet Formings.* CIRP Annals, 2005. **54**(1): p. 213-216.