Machine Learning for the Prediction of Brake Bending Parameters

some hyped-up tagline

Philipp Kurrle

Supervised by Prof. Dr. Ulrike Pado

Co-supervised by Peter Lange

Computer Science

Software Technology Master

Hochschule für Technik

December, 2022

A dissertation submitted in partial fulfilment of the requirements for the degree of M.Sc. in Software Technology.

Abstract

This is the abstract. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

List of Figures

List of Tables

List of Abbreviations

1 Introduction

Sheet metal forming has been used for centuries in different manufacturing industries to create a wide range of products for different applications. Sheet metal bending and stamping can be considered as the most important variants in the forming industry. (Cruz et al., 2021, p. 1) Therefore these have been continuously improved in recent decades to meet the growing demand especially in automotive and aircraft industries with the goal to reduce energy efficiency and emissions. (Zheng et al., p. 4)

Sprinback is a common phenomenon in sheet metal forming processes. It is a deformation of the sheet metal that occurs when the sheet metal is bent. Therefore, predicting the spring back is important to reduce the number of trial and error cycles in the manufacturing process. (Cruz et al., 2021, p. 1) Sheet metal forming is a complex process that involves a large number of variables and parameters Therefore, it is difficult to predict the spring back accurately, which makes it an interesting case for machine learning.

In order to predict springback with minimium errors, this thesis build and evaluates different machine learning models to predict the springback of a sheet metal. The models are evaluated based on the mean absolute error (MAE) and the root mean squared error (RMSE). The best model is then used to predict the springback of a sheet metal with different parameters.

2

Theoretical Foundations

2.1 | Sheet Metal Bending

Sheet metal bending is a forming operation that is commonly used to produce high-volume and low-cost component in different industries. Forces are applied on the sheet metal to change it's geometry to manufacture different shapes. (Dib et al., p. 1) Sheet metal bending is considered complex because only the final state of the sheet metal is known, so there are many possible bending sequences which could get applied. Furthermore, the process is highly nonlinear because large deformations are applied to the sheet metal with the consequence of plastic behavior. Conventional processes are often based on empirical trial and error approaches. (Dib et al., p. 1) A common approach is to experimentally create so named 'technology tables' which contain the bending parameters and the resulting spring back. (Quelle Hochstrate?) This process is time and cost intensive and therefore often not suitable for the production of high-volume and low-cost components.

2.1.0.1 | Sheet Metal

Sheet metal is any form of that has a relatively large length to thickness ratio, their thicknesses are typically from 0.4mm to 6mm. (Baig et al., 2021) Parts made of sheet metals are manufactured with different methods, such as stamping, bending, shaping etc. Bending is one of the most common methods to manufacture sheet metal parts.

2.1.0.2 | Bending

Bending is a forming operation that is used to change the shape of a sheet metal. The applied load is beyond its yield strength but below its ultimate tensile strength to achieve a permanent deformation (Baig et al., 2021, p. 1) During bending the metal on the outside of the neural plane is stretched and the metal on the inside of the neutral plane is compressed. The neutral plane is the plane that is perpendicular to the bending axis. (Baig et al., 2021, p. 3)

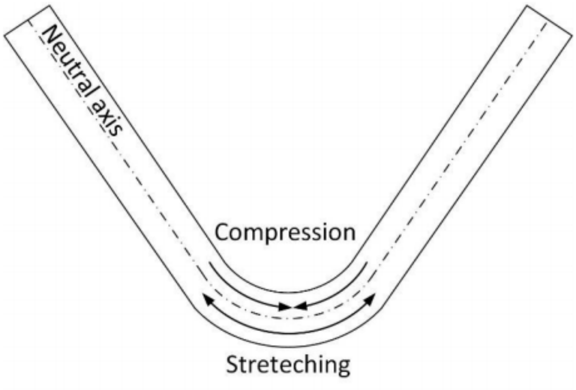

Figure 2.1: Bending plane, compression and stretching of sheet metal (Baig et al., 2021, p. 3)

2.1.0.3 | Air Bending

Air bending is a variant in the V-Bending process which is performed using puch-and-die tooling. (Groover, p. 416) It is commonly used in automotive industry to manufacture sheet metal parts. Kim et al. In this process the punch sheet metal comes in contact of the outside edges of the die, as well as the punch tip, but it does not come in contact with the die surface. It is typically the preferred bending method because, its high flexibility because it is possible to achieve different bending angles using the same punch-and-die tooling. (Miranda et al., p. 3)(Cruz et al., 2021, p. 1)

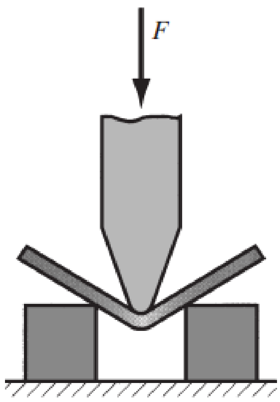

Figure 2.2: Air bending (Groover, p. 416)

Today press brake bending machines like air bending machines are usually equipped with "copmputer numeral control" (CNC) systems that can automatically control the bending process and produce the desired shape." (Miranda et al., p. 3) The air bending process is shows strong nonlinear behavior, considering its parameters and their interrelationships. (Miranda et al., p. 3)

2.1.0.4 | Spring Back

When the punch and therefore the bending pressure is removed at the end of the deformation operation, elastic energy remains in the bent part. This elastic energy is released and the metal sheet partially returns to its original shape. (Groover, p. 113-114)

To address this issue there are several methods to compensate the spring back. For example one common method is over bending, which means that the punch angle and radius are fabricated smaller than the specified angle. (Groover, p. 114) Prerequisite for all compensation methods is that the springback is known therefore the accurate prediction of the springback play an important role in the manufacturing process.

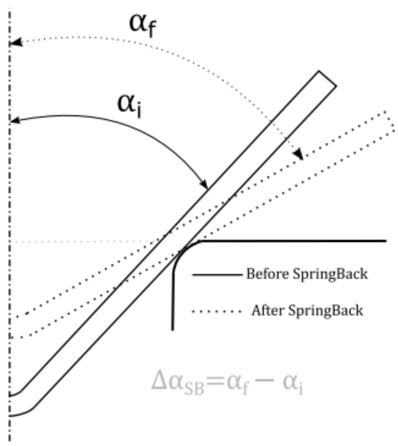

Figure 2.3: Spring back (Cruz et al., 2021, p.)

2.2 | Machine Learning

2.2.1 | Supervised Learning

"Remember that supervised learning is used whenever we want to predict a certain outcome from a given input, and we have examples of input/output pairs. We build a machine learning model from these input/output pairs, which comprise our training set.

Our goal is to make accurate predictions for new, never-before-seen data. Supervised learning often requires human effort to build the training set, but afterward automates and often speeds up an otherwise laborious or infeasible task." (Müller and Guido, p. 34)

2.2.2 | Overfitting and Underfitting

"In supervised learning, we want to build a model on the training data and then be able to make accurate predictions on new, unseen data that has the same characteristics as the training set that we used. If a model is able to make accurate predictions on unseen data, we say it is able to generalize from the training set to the test set. We want to build a model that is able to generalize as accurately as possible. " (Müller and Guido, p. 35)

"Usually we build a model in such a way that it can make accurate predictions on the training set. If the training and test sets have enough in common, we expect the model to also be accurate on the test set. However, there are some cases where this can go wrong. For example, if we allow ourselves to build very complex models, we can always be as accurate as we like on the training set." (Müller and Guido, p. 35)

"The only measure of whether an algorithm will perform well on new data is the evaluation on the test set. However, intuitively3 we expect simple models to generalize better to new data. If the rule was “People older than 50 want to buy a boat,” and this would explain the behavior of all the customers, we would trust it more than the rule involving children and marital status in addition to age. Therefore, we always want to find the simplest model. Building a model that is too complex for the amount of information we have, as our novice data scientist did, is called overfitting. Overfitting occurs when you fit a model too closely to the particularities of the training set and obtain a model that works well on the training set but is not able to generalize to new data. On the other hand, if your model is too simple—say, “Everybody who owns a house buys a boat”—then you might not be able to capture all the aspects of and variability in the data, and your model will do badly even on the training set. Choosing too simple a model is called underfitting." (Müller and Guido, p. 35)

2.2.3 | Ensemble Learning

Ensemble techniques in machine learning to involve the construction of multiple models, called "learners," for a given task. The primary goal of these methods is to improve the accuracy and performance of the model by combining the predictions of multiple learners. Ensemble techniques differ from single classifier methods by constructing multiple models and combining them using a voting strategy in order to highlight different aspects of the data. This can potentially lead to improved overall performance. (Shaik and Srinivasan, p. 253)

2.3 | State of research

With the availability of data there has been an increased use of Machine Learning Machine Learning (ML) in sheet metal forming with the goal to reduce costs and increase manufacturing quality. Bock et al. (Cao et al.) The ML algorithms can be divided into the main categories supervised learning, unsupervised learning and reinforcement learning. (Liu et al., a) Supervised learning is generally used in classification problems and regression problems while unsupervised learning is used to find patterns in data (Cruz et al., 2021, p. 2).

Spring Back Prediction Using Unsupervised Learning

Artificial Neural Networks Artificial Neural Network (ANN)s are widely used in sheet metal forming because of their high accuracy and generalization performance. (Cruz et al., 2021, p. 2) (Narayanasamy and Padmanabhan) which compared regression and neural network modeling for predicting spring back of steel sheet metal during the air bending process. They observed that ANN was able to predict the spring back with higher accuracy. But they had a sample size of 25 and suggested further research. (Inamdar et al.) developed an ANN for the air bending process to predict spring back as well as the punch travel to achieve the desired angle in a single stroke. (Kazan et al.) developed an ANN trained with FEM simulation data to predict the spring back for the wipe-bending process.

Because ANNs need a large amount of data to train the model generating the data with real machines is a time-consuming process. Therefore, it is common to use ANNs trained with Finite Element Method Finite Element Method (FEM) simulation data. *Was sind die Nachteile von FEM? Warum nutze eich "echte" Experimente?*

Spring Back Prediction Using Supervised Learning

Liu et al. (2019) used a Support Vector Machine Support Vector Machine (SVM) to predict the spring back of micro W-Bending operations. Liu et al. (b) Dib et al. (2019) compared different ML techniques (logistic regression, SVM, KNN, ANN, random forest, decision tree, naive Bayes, MSP) to predict the spring back and the occurrence of defects in sheet metal. (Dib et al., p. 1) The authors conclude that the MLP and the SVM are the best performing algorithms and suggest further studies of ML regressions models and kriging regression models. (Dib et al., p. 13)

3

Research methodology

The research method used in this thesis follows the design science research (DSR) approach (von Rennenkampff et al., 2015, p. 17). DSR is a research paradigm in which the designer tries to create artifacts to answer questions for problems.

DSR is a research paradigm in which the designer creates artifacts and uses them to answer questions for problems and generate new scientific knowledge. The designed artifacts are both useful and fundamental to understanding the problem (Hevner and Chatterjee, 2010, p. 10)

Design, according to Peffers et al. (2007), is the creation of an applicable solution to a problem (Peffers et al., 2007, p.47)

According to Hevner et al. (2010) design" is both a process ("a set of activities") and a product ("artifactt"). (Hevner and Chatterjee, 2010, p.78) The design-oriented research approach as a methodological framework seems well suited to answer the research questions. Predicting spring back and bend deduction is a relevant problem in business practice. Also, the conception and implementation of machine learning models is a design activity.

The term artifacts is intentionally broad and can take on different forms. In this work, the artifact is different machine leaning models which are applied on the generated data. DSR can be implemented in various ways, a prominent example is provided by Peffers et al. and shown in Figure 3.1 The approach comprises six steps, which are dived into the superordinate phases "Build" and "Evaluate". This thesis follows these phases.

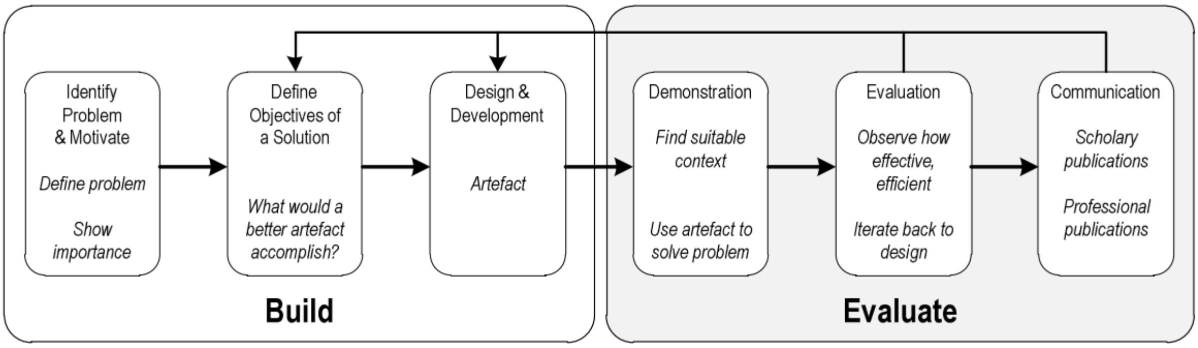

Figure 3.1: Design Science Research Approach according to Peffers et al.
Picture: (Sonnenberg and vom Brocke, 2012, p. 72)

Activity 1 - Problem identification and motivation This activity includes defining a specific research problem and the value of a potential solution. The problem is used for the development of the artifact. To reduce complexity, the problem should be divided into sub-problems. For problem-solving, explicit methods such as system requirements gathering or an implicit method such as programming and/or data analysis. (Peffers et al., 2007, p. 52)

Activity 2: Define the objectives for a solution The goals of a solution are derived from the problem definition. These are derived in the context of what is possible and feasible. Objectives can be quantitative or qualitative. Objectives should be derived from the problem specification and are thus based on the previous step. For knowledge about previous solutions and their effectiveness are required (Peffers et al., 2007, p. 55)

Activity 3: Design and development This step involves the creation of the artifact. An artifact can potentially contain models, methods or constructs, it can be anything that contributes to the solution of the research question. This step includes the definition of the functionality and architecture of the artifacts, followed by the creation of them. (Peffers et al., 2007, p. 55)

Activity 4: Demonstration The use of the previously created artifactt is demonstrated for one or more problems. This requires effective knowledge of the artifact. (Peffers et al., 2007, p. 55)

Activity 5 - Evaluation It is observed and evaluated how well the developed artifact provides a solution to the defined problems in activity 1. Knowledge of relevant metrics and methods of analysis is assumed. Depending on the nature of the problem, the evaluation can take different forms. A comparison of the functionality of the artifact and other solutions can be considered. Furthermore, quantified quantified parameters can be used to measure the performance of the artifacts (Peffers et al., 2007, p. 56) Hevner et al. suggest five different evaluation methods: Observational methods, analytical methods, experiments, testing of the artifact and descriptive methods (Hevner et al., 2004, p. 87)

Activity 6 - Communication The problem and the artifactt and its benefit are communicated externally (Peffers et al., 2007, p. 56) Hevner et al. describe in a conceptual framework guidelines for the

3.1 | Design Principles

Design Principles (DP) are seen as a central part of design-oriented research. (Gregor et al., 2013, p. 348) Design principles are characterized as "principles of form and function" as well as "principles of implementation" of an artifact. (Gregor and Jones, 2017, p.8) They are used to close the gap between researchers and user and allow prescriptive research on systems. They are used to capture knowledge about the artifact. (Sein et al., 2011, pp. 37-56). Koppenhagen et al. suggest generating design principles by grouping requirements for the solution and then creating core requirements, which can which can then be DPs. (Koppenhagen et al., 2012, p. 6)

3.2 | Evaluation of Machine Learning Models

In the field of software engineering there are already standards that define the quality of software systems and its components like ISO/IEC 25010. (Siebert) noted, that such standards can not applied directly to ML and therefore need to be adapted. Therefore they re-interpreted and extended these existing quality models to the ML context. (Siebert, p. 1) In order to define the Design Principles for the artifacts in this work, the considerations of (Siebert) are used. This enables a systematic process in order to assess the quality of the developed artifacts.

3.2.1 | Goal Question Metric Approach

To make the defined quality attributes measurable, the “Goal-Question-Metric”-approach GQM was chosen in this work. It is one of the most common approaches in DSR and is divided into three levels. (Basili et al., 2002, p. 3)

1. Conceptual level (goal): "A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment." (Basili et al., 2002, p. 3)

2. operational level (question): "A set of questions is used to characterize the way the assessment/achievement of a specific goal is going to be performed based on some characterizing model. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint." (Basili et al., 2002, p. 3)

3. quantitative level (metric): "A set of data is associated with every question in order to answer it in a quantitative way." (Basili et al., 2002, p. 3)

Objectives, questions and metrics can be presented in a hierarchical structure.

Figure 1

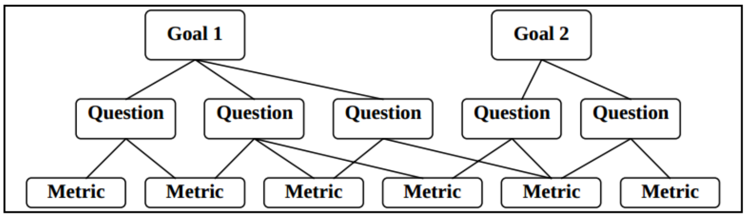

Figure 3.2: Goal-Question-Metric Ansatz (Basili et al., 2002, p. 3)

4

Build

4.1 | Problem Statement

"There are two major types of supervised machine learning problems, called classification and regression." (**?**, p. 34)

"For regression tasks, the goal is to predict a continuous number, or a floating-point number in programming terms (or real number in mathematical terms). Predicting a person's annual income from their education, their age, and where they live is an example of a regression task. When predicting income, the predicted value is an amount, and can be any number in a given range. Another example of a regression task is predicting the yield of a corn farm given attributes such as previous yields, weather, and number of employees working on the farm. The yield again can be an arbitrary number." (Müller and Guido, p. 34)

4.2 | tbd

The following design principles is a selection of Siebert's quality parameters for ML models. In this Design Science Research (DSR) work the artifacts are ML models therefore these design principles are used to evaluate them.

Design Principle 1: Correctness

Does the artifact predict the spring back of a sheet metal with a high accuracy and correctness? With progression in manufacturing there is a growing demand for high-quality products, that means that the meta parts needs to be produced with high precision and accuracy. Here the sprin back is an undesired side effect which need to be mimimized. (Baig et al., 2021, p. 1) Sheet metal forming in manufacturing need a high level of quality and precision. Therefore, the spring back of a sheet metal is an important parameter to consider. (Cruz et al., 2021, p. 1) Predicting spring back is important to reduce the number of trial and error cycles in the manufacturing process. Also predicting spring back is complex because of many variables and parameters and often not all of them are known. Therefore, a machine learning model should predict the spring back of a sheet metal with a high accuracy and correctness. When using the ML model small errors in the prediction can cause fitting problems in the manufacturing process.

Design Principle 2: Appropriateness

Is the artifact appropriate for the given problem? While selecting a model it is important that it fits the problem/task and can deal with the given data. (Siebert, p. 16)

Design Principle 3: Relevance

Does the artifact achieve a good bias-variance trade-off?

In addition to measure the correctness it is important to understand "why" the learner has this performance. This is important to understand the limitations of the model and to improve it. Therefore, it is important to understand the bias-variance trade-off. (Zhou, p. 50) Bias measures the differences between the learnesrs expected prediction and the ground-truth label. This results in the fitting ability of the learner. Variances measures the change of learning performance of the learner because of changes in the training set. This results in the impact of data disturbance on the results. (Zhou, p. 51)

Design Principle 4: Robustness

How well does the artifact handle outliers, noise and missing data? Using real-world data noice is a common problem and can have a negative impact on the performance of the learner. Therefore, it is important to measure how good the artifact performs when dealing with impact data. Sáez et al. proposed a new measure to establish the expected behavior of a learner with noisy data trying to minimize the problems: the Equalized Loss of Accuracy (ELA). (Sáez et al., p. 3)

Design Principle 5: Stability

Does the artifact generate repeatable results when trained on different datasets?

Design Principle 6: Interpretability

Is the artifact easy to understand and explain?

It should be noticed, that there are many parameters and variables involved in the sheet metal forming process. That makes the process design quite complex, particularly in the production of components which require several stages, and thus more than one set of tools. (Dib et al., p. 1) A model which allows conclusions how the results where generated is better.

Design Principle 7: Resource utilization

How many resources does the artifact need to train and predict?

Conventional processes are often based on empirical trial and error approaches. (Dib et al., p. 1) A common approach is to experimentally create so named 'technology tables' which contain the bending parameters and the resulting spring back. (Quelle: Hochstrate?) This process is time and cost intensive and therefore often not suitable for the production of high-volume and low-cost components. Therefore, one of the benefits of using machine learning should be the reduction of the number of trial and error cycles in the manufacturing process. Furthermore, training the model should take not too much time and resources. As mentioned before often FEM-simulation are used to virtually try out metal forming processes. However, fully exploring the design space is computationally expensive and often not possible. (Dib et al., p. 3) The number of experiments can be reduced using a meta-model like ANN. (Dib et al., p. 3) A approach fully based on ML should perfor

4.3 | Dataset generation

For the dataset generation, bending experiments were performed on metal sheets with different thicknesses. The material used is cold rolled steel sheets of the norm DIN EN 10130. The thicknesses used were 0.5mm, 1mmm and 2mm. The material was used because it is commonly used in bending processes and its high availability. In previous tests, it was observed, that the spring back are well observable with this material. Using this material, 200 single bending pieces of the dimension 20×100 mm have been cut. Each piece was bend one time using a *Zwick* three-point-bending machine.

Python script where developed to covert the output data format from the machine to CSV files. The following describes the experimental setup used for the experiments performed.

4.3.1 | Preliminary Tests

A number of preliminary tests were conducted to determine the influence of the punch penetration on the spring back.

4.3.1.1 | Multiple Cycles

One approach was to test if multiple spring back can be measure using only one sheet. Therefore, the machine was programmed to perform multiple cycles in one attempt and bend the metal sheet multiple times. The benefit of this approach would have been a faster generation of the dataset because spring backs could be measured in on attempt, also less material would have been used.

Figure 4.1 shows one of these attempts. The metal sheet was bent 4 times using y_p values from 5 to 8. The results show, that 4 different spring backs can be measured, but the spring back does not vary like expected. It was observed as well, that the spring backs are different in every attempt, this is shown in Figure 4.2. Bending 4 different metal sheets each only one time returned very different results. A possible explanation could be the cold deformation of the steel, which is not reversible. Because this approach did not work, the machine was programmed to perform one cycle at a time.

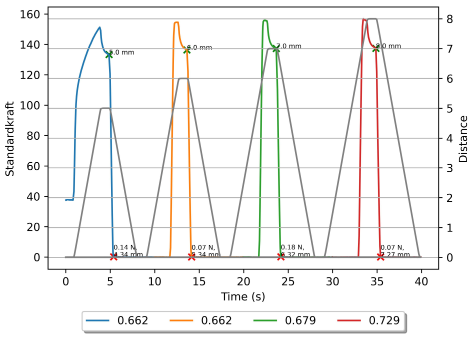

Figure 4.1: Experiment: Bending one metal sheet multiple times with different y_p values.

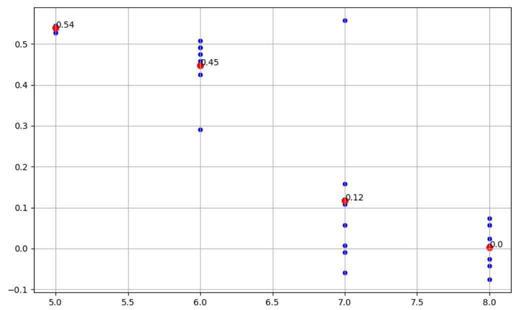

Figure 4.2: Inconsisten results bending one metal sheet mutliple times. The spread of the results is very large.

4.3.1.2 | Brake Bending Machine

Before using the three point bending machine, a brake bending machine was used to test the influence of the bending on the spring back. The brake bending machine is a machine used to bend metal sheets. It is a very common machine in the industry and is used to bend metal sheets to a specific radius. The brake bending machine used is a *Bendmaster 1000* from *Bendmaster*.

After a series of bends it was observed, that the spring back values where much higher than expected. The explanation for that behavior was, that altering the position the bending beam of that specific machine was not enough to get the desired angle.

Thus, the machine excluded for the generation of the data and the three point bending machine was used instead.

Despite the inaccurate data, it was later observed, that the distribution of the spring backs was very similar to the later experiments with the three point bending machine.

4.3.2 | Experimental setup

The setup consists of a three-point-bending machine with a punch and a die with no bottom. The machine used is the *Zwick MX 25A* material testing machine. The machine is equipped with a load cell and a displacement sensor. The load cell is used to measure the force applied to the sheet and the displacement sensor is used to measure the displacement of the punch. The machine is controlled by a computer and a software called *ZwickRoell TestXpert*. The software is used to control the machine and to save the output data.

The experimental setup and the process parameters are shown in Figure 4.3 where V is the die opening, y_p is the punch penetration which is the distance the punch is moved into the sheet. The paramter t is the sheet thickness, α is the sheet corresponding bending angle. Parameter r_p is the punch radius which is the radius of the tip of the punch and r_m is the die radius.

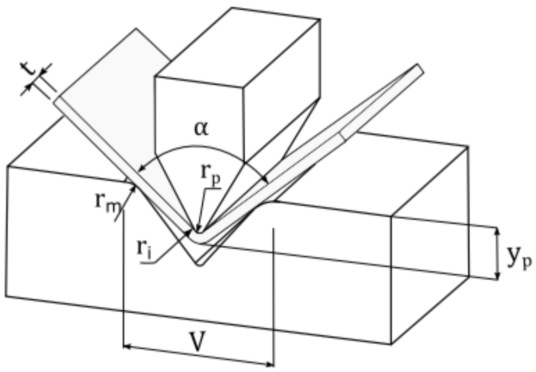

Figure 4.3: Process parameters: Sheet bending angle (α), sheet thickness (t), punch penetration (y_p), die opening (V), punch radius (r_p), die radius (r_m), inside bending radius (r_i).

In order to get consistent results, a number of constant and variable parameters were chosen. The parameters include the punch-and-die tooling made of steel where die punch had a radius (r_p) of 5 mm and and die radius (r_m) of 10 mm. The die opening V

was varied between 10 and 50 mm and the punch penetration y_p was varied between 0 and 20 mm. The machine was configured to move the punch with a constant speed of 100 mm/min until it measured a resistance of 1 N. That meant, that the punch reached the metal plate and the actual bending process can start. After a hold time of 1 second the punch was moved with a slower speed of 8 mm/min until the specified punch penetration was reached. The length and width of the metal sheet was 100 mm and 20 mm respectively. The sheet thickness was varied between 0.5 and 3 mm. The constant parameters are shown in Table 4.2 and the varying parameters are shown in Table 4.1.

Parameter		**Value**	**Unit**
Punch penetration	y_p	2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20	mm
Die opening	V	10, 20, 30, 40, 50	mm
Thickness	t	0,5, 1, 1.5, 2, 2.5, 3	mm

Table 4.1: Experimental setup varying parameters

Parameter	**Value**	**Unit**
Punch radius	5	mm
Die radius	5	mm
Sheet thickness	0.5, 1, 2	mm
Sheet width	20	mm
Sheet length	100	mm
Punch speed	100	mm/min
Punch speed after penetration	8	mm/min
Punch force	1	N

Table 4.2: Experimental setup constant parameters

4.3.3 | Measuring The Spring Back

The output data contained different data points, which were used to calculate the spring back. Important parameters for the calculation are the force, punch penetration and testing time. As shown in Figure 4.4 at the yp maximum the punch penetration and the force are maximized as well. The punch stays at that position for 1 second and then moves back with a slower speed. This hold time a limitation of the machine and can not be changed. After the punch is moved back, the force is reduced and the punch penetration is reduced as well, until the punch is at the initial position. For a short time after the lift, the load cell still measures a force. That is because the metal sheet springs back and the punch is still in contact with the sheet. This was measured using a python script, the green and the yellow point represent the resulting spring back distance.

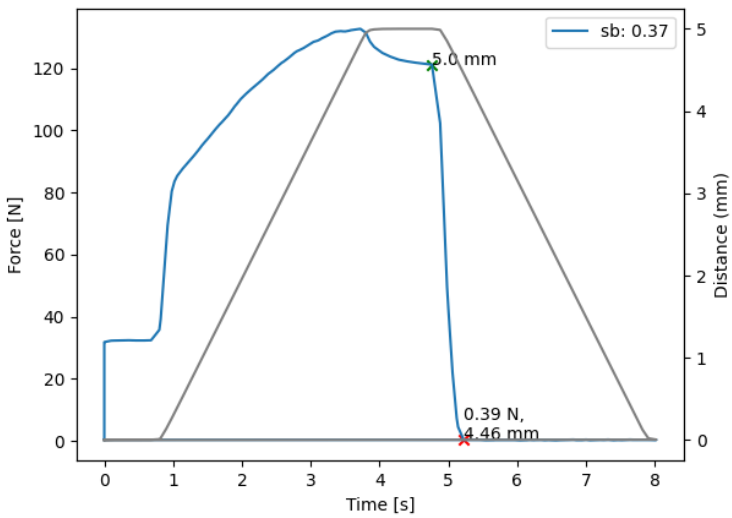

Figure 4.4: A steel metal sheet was bent with a punch penetration of 5 mm the spring back is 0.37 mm. The blue line shows the force and the blue line shows the punch penetration.

4.3.4 | Exploring The Dataset

The output data of the bending machine contained 26 features which can be found in the appendix. Out of these features only the standard power and the distance y_p and the force are relevant for calculating the spring back which was described in the last section 4.3.3. The final dataset therefore contained 3 features plus the added spring back. In total 396 data points where crated using the described approach. An example of the dataset is shown in Table 4.3.

	distance	spring back	thickness	die opening
1	5	0.6667	2.0	50
2	15	0.9164	2.0	50
3	10	0.6829	2.0	50
...	...	...	...	...
396	5	0.6667	3.0	10

Table 4.3: Features used for the machine learning models

4.3.5 | Computational Setup

For training the machine learning models a ThinkPad X1 Carbon 2019 with an Intel Core i7-10610U CPU @ 1.80GHz and 16 GB RAM was used. The operating system used is Ubuntu 20.04.2 LTS. The code for the model is written in Python 3.8.5 using the IDE PyCharm. The libraries used are mentioned in Table 4.4.

Library	Version
numpy	1.23.2
pandas	1.5.1
matplotlib	3.6.2

Table 4.4: Libraries used for the machine learning models.

Looking at the correlation matrix shown in Figure 4.5 it can be seen, that the distance and the spring back are more correlated than the other features. This is expected, because the punch penetration y_p is the main factor which influences the spring back. The other features are not correlated with each other, no multicollinearity is present which is good for the machine learning models.

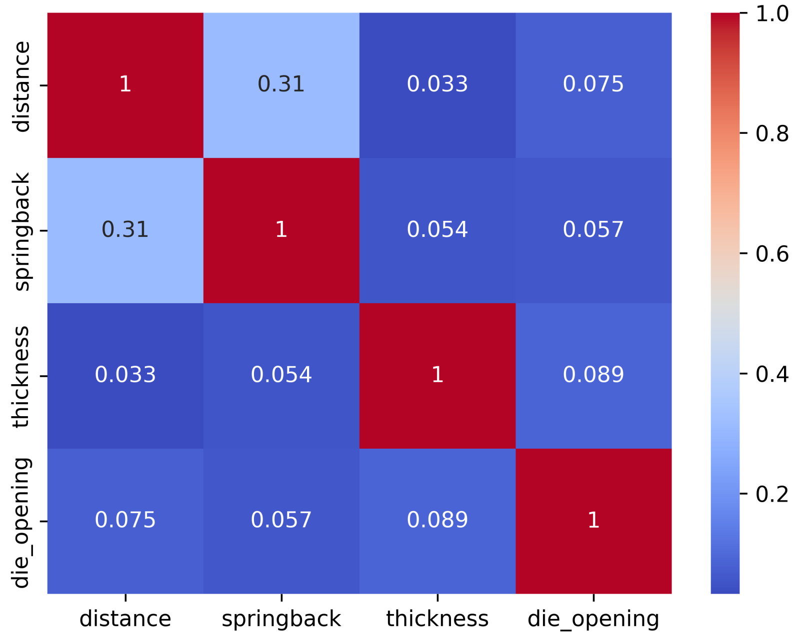

Figure 4.5: Correlation matrix

4.4 | Model Selection

4.4.1 | Support Vector Regression (SVR)

Support Vector Machines SVM are used for classification problems. The SVM algorithm is used to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. (Awad and Khanna, p. 42) Predicting the spring back is a regression problem, so the SVM algorithm is not suitable for this problem. It therefore is a generalization of classification problems, where the model returns continuous values instead of finite set of values. The Support Vector Regression Support Vector Regression (SVR) algorithm is derived from the SVM algorithm and accomblishes this generalization by introducing an ϵ-intensive region around the function ($\epsilon - tube$). (Awad and Khanna, p. 67)

4.4.2 | Multi Linear Regression

4.4.3 | Polynomial Regression

4.4.4 | Decision Tree Regression

4.4.5 | Random Forest Regression

A commonly used method in machine learning. The goal is to solve classification or regression problems by predicting the value of a output variable by one or multiple input variables. (Shaik and Srinivasan, p. 253) To build a Decision tree (DT) the source dataset represents the root node of the tree this data set is split into leafs (children) by using a set of spitting rules until each leaf in the DT is "pure" and only contains one target value. Depending on the use cases this is a single class or a single regression value. (Müller and Guido, p. 70-72) The main drawbback of DTs is the tendency to overfit and poor generalization performance, what makes them not paticaly for most use cases. Therefore usualy ensemble methods are used instead of a single DT. (Müller and Guido, p. 78) (Liu et al., c, p. 251) Random forest (Breiman) is a type of ensemble learning algorithm in which multiple decision trees, which are "weak learners," are trained and combined to produce a more accurate and stable prediction, known as a "strong learner." (Awad and Khanna, p. 24) The risk of overfitting is mitigated by subset and feature randomization. Each root node uses a unique subset of the data and each leaf is split using a random features. This ensures that no single tree sees all of the data, allowing the model to focus on general patterns rather than being sensitive to noise. (Liu et al., c, p. 251) In this supervised learning method, a "divide and conquer" approach is used. This involves dividing the data into smaller samples, incrementally building a randomized tree predictor for each sample, and then combining (aggregating) these predictors together. This approach has proven to be effective. Because not only one but multiple classifiers are used the random forest learning is known as ensemble model. (Shaik and Srinivasan, p. 254)

This mechanism is flexible enough to handle classifications and regression problems, this is one of the reasons that random forests count to the most successful ML methods. (Biau and Scornet, p. 3-4) (Breiman, p. 25)

Random forests are a type of machine learning algorithm that uses bagging and the random selection of features to produce accurate results. They are effective at handling noise and can work with both continuous and categorical variables. This combination of techniques helps improve the performance of the algorithm. (Liu et al., c, p. 259) Decision trees have a limitation in their ability to overfit, which is a disadvantage. This is mitigated by the use of subset and feature randomization. Specifically, each base model uses a unique subset of the data, and each node in the decision tree is split using a random set of features. This ensures that no single tree sees all of the data, allowing the model to focus on general patterns rather than being sensitive to noise. (Liu et al., c, p. 259)

4.4.5.1 | Gradient Boosting Regression Tree

A gradient boosting regression is a type of ensemble learning algorithm in which multiple decision trees are combines to produce a more accurate and stable prediction. Similar to the random forest algorithm gradient boosting combines multiple weak learners to create a strong learner. The difference to a random forest is, that the trees are trained in a serial manner and each tree corrects the errors of the previous tree. (Müller and Guido, p. 88-89) Gradient boosted tree use strong pre-pruning and therefore produce shallow trees with a depth of one to five. This brings the advantage of a smaller model which uses less memory and also results in a faster prediction. Usually generating more trees improves the overall performance of the model. (Müller and Guido, p. 88-89) Also the algorithm performs well without scaling the dataset and can handle a mixture of binary and continuous features. (Müller and Guido, p. 88-89) Like other tree-based models it does not perform well on high-dimensional data.

Results

4.5 | Model Training

4.5.1 | Training-Test Split

Figure 4.6 shows the data set and which parts of it is used for training and testing the used models. Samples with a die opening of 30 are used to test the performance and the remaining part is used for training. A different approach would be to us a random test and train split, this would lead to a better performance of the models but would not evaluate their ability to predict new data of a different die opening. The die opening 30 was chosen because it is in the middle of the selected data set and therefore the models should be able to predict the data of this die opening. All models are trained with the same data set and the same parameters. The only difference is the used algorithm.

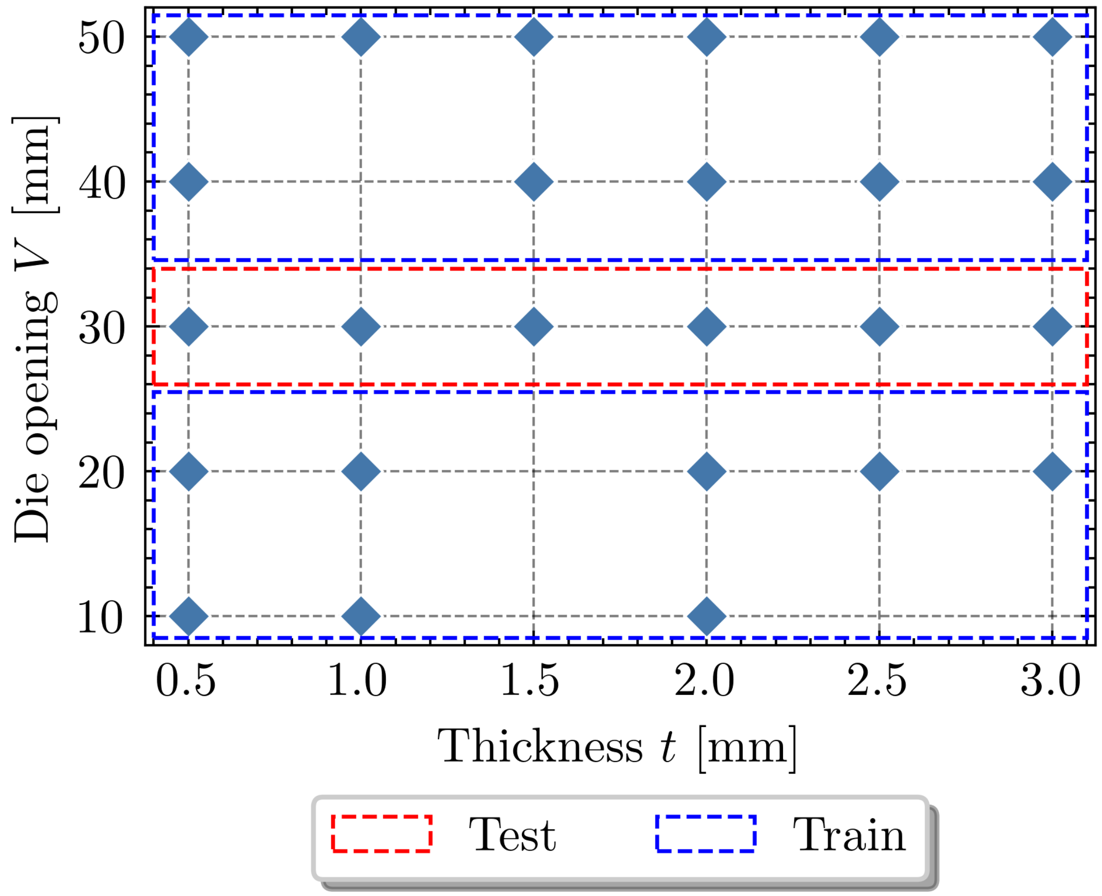

Figure 4.6: The training and test dataset. (Data especially for V40 is still missing)

4.5.2 | Random Forest

Hyperparameter Tuning

$n_e$stimators Sets the number of decision trees to construct. bootstrap: true, criterion: squared error, min samples leaf : 1, min samples split: 2, max features: 0.5119924258446656, n estimators: 35, n jobs: -1.

maxdepth "As expected, the accuracy on the training set is 100pure, the tree was grown deep enough that it could perfectly memorize all the labels on the training data. The test set accuracy is slightly worse than for the linear models we looked at previously, which had around 95accuracy. " (Müller and Guido, p. 133-136)

"If we don't restrict the depth of a decision tree, the tree can become arbitrarily deep and complex. Unpruned trees are therefore prone to overfitting and not generalizing well to new data. Now let's apply pre-pruning to the tree, which will stop developing the tree before we perfectly fit to the training data. One option is to stop building the tree after a certain depth has been reached. Here we set $max_depth = 4$, meaning only four consecutive questions can be asked (cf. Figures and). Limiting the depth of the tree decreases overfitting. This leads to a lower accuracy on the training set, but an improvement on the test set:" (**?**, p. 133-136)

4.5.3 | Gradient Boosted Regression Trees

Hyperparameter Tuning

"Gradient boosted trees are frequently the winning entries in machine learning competitions, and are widely used in industry. They are generally a bit more sensitive to parameter settings than random forests, but can provide better accuracy if the parameters are set correctly." (Müller and Guido, p. 88-89)

"Apart from the pre-pruning and the number of trees in the ensemble, another important parameter of gradient boosting is the learning rate, which controls how strongly each tree tries to correct the mistakes of the previous trees. A higher learning rate means each tree can make stronger corrections, allowing for more complex models. Adding more trees to the ensemble, which can be accomplished by increasing n estimators, also increases the model complexity, as the model has more chances to correct mistakes on the training set." (Müller and Guido, p. 88-89)

"The main parameters of gradient boosted tree models are the number of trees, n estimators, and the learning rate, which controls the degree to which each tree is allowed to correct the mistakes of the previous trees. These two parameters are highly interconnected, as a lower learning rate means that more trees are needed to build a model of similar complexity. In contrast to random forests, where a higher n estimators value is always better, increasing n estimators in gradient boosting leads to a more complex model, which may lead to overfitting. A common practice is to fit n estimators depending on the time and memory budget, and then search over different learning rates." (Müller and Guido, p. 88-89)

5

Evaluation

This chapter critically examines the machine learning models conceived and partially implemented in the previous chapter. Table 5.1 shows an overview of the evaluated criteria. These are structured according to the Goal-Question-Metric approach

Mean Absolute Error (MAE) how far the predictions are from the actual output

Goal	Question	Metric
Appropriatness	How well does the model type fit the current task?	Prerequisites for model type
Correctness	Ability of the model to perform the current task measured on the development dataset and the runtime dataset	Precision, Recall, F-score
Relevance	Does the model achieve a good bias-variance tradeoff? Which means neither overfitting or unterfitting the data.	Variance of cross-validation and fit
Robustness	Ability of the model to outliers, noise and other data quality issues	Variance of cross-validation, fit
Stability	Does the artifact generate repeatable results when trained on different data?	Equalized Loss of Accuracy (ELA)
Interpretability	How well can the model be explained?	Complexity measures (e.g., no. of parameters, depth)
Resource utilization	How much resources are required to train and run the model?	Training time, runtime, storage space

Table 5.1: Overview of the goals, questions and metrics for the evaluation of artifacts following the GQM approach.

5.1 | Correctness

5.1.1 | Random Forest

With 100 estimators in Random forest (RF) a mean squared error (mse) of 0.15 and root mean squared error (rmse) 0.39 was achieved. Figure 5.1 compares and visualizes the relative importance of the features used for training the model. As shown, the thickness is the most important feature followed by distance and die open. The results show, that all three featured are relevant for the outcome and so no feature can be removed from the dataset to get a better performance of the model.

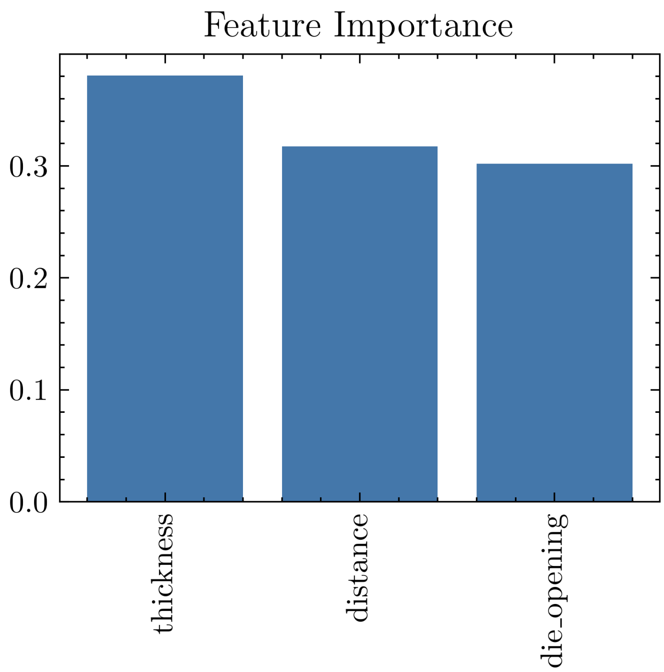

Figure 5.1: Relative feature importance

In further experiments boosting was used with the goal of predicting the target vector more precise. A mse of 0.27 and rmse of 0.52 as achieved and therefore no better performance to the default random forest.

5.2 | Appropriateness

5.3 | Correctness

To measure the correctness of the model, the following metrics are used:

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{5.1}$$

Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}e^2 \tag{5.2}$$

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE} \tag{5.3}$$

In the formulas 5.1, 5.3 and 5.3 e_i is the prediction error which is the difference between the predicted value by the model the actual value. y_i is the actual value and n is the number of samples in the testing data set.

"The mean absolute error (MAE) and mean squared error (MSE) are the most commonly used metrics for evaluating the performance of regression models. The MAE is the average of the absolute differences between the predicted and actual values. The MSE is the average of the squared differences between the predicted and actual values. The MSE is more sensitive to outliers than the MAE. The RMSE is the square root of the MSE. The RMSE is the most popular metric for evaluating the performance of regression models. The RMSE is interpretable in the same units as the response variable. The RMSE is more sensitive to outliers than the MAE.

5.3.1 | Overview of the used machine learning models and their metrics

Model Name 2	MAE	MSE	RMSE
Random Forest	0.15	0.03	0.18
Boosting	0.27	0.07	0.26
Linear Regression	0.27	0.07	0.26
Support Vector Regression	0.27	0.07	0.26

Table 5.2: Overview of the used machine learning models and their metrics.

5.4 | Relevance

5.5 | Robustness

5.6 | Stability

To measure the stability of the model, the following metrics are used:

Equalized Loss of Accuracy (ELA)

$$ELA = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\hat{y}_i}|y_i - \hat{y}_i| \tag{5.4}$$

5.7 | Interpretability

5.8 | Resource utilization

5.9 | Summary

6

Conclusions

6.1 | Revisiting the Aims and Objectives

6.2 | Critique and Limitations

6.3 | Future Work

6.4 | Final Remarks

References

Awad, M. and Khanna, R. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. The Expert's Voice in Machine Learning. Apress Open.

Baig, S. U. R., Wasif, M., Fatima, A., Baig, M. M. A., and Iqbal, S. A. (2021). Machine Learning for the Prediction of Springback in High Tensile Strength Steels after V-Bending Process Using Tree-Based Learning. Preprint, In Review.

Basili, V. R., Caldiera, G., and Rombach, H. D. (2002). THE GOAL QUESTION METRIC APPROACH. *NA*, page 10.

Biau, G. and Scornet, E. A random forest guided tour. 25(2):197–227.

Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R., and Klusemann, B. A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics. 6:110.

Breiman, L. Random Forests. 45(1):5–32.

Cao, J., Brinksmeier, E., Fu, M., Gao, R. X., Liang, B., Merklein, M., Schmidt, M., and Yanagimoto, J. Manufacturing of advanced smart tooling for metal forming. 68(2):605–628.

Cruz, D. J., Barbosa, M. R., Santos, A. D., Miranda, S. S., and Amaral, R. L. (2021). Application of Machine Learning to Bending Processes and Material Identification. *Metals*, 11(9):1418.

Dib, M. A., Oliveira, N. J., Marques, A. E., Oliveira, M. C., Fernandes, J. V., Ribeiro, B. M., and Prates, P. A. Single and ensemble classifiers for defect prediction in sheet metal forming under variability. 32(16):12335–12349.

Gregor, S., Hevner, A. R., and University of South Florida (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37(2):337–355.

Gregor, S. and Jones, D. (2017). THE ANATOMY OF A DESIGN THEORY. *College of Business and Economics The Australian National University*, page 60.

Groover, M. P. *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*. John Wiley & Sons, Inc, seventh edition edition.

Hevner, March, Park, and Ram (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75.

Hevner, A. and Chatterjee, S. (2010). *Design Science Research in Information Systems*, volume 22, pages 9–22. Springer US, Boston, MA.

Inamdar, M., Date, P. P., Narasimhan, K., Maiti, S. K., and Singh, U. P. Development of an Artificial Neural Network to Predict Springback in Air Vee Bending. 16(5):376–381.

Kazan, R., Fırat, M., and Tiryaki, A. E. Prediction of springback in wipe-bending process of sheet metal using neural network. page 6.

Kim, H., Nargundkar, N., and Altan, T. Prediction of Bend Allowance and Springback in Air Bending. 129(2):342–351.

Koppenhagen, N., Gaß, O., and Müller, B. (2012). Design Science Research in Action - Anatomy of Success Critical Activities for Rigor and Relevance. *NA*.

Liu, S., Shi, Z., Lin, J., and Li, Z. Reinforcement learning in free-form stamping of sheet-metals. 50:444–449.

Liu, X., Du, Y., Lu, X., and Zhao, S. Springback Prediction and Forming Accuracy Control of Micro W-bending Using Support Vector Machine. In *2019 6th International Conference on Frontiers of Industrial Engineering (ICFIE)*, pages 23–27. IEEE.

Liu, Y., Wang, Y., and Zhang, J. New Machine Learning Algorithm: Random Forest. In Liu, B., Ma, M., and Chang, J., editors, *Information Computing and Applications*, volume 7473 of *Lecture Notes in Computer Science*, pages 246–252. Springer Berlin Heidelberg.

Miranda, S. S., Barbosa, M. R., Santos, A. D., Pacheco, J. B., and Amaral, R. L. Forming and springback prediction in press brake air bending combining finite element analysis and neural networks. 53(8):584–601.

Müller, A. C. and Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc, first edition edition.

Narayanasamy, R. and Padmanabhan, P. Comparison of regression and artificial neural network model for the prediction of springback during air bending process of interstitial free steel sheet. page 8.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77.

Sein, Henfridsson, Purao, Rossi, and Lindgren (2011). Action Design Research. *MIS Quarterly*, 35(1):37.

Shaik, A. B. and Srinivasan, S. A Brief Survey on Random Forest Ensembles in Classification Model. In Bhattacharyya, S., Hassanien, A. E., Gupta, D., Khanna, A., and Pan, I., editors, *International Conference on Innovative Computing and Communications*, volume 56 of *Lecture Notes in Networks and Systems*, pages 253–260. Springer Singapore.

Siebert, J. Construction of a quality model for machine learning systems. page 29.

Sonnenberg, C. and vom Brocke, J. (2012). Evaluation Patterns for Design Science Research Artefacts. In Helfert, M. and Donnellan, B., editors, *Practical Aspects of Design Science*, volume 286, pages 71–83. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sáez, J. A., Luengo, J., and Herrera, F. Evaluating the classifier behavior with noisy data considering performance and robustness: The Equalized Loss of Accuracy measure. 176:26–35.

von Rennenkampff, A., Nissen, V., and Stelzer, D. (2015). *Management von IT-Agilität: Entwicklung eines Kennzahlensystems zur Messung der Agilität von Anwendungslandschaften*. Number Band 2 in Ilmenauer Schriften zur Wirtschaftsinformatik. Univ.- Verl. Ilmenau, Ilmenau.

Zheng, K., Politis, D. J., Wang, L., and Lin, J. A review on forming techniques for manufacturing lightweight complex—shaped aluminium panel components. 1(2):55–80.

Zhou, Z.-H. *Machine Learning*. Springer Singapore.