Comparative Analysis of Machine Learning Algorithms for Predictive Maintenance in the Manufacturing Industry

Roland Zsolt Nagy

2024

Comparative Analysis of Machine Learning Algorithms for Predictive Maintenance in the Manufacturing Industry

Corvinus University of Budapest

Institute of Data Analytics and Information Systems

Business Informatics Engineer BSc

Thesis Advisor:
Dr. László Kovács, PhD

2024

Table of contents

List of figures and tables

1. Introduction

This thesis introduces using different machine learning algorithms in a manufacturing environment to predict a maintenance-related value (a machine's energy consumption). Among the applied models, it investigates the most effective approach (algorithm) based on a comparative analysis of their predictive performance in a real-world manufacturing problem. The entire machine learning process and pipeline is presented, beginning from data preprocessing and exploratory data analysis through model training to performance evaluation. Besides, the utilised models are discussed in a systematic way as well, and in order to achieve a more in detail and comprehensive understanding of their predictive performance the underlying relationship in the data are also investigated.

The addressed manufacturing problem is about estimating the energy consumption or energy requirement (used interchangeably throughout the study) of a CNC (Computer Numerical Control) milling machine based on its pre-set working parameters (axis, feed [mm/min], path [mm]) set prior to the milling process. As the variable to be predicted is a continuous value (kJ), regression algorithms are utilised, specifically Linear Regression, Random Forest Regression, and Support Vector Regression.

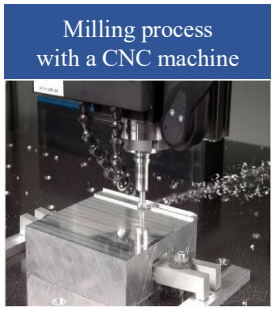

Figure 1: CNC milling process.
Source: PCBWay, 2021.

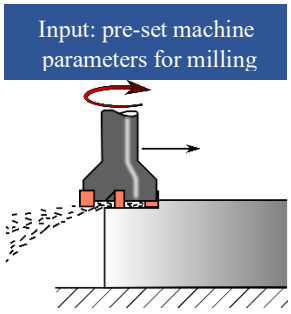

Figure 2: Pre-set parameters of the machine.
Source: „Pasixxxx", 2009

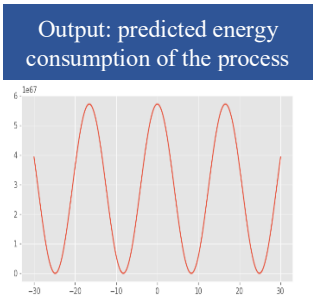

Figure 3: Energy consumption prediction.
Source: „sbhhdp", 2020.

2. Preliminary considerations

2.1. Literature overview

Utilising machine learning algorithms and conducting a comparative analysis to investigate the most accurate model to a specific problem is an important area of research within the manufacturing industry. The results from this kind of research has implications in predictive maintenance, operational efficiency and environmental impact (Çınar et al., 2020). Potentially, economic benefits and sustainability goals can be achieved through predicting (and optimizing) important manufacturing-related indicators.

Addressing one of the previously undertaken academical initiatives to develop predictive models for manufacturing issues, Wu et al. (2017) investigates predicting tool wear in smart manufacturing systems with a focus on improving predictive maintenance strategies. The study explores the effectiveness (accuracy) of Random Forest Regression, Artificial Neural Network and Support Vector Regression and compares their performance. Data collected from milling tests involving various sensors is used to monitor tool conditions, the input features for the models were statistical features (maximum, minimum, median, etc.) extracted from cutting force, vibration, and acoustic emission. The comparative analysis across the applied models highlighted that Random Forest offered the highest accuracy and reliability for predicting tool wear in milling processes when compared to the other models.

Schorr et al. (2020) studied the application of different machine learning techniques to predict workpiece quality (the concentricity and the diameter of drilled and reamed bores in hydraulic valves) based on torque measurements from a milling machine. The approach presents a method to increase manufacturing efficiency by predicting product quality early in the production process, thus minimizing waste and improving quality control. The utilised models were Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Support Vector Regression (SVR), Random Forests Regression (RF), and AdaBoost Regression (ABR), the input features were derived from the torque measurements. The best performance was achieved by Random Forest Regression showing very precise predictions and the lowest MAE values, while the worst predictive accuracies was yielded by CNN and ANN.

Besides, Agrawal et al. (2015) explored the prediction of surface roughness during the hard turning of a specific (AISI 4340) steel. The used input features were feed rate, depth of cut and spindle speed, for predicting (Multiple) Linear Regression and Random Forest Regression were utilised. The latter model showed superior performance over the other one in terms of prediction

accuracy, and a strong correlation was observed between the model predictions and the actual results. Quantile Regression was used to examine the effects of input features across different quantiles of the surface roughness distribution, and the results highlighted that the impact of feed rate is less at lower quantiles of roughness, and it increases in the upper quantiles roughness. It was also found that the feed rate is the most influential parameter affecting surface roughness, followed by spindle speed and depth of cut.

In addition, Brillinger et al. (2021) conducted a study that is significant in the context of global efforts to reduce energy consumption and CO2 emissions in the industrial manufacturing sector. In this research the energy consumption during CNC machining was predicted with the main input parameters length of tool path, feed rate and operating method of the machine. Variations of decision trees were utilised for modelling, specifically Decision Tree Regression, Random Forest Regression, and Boosted Random Forest Regression (with AdaBoost). After evaluating and comparing them based on predictive accuracy, Random Forest Regression (without boosting) provided the best performance and appeared to be notably accurate in predicting the energy demand of CNC machining operations.

Next, Dubey et al. (2022) investigated predicting the surface roughness in the turning process of a specific (AISI 304) steel under minimum quantity lubrication (MQL). This process incorporates alumina nanoparticles within the cutting fluid, testing two particle sizes (30 nm and 40 nm). The study's primary objective is to optimize the machining parameters to enhance surface quality and efficiency. The input features were cutting speed, depth of cut, feed rate and nanoparticle concentration, and the models Linear Regression, Random Forest Regression and Support Vector Regression (SVR) were utilised. Again, the Random Forest model outperformed SVR and Linear Regression in predicting surface roughness for both particle sizes. It was also found that the use of nanoparticles in the cutting fluid significantly impacts the machining outcomes, with different sizes showing varying effects on the surface roughness.

Finally, Charalampous (2021) addressed the challenge of predicting cutting forces in the face milling process of a specific (AISI 4140) steel. This study explored the application of both machine learning algorithms and finite element analysis. The objective was to contribute to optimizing the milling process and to the possibility of integrating these models into CAM software to enhance productivity. The data for model training were obtained from multiple milling experiments. They key input features included cutting speed, radial depth of cut and feed per tooth. The models Random Forest Regression, Support Vector Regression (SVR), K-Nearest Neighbors Regression, and Polynomial Regression were utilised. Contrary to the findings of the studies presented so far, in this research SVR yielded the best predictive

accuracy, while Random Forest Regression performed the worst. The four machine learning models were found to be faster and at least as accurate as the finite element model in predicting outcomes, but the latter model provided detailed insights into the stress and temperature distributions during cutting.

Overall, the presented studies highlight the intersection of machine learning and manufacturing, demonstrating the potential of advanced analytical techniques to drive efficiency and sustainability in industrial operations, which validates the relevance of the current study.
Based on the findings of these former studies, Random Forest Regression can be assumed to have the best predictive performance among the models applied in this study for the addressed energy consumption prediction problem. However, every industrial setting is different yielding different predictive performance results, and as for example the study by Charalampous (2021) demonstrated, for some manufacturing problems Support Vector Regression may well outperform Random Forests. In addition, the specific problem of predicting the energy consumption of a milling process was addressed in the study by Brillinger et al. (2021), where only variations of decision trees were utilised for modelling. This also validates the relevance of the current study, which besides Random Forest utilises and compares the performance of Support Vector Regression and Linear Regression as well.

2.2. Research question

The research question

The research question is, among the applied machine learning algorithms which one is the most accurate in predicting the energy consumption of a milling machine investigated in this study?

Reason and objective of the research question: the motivation behind

The motivation for finding the most accurate machine learning algorithm is the possibility of optimizing the energy consumption of a milling machine by predicting the consumed energy based on the machine's pre-set working parameters. This supports forecast-based maintenance, and in the long run leads to improved productional and operational efficiency, reduced downtime, and cost savings (Bányai, 2021). The motivation for predicting the energy consumption also involves the possibility of load management and energy planning, of balancing peak power to minimize overall power surges across the factory (Brillinger et al., 2021), and of the detection of deviations through the comparison of the predicted and the actual energy profiles of the milling process (Pawanr et al., 2021). As a consequence, the concept of

a more sustainable type of manufacturing and of Industry 4.0. becomes more achievable (Çınar et al., 2020).

Expected answer

For the energy consumption problem addressed in this Thesis, the performance of three regression models will be compared that implement substantially different approaches for the regression task as machine learning models (section 5.2 discusses more in detail the utilised models):

- Linear Regression, as it serves as a good baseline model. It is simple, models linear relationships, and provides a point of reference to measure the performance of more complex models (Seber & Lee, 2012).
- Random Forest Regression is an ensemble method (combines predictions from multiple models) that is robust to overfitting and can capture non-linear relationships without needing to explicitly define them. It is also less sensitive to outliers, and by comparing it with Linear Regression, it can be understood if capturing non-linear relationships is necessary for the addressed problem (Vinh & Huy, 2022).
- Support Vector Regression can model complex relationships thanks to using the kernel trick, which transforms or projects the data that is not linearly separable in the original space into a higher-dimensional kernel space (feature space) where it becomes linearly separable. For example, when there are relationships in the data that can be modelled by a parabola in the original two-dimensional space, they become capturable by a linear plane in the higher-dimensional kernel space. It's valuable to see if such complex transformations result in better performance (Smola & Schölkopf, 2004).

Considering the former studies presented in section 2.1. (Literature overview), Random Forest Regression is most likely to have the best predictive performance. This might be since it has a great balance between handling non-linear relationships and robustness to overfitting (Schonlau & Zou, 2020), which are crucial as non-linear patterns and anomalies are likely to be present in the addressed real-world milling problem (Nguyen, 2019).

More complex regression models are not used due to their need for large amounts of data to train properly (later in Table 1. it is shown that there are 226 observations before removing outliers, which is considered a smaller dataset), and due to the desire to prevent overfitting (Bousquet et al., 2004). Another reason is that in an industrial manufacturing setting interpretability and simplicity is preferred (Doshi-Velez & Kim, 2017). For example, neural

networks without sufficient training data usually do not outperform the simpler models due to their dependency on large datasets for optimal performance, and they are also computationally quite intensive (Caruana & Niculescu-Mizil, 2006). Applying other different models to the addressed manufacturing problem is subject to further research and is mentioned in section 10, where other limitations of this study as possible research directions are listed too.

2.3. Hypotheses

- As non-linear relationships are likely to be present in the data that describes the addressed milling problem, Linear Regression will not be able to properly model the underlying patterns and will demonstrate a poor predictive performance compared to the other two models (Nguyen, 2019).
- As Support Vector Regression can capture non-linear relationships, it will be able to properly model the relationships in the data and will not substantially underperform Random Forest Regression (Charalampous, 2021).
- By examining patterns in the data in detail, it is possible to find relationships that explain the results of the performance comparison (Murphy, 2012).

2.4. Applied methodologies for data collection and analysis

In this chapter, proper data collection and analysis methodologies are selected to investigate the hypotheses and the research question. The required information (data) to investigate the hypotheses and the research question is the (visual and numerical) data based on which the applied machine learning models are evaluated. The predictive performance of the models is going to be measured, analysed and compared visually through Residual plots, and numerically through Mean Square Error and Mean Absolute Error metrics (using other evaluation metrics is subject to further research as discussed in section 10).

Residual plots

Residual plots are a diagnostic tool used in statistical analyses to visualise the differences between observed and predicted values (as residual = actual value – predicted value). Plotting these differences (the residuals) on the vertical axis against the predicted values on the horizontal axis allows to detect non-linearity and unequal error variances (heteroscedasticity) and helps to examine whether the model fits the data appropriately (Anscombe & Tukey, 1963). In terms of a residual plot in the ideal case the residuals are randomly distributed around the horizontal axis, indicating that the model has captured most of the explanatory information

(captured effectively the variance in the data) and that the model's predictions are unbiased. Specifically, in this case residuals are randomly scattered around the center line of zero with no obvious pattern and look like an unstructured cloud of points meaning that the magnitude of the error does not depend on the estimated value of the target variable, thus the residual is generally not larger or smaller in case of any larger or smaller prediction.

In case of a non-random pattern visible in the plot, it is suggested that the given model is not the best fit for the problem as it is missing some aspect of the underlying data structure, which could mean that non-linear or more complex relationships exists in the data that has not been properly accounted for.

If the spread of residuals increases or decreases with the predicted values, it indicates heteroscedasticity (unequal error variances), which suggest poor predictive performance and also violates the assumption of linear regression regarding homoscedasticity (constant error variance). Section 5.2. discusses more in detail homoscedasticity and other assumptions and limitations of linear regression.

Mean Square Error (MSE)

MSE calculates the average squared difference between the estimated or predicted values and the actual values. MSE is particularly useful because it penalizes larger errors more severely, emphasizing significant discrepancies between predicted and observed values (Willmott & Matsuura, 2005). The formula for MSE is as follows:

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

where n is the number of observations, Y_i-s are the actual observed values and $\hat{Y}_i$-s are the predicted values.

Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is calculated as the average of the absolute differences between predictions and actual values where all individual differences have equal weight, thus it offers a more robust measure against outliers than MSE. MAE is particularly useful when a measure is needed that reflects the extent of the error in the same unit as the data (Willmott & Matsuura, 2005). The formula for MSE is as follows:

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |Y_i - \hat{Y}_i|$$

, where the meaning of the variables is the same as for the MSE formula.

Data collection

For residual plots Matplotlib scatter plots are created, in order to get the MSE and MAE measures the respective functions from the *sklearn.metrics* Python library are used.

Data analysis

Residual plots are visually analysed examining patterns and the magnitude of the errors to derive insights of the models' predictive performance.

To analyse the numerical evaluation metrics (MSE, MAE) of the applied models, a bar plot is created using Matplotlib's proper function (*matplotlib.pyplot.bar*), on which plot the 3x2 bars (number of models x number of metrics) are displayed side-by-side allowing a visual comparison of the applied models' performance and the possibility to prove (or disprove) the respective hypotheses.

2.5. Expected findings and use of results

Expected findings and results

Identification of the best-performing model for the addressed milling machine energy consumption problem, and comparison of how the applied regression models' performance differ in terms of predictive accuracy.

Practical use of results, as the motivation to achieve them

As the industry of manufacturing is confronted by trends such as continuing industrial automation and predictive maintenance, it is beneficial to predict accurately and in a repeatable way the energy consumption of machining processes, which can lead to several positive implications (Bányai, 2021).

Taking into consideration and customizing the approaches of this study, hence using different machine learning algorithms for an industrial problem and evaluating their effectiveness (accuracy) in the context of the addressed setting, manufacturing practitioners can utilise data and machine learning and choose the most suitable model for their specific production processes (Theissler et al., 2021). This can help optimize the use of resources and of machines, reduce downtime, save costs, and ultimately lead to a greener future for the manufacturing industry (Çınar et al., 2020).

Accordingly, the prediction of a machine's energy profile can lead to substantial energy savings in manufacturing plants, which not only contributes to reducing energy consumption and related costs, but also has a positive impact on the environment (Sangwan & Sihag, 2019).

In the broader context, predicting energy consumption of a CNC machine not only aligns with economic benefits, but also with the global push towards more sustainable and responsible manufacturing practices. The approach, methods and findings of this thesis can help manufacturers achieve their environmental and business goals and objectives by optimizing their production systems (Groten & Gallego-García, 2021).

3. The addressed manufacturing problem

3.1. Introduction to the milling process

The context of the addressed problem and the objective is to perform a regression analysis on a CNC (Computer Numerical Control) milling machine using different models to predict the target variable, which is the energy consumption of the machine or in other words the energy requirement to execute the milling process. The milling process is a machining process where rotary cutters are utilized to remove material by moving a cutter into a piece of work, and is usually used for creating flat surfaces.

Numerical observations were carried out on the specific CNC milling machine under investigation in this study to collect sufficient data for proper training of the regression models. The observations are assumed to be random and independent of each other, thus the resulting dataset is presumed to be representative of the addressed real-world manufacturing problem. The energy forecast is based on the input variables, which are parameters of the machine set prior to the milling process (before starting the machining operation):

- *Axis*
- *Feed* [mm/min]
- *Path* [mm]

To get a better understanding of these parameters for the machine, please see Figure 4. below.

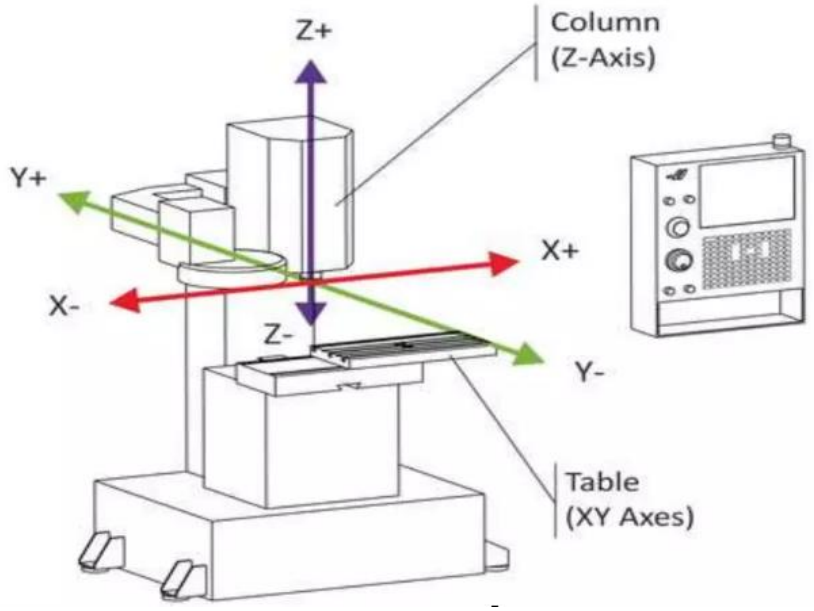

Figure 4: CNC milling machine. Source: “VinothkumarSivaraj”, 2020.

The milling machine investigated in this study has multiple axes for movement. To be able to measure and collect data, the Cartesian (three-dimensional) coordinate system is used, where the milling machine's cutting tool can be controlled and moved along 3 axes. Based on each axis, we typically get the following movements from the perspective of an operator facing the machine:

- X axis allows movement “left” and “right”
- Y axis allows movement “forward” and “backward”
- Z axis allows movement “up” and “down”

In Figure 4, the coloured arrows show these 3 axes. Accordingly, the parameter *Axis* can have one of these 3 values: 1 (X axis), 2 (Y axis), or 3 (Z axis), indicating along which axis the machine's cutting tool was moving during that particular milling process.
The *Path* parameter shows the distance that the milling machine's cutting tool has moved along the indicated axis and through the material being milled. It is measured in millimetres (mm) (Yuwen et al., 2022).
The *Feed* machine parameter refers to the speed at which the milling machine's cutting tool has moved along the indicated axis and through the material being milled and is measured in millimetres per minute (mm/min). A higher feed rate would indicate a faster cut, while a lower feed rate a slower, more careful cut.
These parameters (input variables or features) are of direct impact on the energy consumption of a milling process, thus they provide a quality input for analysing the energy profile of the milling process (Brillinger et al., 2021).

3.2. Statistical overview of the data

Loading necessary libraries

All the calculations, modelling and visualisation in this study was carried out in a Python project in the form of a Jupyter Notebook. I started with a blank project, and everything was built and computed from scratch in order to achieve maximum transparency. All the results on which the discussions in the thesis are based can be found in the Notebook and can be exactly reproduced. I uploaded the Notebook file to GitHub (“Thesis_RolandZsoltNagy.ipynb”), and it can be accessed through the following link: https://github.com/rolandzsoltnagy/Thesis_RolandZsoltNagy/blob/59bca8c0ebabdf101fab1a11a4766b78a132503f/Thesis_RolandZsoltNagy.ipynb.

The heading titles and their numbering in the Notebook are corresponding to the section titles and the section numbering in the thesis. In the beginning of the Notebook the necessary libraries and functions are imported to perform the regression analysis. The Python libraries that will be used throughout this study: *NumPy, Pandas, Matplotlib*, and *Scikit-learn* (Shang & Lakshmi T C, 2020). In addition, for figures white background with a grid is set with the *Seaborn* library to make the generated plots more visually analysable.

Loading the dataset

In this step the dataset generated by numerical observations on the investigated machine is accessed from the data repository *Zenodo* (Sule, 2021). For this, the *read_csv* function is used from the Pandas library to read the locally downloaded dataset (stored as a csv file) and load it into a *Pandas DataFrame* object.

Statistical data analysis

For the overview of the dataset, descriptive statistics are generated with the built-in Pandas *describe* function. A summary can be seen of the count, mean, standard deviation, minimum, 25th percentile, 50th (median) and 75th percentile, and the maximum for each of the variables in Table 1. The measurement scale of the *Axis* variable is nominal (1, 2, 3 is just coding for the X, Y, Z axes), the 4 basic mathematical operations between its values (1, 2, 3) are not reasonable, therefore the statistical indicators (mean, standard deviation, etc.) cannot be interpreted for this variable.

	Axis	Feed	Path	Energy_Requirement
count	224.000000	225.000000	225.000000	226.000000
mean	1.977679	1759.200000	1.644444	0.068117
std	1.545844	887.559998	43.726604	0.155441
min	-15.000000	20.000000	-200.000000	-0.262149
25%	1.000000	1000.000000	-30.000000	0.022769
50%	2.000000	2000.000000	10.000000	0.048631
75%	3.000000	2500.000000	40.000000	0.073982
max	10.000000	3000.000000	150.000000	0.800000

Table 1: A statistical overview of the dataset. Source: own editing.

Based on this overview, for example by examining the standard deviation, the variability or (average) dispersion of data points from the mean can be seen for each variable (for *Axis* it's not applicable due to the mentioned reason). As the variables are on different units of

measurement (mm/min, mm and kJ), instead of comparing the displayed “std”-s, it is more beneficial to compare the ratio of these standard deviations to the respective means. This measure is called relative standard deviation, and provides an intuitive way of comparing the dispersion of values across variables measured on different scales by using a relative approach. For instance, a randomly selected milling process's feed is expected to differ by +- 887.6 mm/min from the mean of 1759.2 mm/min, resulting in 50.5% relative standard deviation. For the *Path* variable, the relative standard deviation is 2659,1%, as the standard deviation (43.7 mm) is more than 26 times (calculated with the exact numbers) higher than the average value (1.6 mm). Comparing the these two relative standard deviations, it can be stated that the *Feed* is far less volatile than the *Path* variable, or in other words *Path* displays a relatively much wider distribution of values.

Furthermore, it is possible to check for data quality issues to see if any changes need to be made to the dataset: as mentioned earlier, the *Axis* feature can only take the values 1, 2, or 3 (according to the 3 axes of the machine), therefore the minimum and the maximum values for this parameter (-15.0 and 10.0) indicate incorrect data entries that have to be addressed before using the dataset.

Besides, the count metric for each variable should also be taken into consideration, as for all 3 features it is indicating missing values: there are 2 observations (records) in the dataset where the value for the *Axis* parameter is missing, and 1-1 data point (or they can be the same observation) where values for the *Feed* and/or *Path* variables are missing. For the target variable, there are no observations where the energy requirement value is missing.

Another approach to analysing the results of the generated descriptive statistics is to compare the mean and the median values, an idea of the shape of the distribution for each feature can be concluded: If the distribution of the data is symmetrical, the mean and median will be approximately equal. However, if the data is skewed, the mean will be pulled towards the tail as shown in Figure 5. below, and a significant difference between the mean and median will be observable.

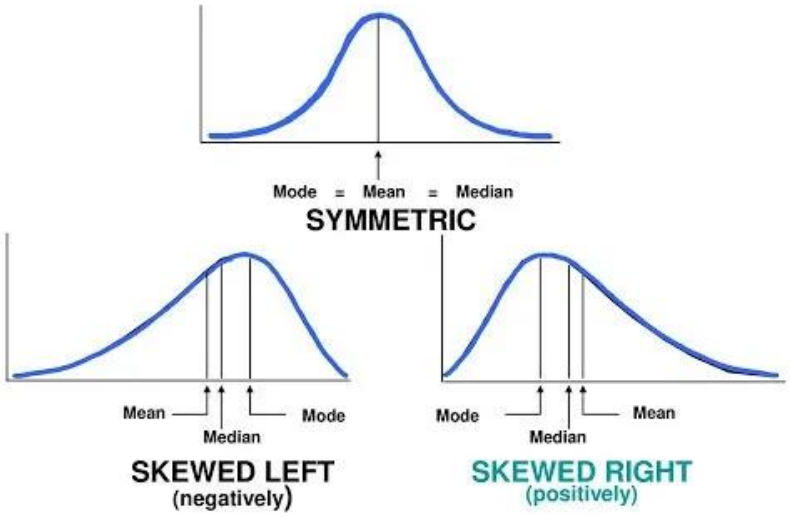

Figure 5: Skewness of distribution. Source:"Samueldavidwinter", 2022.

Investigating the realized statistical measures in Table 1., the *Feed* and *Path* variables suggest left-skewed distributions: the mean is lower than the median, signifying a distribution where the majority of the data points are relatively high, with a few smaller values (indicating outliers) pulling the mean to the left of the median. Left skewness is also known as negative skewness (the skewness measure like Pearson's is negative), which indicates that the tail on the left side of the probability density function is longer or "fatter" than on the right side, meaning that the mass of the distribution is concentrated on the right of the figure (Groeneveld & Meeden, 1984). In case of the target variable, the mean is higher than the median suggesting a right-skewed distribution, where the tail is longer on the right side of the distribution's peak, indicating that a few unusually high values are pulling the mean to the right of the median. For positive skewness (the skewness measure like Pearson's is positive), the small amount of very high values stretching out the tail can be an indicator of outlier values.

4. Data preprocessing

4.1. Visual data exploration and data cleaning

Distribution visualised on histograms

In the context of exploratory data analysis (EDA), the visualisation of the distribution of variables is a critical step in understanding the underlying tendencies and anomalies within the dataset, which allows data cleaning and preparation. In the Python project Matplotlib's built-in histogram (and bar plot) function is used for visualisation. Among the various tools for EDA, histograms stand out as a fundamental and highly effective method for understanding the distribution of variables (Behrens, 1997).

For example, one of the advantages of using histograms is that they make it possible to easily identify the shape of the distribution (according to Figure 6) of each variable, thus whether they are normally distributed, skewed (left or right), bimodal, multimodal, or uniform. Identification of the shape of the distribution is useful, as it can affect the performance of machine learning models (Murphy, 2012).

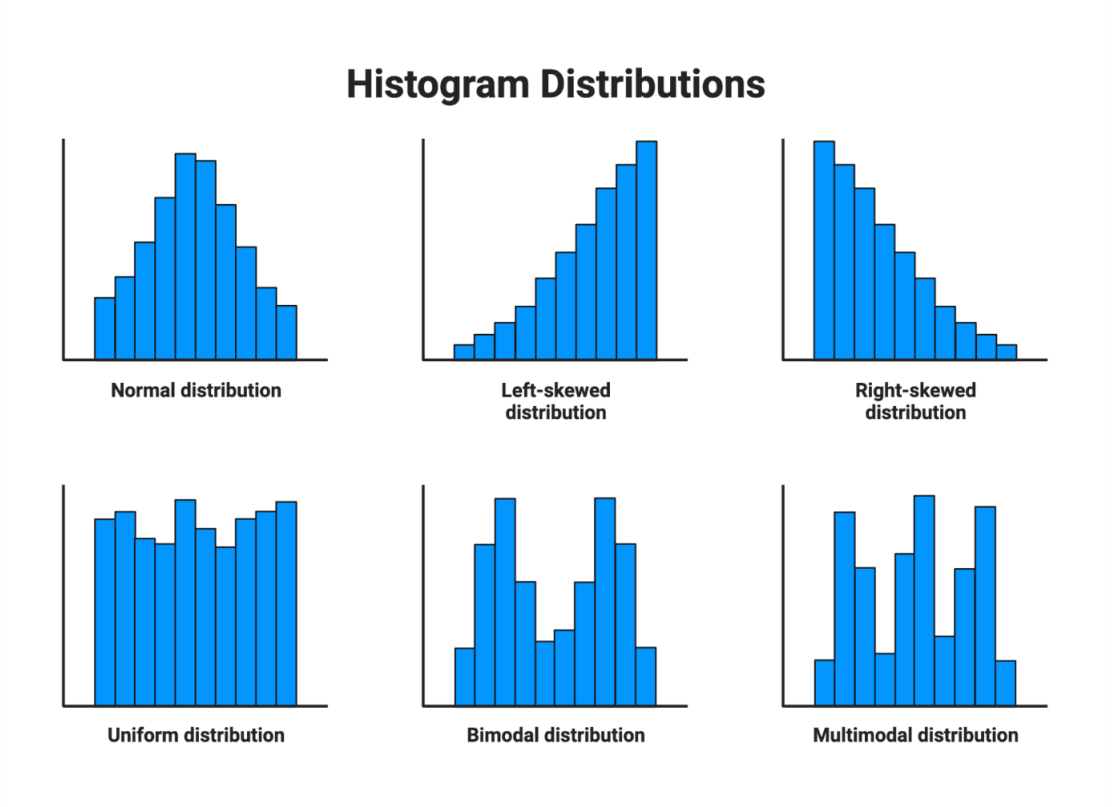

Figure 6: Identifying shape of the distribution using histograms. Source: Huang, n.d.

Another advantage of histograms is that they highlight outliers effectively. Outliers can disproportionately influence model training, and by using histograms they are easily spotted, allowing for informed decisions regarding treatment of these extreme values (García et al., 2015).

In this section, based on the findings and conclusions drawn from the histograms, the dataset is cleaned from outliers and missing values to prepare it as the input for the utilised machine learning models.

***Axis* feature**

As this variable is nominal (1, 2, 3 values are encodings for the axes), in this exceptional case drawing a histogram would just make anomalies harder to identify, thus instead of a histogram, a bar plot is drawn from the result of the Pandas *value_counts* method applied on this variable.

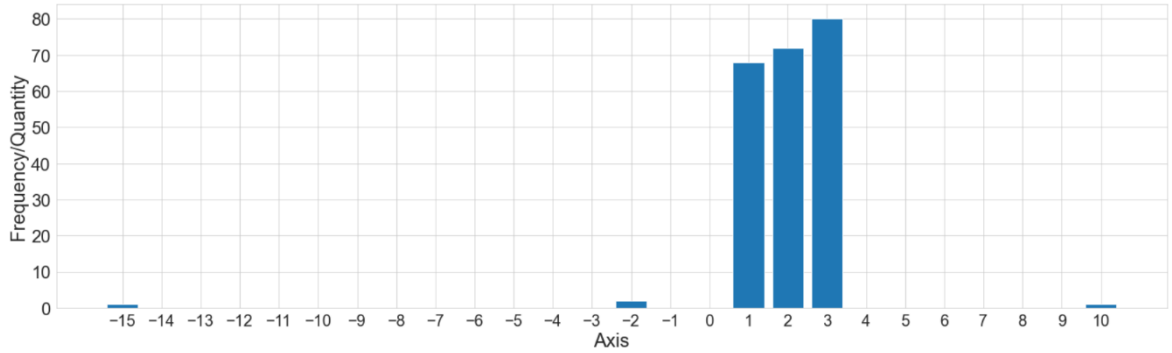

Figure 7: Frequency chart for the Axis variable. Source: own editing

The *unique* method applied on the *Axis* variable of the *DataFrame* (which is a *Pandas Series*) shows the actual unique values: [-15.0, -2.0, 1.0, 2.0, 3.0, 10.0, nan].

Outliers

This variable should nominally contain a discrete set of such values that correspond to the axes of movement of the CNC machine (the X, Y, and Z axes, represented by 1, 2, and 3 respectively). However, in Figure 7. a smaller number of occurrences can be seen for some values outside of the 1 to 3 range. These data points do not correspond to the standard axis designations and therefore indicate either data recording errors, data entry anomalies, or placeholder values that have not been properly cleaned. This, and the fact that the *unique* function returned a missing value (“nan”) too as a unique value in the dataset for this variable, indicates the need for data cleaning to address these invalid and missing values.

Removing invalid values

According to the findings and conclusions above, in the data cleaning process only samples with their *Axis* variable set to either 1, 2, or 3 are kept in the *DataFrame*, all other entries are removed.

After checking the relevant counts in the Python project, it can be stated that altogether 6 observations were removed due to anomalies and missing values in the *Axis* variable, accounting for 2,7% of the whole dataset (226 observations), which is not a significant loss of data.

The distribution of the variable can be examined again in Figure 8. after the data cleaning process.

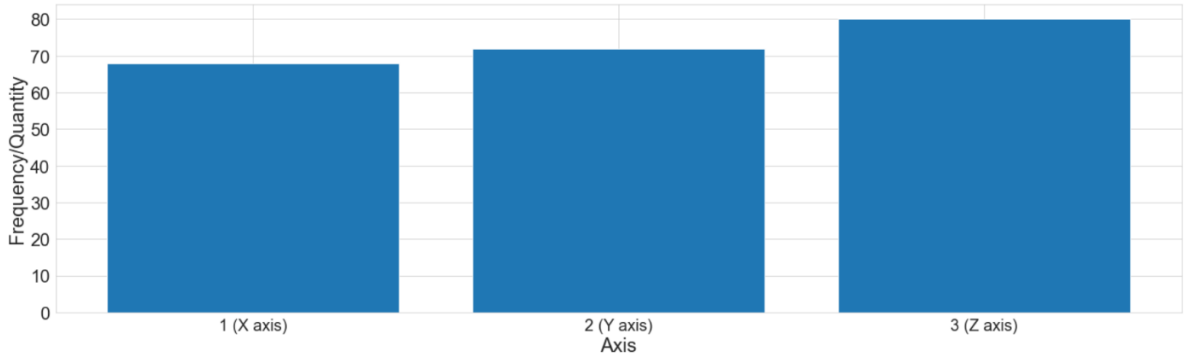

Figure 8: Frequency chart for the Axis variable after cleaning. Source: own editing.

In this chart it can be seen that the data cleaning was successful. It is also shown that among the collected data, most of the tests (numerical observations) were carried out on the third (*Z*) axis of the CNC machine, then a bit less on the second (*Y*) axis, and the least on the first axis (*X*).

Feed feature

The next feature to visualise the distribution of is the *Feed*. This variable represents the speed at which the CNC milling machine's tool moves through the material, measured in millimetres per minute (mm/min).

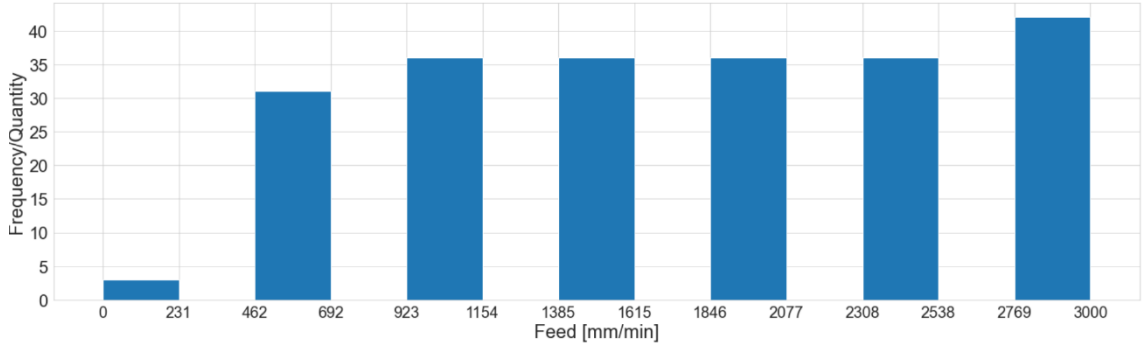

Figure 9: Distribution of the Feed variable. Source: own editing.

A first visual inspection of the distribution seen in Figure 9. suggests that except for the first bin, the feed rates are uniformly distributed: the rates tend to cluster around specific intervals like for example between 2308 and 2538 with nearly the same frequency. Based on the last bin it seems like the highest (fastest) feed rates are used slightly more commonly, while based on the bin after the first one it can be stated that the slowest (lowest) feed rates are used slightly less frequently. Therefore, feed rates are not perfectly uniformly, but nearly uniformly distributed.

Calling the *unique* function on the *Feed* variable of the *DataFrame* shows the actual unique values of this variable, after sorting them in ascending order: [20.0, 50.0, 150.0, 500.0, 1000.0, 1500.0, 2000.0, 2500.0, 3000.0]. Based on this result and taking into consideration that the feed rate is a parameter of the CNC machine which is set prior to its operation, it can be stated that it is a discrete variable instead of being continuous. Considering the frequency of each bin (seen in Figure 9) and the actual unique values in the bins, it seems like a feed rate below 500 (mm/min) is invalid, and valid feed rates are values between 500 and 3000 mm/min equally spaced with 500 as the difference between them**.** These specific multiples of 500 can reflect standard operating procedures or presets in the machine's programming. For example, certain tasks may require specific feed rates for optimal performance. It is also possible that different materials or tools require adjusting the feed rate to one of these multiples of 500. The nearly same frequency of the bins indicates that different tasks, materials or tools requiring different feed rates tend to occur with similar probability for one operation of the machine (one record in the dataset).

Skewness

Contrary to the suggestion derived from solely comparing the mean and the median from Table 1. (left-skewness), as per the discussion above the distribution does not exhibit a clear single skew like a typical left or right skew, instead, it seems to have several common feed rate values (nearly uniform distribution) that correspond to specific operational modes of the machine (multiples of 500 mm/min between 500 and 3000 mm/min).

Outliers

As discussed earlier, feed rates below 500 (mm/min) are invalid and indicate either data recording errors, data entry anomalies, or placeholder values that have not been properly cleaned. This indicates the need for data cleaning to address the concerned values.

Removing invalid values

According to the findings and conclusions, in the data cleaning process only samples with their *Feed* variable set to between 500 and 3000 are kept in the *DataFrame*, all other entries are removed.

Based on checking the relevant counts in the Python project, it can be stated that this time 3 observations were removed due to anomalies and missing values in the *Feed* variable, accounting for 1,4% of the whole dataset (226 observations), which is again not a significant loss of data.

After executing the data cleaning process, the distribution of the variable can be checked again in Figure 10.

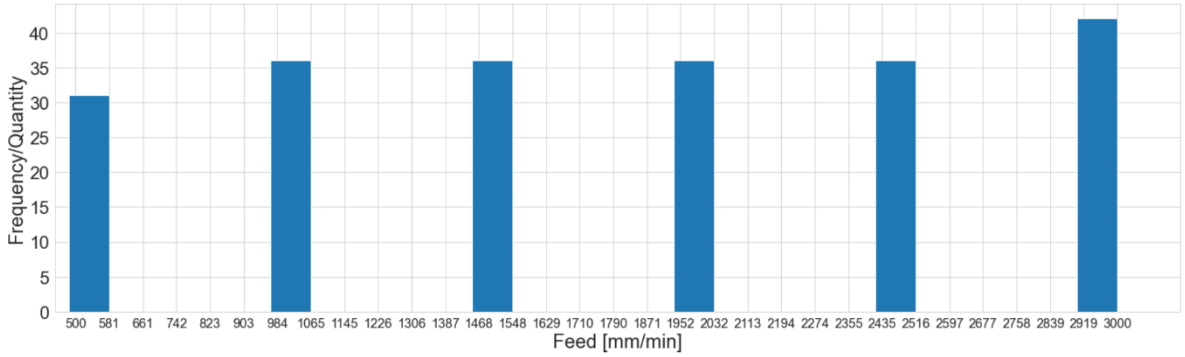

Figure 10: Distribution of the Feed variable after cleaning. Source: own editing.

In this chart it can be seen that the data cleaning was successful. It is also shown most of the tests were carried out with the highest feed rate setting of the CNC machine (highest speed), but there is no significant difference in the frequency of using different feed rates.

Path feature

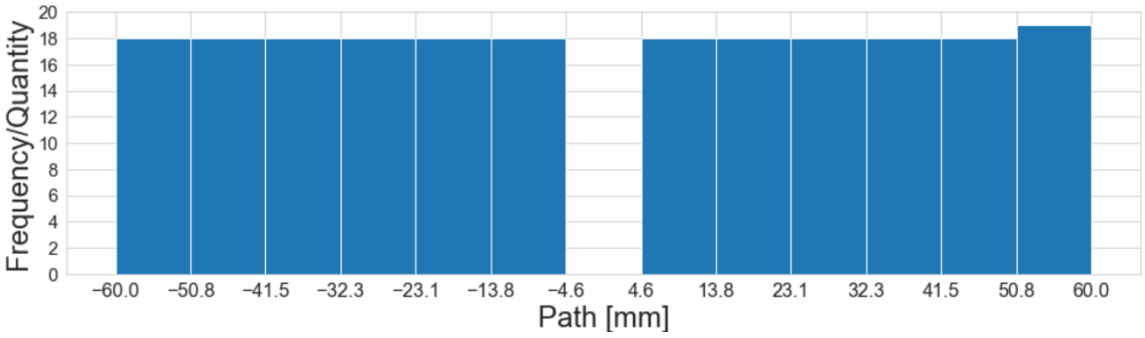

Figure 11: Distribution of the Path variable. Source: own editing.

The next feature to investigate the distribution of is the *Path.* In Figure 11. above a visual representation of this variable can be seen.

The values of this variable represent the distance travelled by the CNC machine's cutting tool during an operation, measured in millimetres (mm). More precisely, according to Figure 4, the *Path* values also indicate the direction of the movements along the particular axis that was used for the milling process. Therefore, the negative values are not indications of error, rather, they denote movement in opposite directions. For example, a positive value might indicate movement to the right (X+) or upward (Z+) direction, while a negative value might indicate movement to the left (X-) or downward (Z-) direction.

Regarding the shape of the distribution of this feature, the path values are uniformly distributed: there is no difference in the frequency of different *Path* values except for the last bin in Figure 11, where only a slight difference in the number of occurrences (19 instead of 18) can be seen (nearly uniform distribution).

Calling the *unique* function on the *Path* variable of the *DataFrame* shows the actual unique values of this variable, after sorting them in ascending order: [-60.0, -50.0, -40.0, -30.0, -20.0, -10.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0]. Based on this result and taking into consideration that the this feature is a parameter of the CNC machine set prior to its operation, it can be stated that it is a discrete variable instead of being continuous. Considering the frequency of values in the bins and the actual unique values present in the bins, it can be stated that all the values taken by this variable can qualify as valid values: the distance travelled by the CNC machine's cutting tool during an operation (one record in the dataset) can be a negative or positive multiple of 10 mm ranging from -60 mm to +60 mm, with equal likelihood (except for 60 mm) of being set for the path parameter of the machine for an operation. These specific multiples of 10 mm are likely to reflect standard operating procedures or presets in the machine's programming.

It is important to note that the *Path* variable can logically take on negative values due to the bidirectional nature of CNC operations. The uniform distribution seen is consistent with the CNC machine's same ability to move equal distances in positive and negative directions along its 3 axes.

Skewness

The distribution does not exhibit a clear single skew like a typical left or right skew. Instead, it seem to have multiples of 10 (between -60 and +60, excluding 0) as common distances travelled by the machine’s cutting tool, which are assumed to correspond to different operational modes of the machine.

Outliers

As discussed earlier, based on the histogram and the unique values, there is no indication of outliers for this variable.

Removing invalid values

No outliers were identified, but as stated earlier based on Table 1., there is 1 observation where the value for this variable is missing (a missing value is also considered invalid). This observation might have been removed when data cleaning was performed for the previous two features (in a lucky case the missing *Path* value was in the same row as a missing or outlier value for the *Axis* or *Feed*). The sum of missing values for this variable is checked in the Python project, and as the result is 0, it can be stated that there is no missing value in this variable anymore.

Because of the absence of outliers and missing values regarding this feature, performing data cleaning is not needed in case of this variable.

***Energy requirement* target variable**

Lastly, the Energy requirement [kJ] target variable is addressed. A visual representation of the distribution of this (dependent) variable can be seen in Figure 12 below.

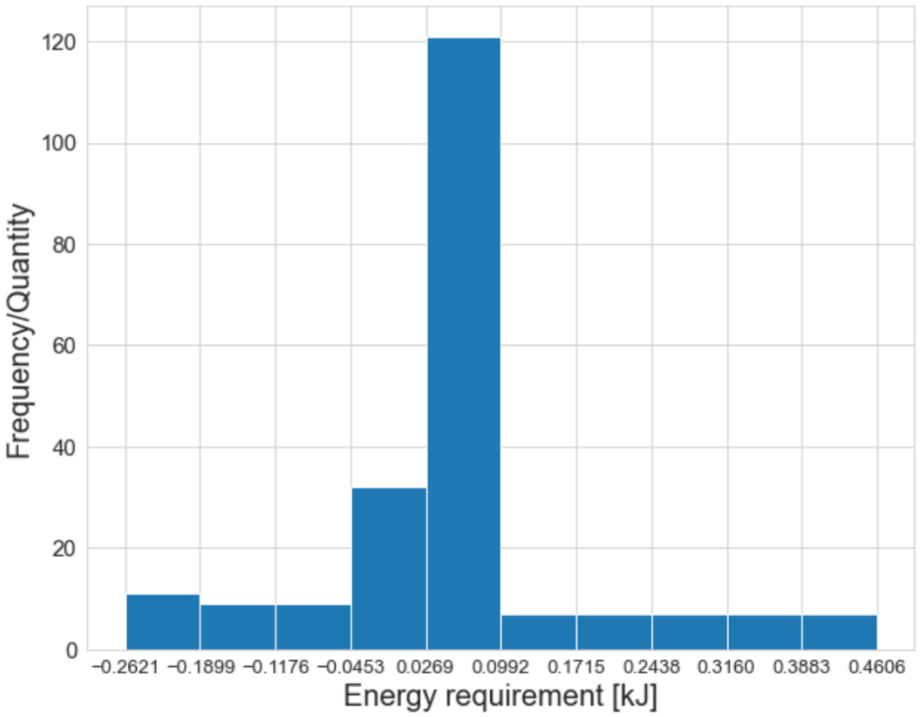

Figure 12: Distribution of the Energy requirement variable. Source: own editing.

The values of this variable represent the energy consumption of the CNC machine to execute an operation, or in other words execute the milling process. It is measured in kilojoules (kJ). There is no original information about negative energy requirement values in the dataset's description, but they are assumed to be present due to calibration and baseline shifts: the energy consumption measurement is most likely calibrated against a certain baseline (0 kJ), where the machine is expected to consume a certain amount of energy. Any operation (a milling process) that falls below this baseline is assumed to have been recorded as a negative value, indicating less-than-expected amount of energy consumption.

Regarding the shape of the distribution, the high frequency of Energy requirement values between 0.0269 kJ and 0.092 kJ (peak) suggests a high concentration of data points around this energy consumption/requirement value, meaning that it is common to consume about these kilojoules of Energy for a milling process or in other words for one operation of the machine. In statistical terms this peakedness is referred to as positive excess kurtosis (leptokurtosis), consequently the distribution of this variable is called leptokurtic. To measure kurtosis, Pearson's kurtosis measure is calculated. As for a normal distribution Pearson's kurtosis is 3 (known as mesokurtic distribution), the result of 4.7 for this variable confirms the leptokurtic

nature of the distribution. This indicates that milling processes typically require an amount of energy close to the central peak (between 0.0269 kJ and 0.092 kJ), but there are occasional operations of the machine that significantly differ in their energy requirements, potentially due to variations in the complexity or duration of the machining tasks.

Calling the *unique* function on the *Energy requirement* variable of the *DataFrame* shows the actual unique values of this variable, after sorting them in ascending order:

[-0.2621495, -0.2601395, -0.2547485, -0.2389, ……, 0.435702, 0.4423715, 0.4546375, 0.460567]. The maximum of the *value_counts* method returns 2, indicating that the maximum number of occurrences for the same values is 2. Based on this result, considering the distribution displayed in Figure 12, and the fact Energy requirement is the dependent or target variable suggests that it is a continuous variable, and all the values taken by this variable can qualify as valid energy requirement/consumption values.

Skewness

As it was seen before by the statistical overview (according to Table 1), the fact that the mean is higher than the median introduces a slight right-skewness for the distribution.

Outliers

As discussed earlier, based on the histogram and the unique values, there is no indication of outliers for this variable.

Removing invalid values

As no outliers were identified and there are no missing values for this variable (as seen earlier according to Table 1), it is not needed to perform data cleaning in case of the target variable.

Missing values across all the variables

Invalid values (outlier or missing) were removed first in case of the *Axis* variable, when only rows with a valid *Axis* value were kept in the *DataFrame*, thus rows with either an outlier or a missing *Axis* value were excluded from the dataset. Afterwards, the same invalid value removal was performed in case of the *Feed* variable too. As a result of removing invalid outliers, all the observations with a missing *Axis* or *Feed* value were dropped. Altogether 6 + 3 rows were removed out of the 226 original observations due to outliers.

In case of the *Path* variable, 1 observation was identified to have a missing *Path* value, but as it was checked before this row had already been removed when rows with invalid *Axis* and then invalid *Feed* values were dropped.

In case of the target variable, there was no identification of a missing value according to Table 1. Consequently, now it can be assumed that there are no missing values anymore in the dataset. To check this, the number of missing values in each column of the *DataFrame* is counted in the Python project.

```
Axis                0
Feed                0
Path                0
Energy_Requirement  0
dtype: int64
```

Figure 13: Count of missing values in each column of the DF. Source: own editing.

4.2. Target separation, train-test split, and feature scaling

Feature-target separation

Separating input variables from the output variable is a critical step in the data preprocessing phase of any machine learning project, pivotal to the success of predictive modelling (Kuhn & Johnson, 2013). This preprocessing procedure enables the model to establish a functional mapping from the features to the outcome of interest. The input variables or in other words the features *Axis*, *Feed*, *Path* are the independent variables of the dataset that provide the basis upon which the model will learn. The output variable Energy requirement, known as the target, is what the model aims to predict.

This segregation is vital for supervised learning, where the goal is to predict an outcome based on input data. The underlying foundation of feature-target separation is rooted in statistical learning theory, which states that the goal of predictive modelling is to approximate the unknown function that maps inputs to outputs. The efficacy of this approximation depends on the quality of the feature representation and the correctness of the target variable's alignment (James et al., 2013).

Isolating the target variable is paramount for the integrity of predictive models. It guards against data leakage, where information from the target variable could unintentionally be used by the model during training, leading to overly optimistic performance estimates (Kuhn & Johnson, 2013).

In practice, the process of separating features from the target variable means slicing the dataset into a matrix (as there are multiple features) of features (X) and a vector of outcomes (Y). At this point in the Python project the dataset with only valid values is given as a *DataFrame* object including both the features and the target variable. The goal is to remove the column from the *DataFrame* that contains the target variable, which is done by "dropping" the column,

effectively creating a new dataset that includes only the features. In the *drop* method in order to specify that the operation should be performed on columns rather than rows, the *axis* parameter is set to 1. After dropping the target variable column, what remains is stored as a new collection of data that will serve as the input for the machine learning model in form of a new *Pandas DataFrame* object (matrix of features, *X_multi*). Separately, the column that was dropped — the target variable, representing the energy requirement— is saved on its own in form of a new *Pandas* Series (vector of outcomes, *y_target*). This becomes the set of values that the model will be trained to predict.

Dividing dataset into training and test sets

The process of allocating portions of the dataset for training and testing is another fundamental step in the data preprocessing phase of developing a robust machine learning model. It is crucial for evaluating the true predictive performance of the model. The training set is utilized to teach the model, allowing it to recognize or uncover patterns and relationships within the data. The test set, conversely, is used to evaluate the model's predictive performance on unseen data, providing an unbiased evaluation of its generalization capabilities to new, out-of-sample observations (James et al., 2013). The separation into different datasets also helps to mitigate overfitting, which happens when a model learns the training data too well including noise and outliers, to the detriment of its performance on new data.

For the practice of splitting data, researchers stress the importance of using a large enough training set to capture the full complexity of the data while ensuring the test set is representative of the general problem space (Kang & Tian, 2018). The size of the training and test sets can vary depending on the total amount of data available and the specific requirements of the modelling process. Commonly, a larger portion is allocated to training to provide the model with the broadest possible learning base, but at the same time it is ensured that enough data is reserved for an unbiased evaluation of the model's performance. A common split ratio is 80%-70% for training and 20%-30% for testing, but as discussed, this can change based on the number of observations in the initial dataset. Researchers have proposed various strategies for splitting the dataset, often recommending a randomized approach to avoid potential biases that could be introduced by non-random sampling. Furthermore, it is stated that while larger training sets generally lead to better-performing models, the marginal benefit decreases as the amount of available data increases, and retaining a sufficient test set is crucial for reliable performance estimation (Hastie et al., 2009). For very large datasets a sufficient test set might mean a smaller proportion of the data (e.g., 10% or 20%), while for smaller datasets a 70/30 or even a 60/40

split might be used to ensure enough test data is available. Ultimately, the aim is to achieve a balance that maximizes learning while ensuring a robust evaluation.

Regarding the Python project, at this point a *DataFrame* is given containing the features as the input for the machine learning model (matrix of features, *X_multi*), and there is also a Series containing the target variable (vector of outcomes, *y_target*). These two Pandas objects are the input for *train_test_split* function from *sklearn.model_selection*.

This built-in function executes the division randomly to prevent any unintentional bias that might influence the model's learning process. However, a specific random state or seed is set as a parameter of the function, which allows this random process to be replicated exactly in the future. By fixing the seed for the random number generator, it is ensured that the split is consistent across different runs of the experiment, which is essential for scientific rigor and for comparing model performance across different runs or by different users.

As the dataset for the energy requirement prediction problem is considered a smaller dataset (217 observations in the cleaned dataset), with the parameters of the function 30% of the data is set to be allocated to the test set, and the remaining 70% to the training set. As it is checked in the Python project, after the separation 66 observations are available for testing model performance.

After executing the function with these inputs, four separate collections of data are returned: one set of features as a *DataFrame* for training (*X_train*), one set of corresponding target variables as a Series for training (*y_train*). For testing, another set of features as a *DataFrame* is returned serving as 'unseen' data (*X_test*), and also its corresponding set of target variables is returned as a Series, allowing to compare predicted target values to the true ones (*y_test*).

Scaling of the features

Feature scaling is a crucial step in the preprocessing of data for machine learning models. It means transforming all the features to the same level of magnitude in order to avoid features with larger scales dominating those with smaller ones resulting in biased model outcomes. Applying feature scaling leads to improved training stability and better predictive performance of the models (Navlani et al., 2021) Eliminating the disproportionate influence of features with larger scales is particularly beneficial for algorithms that calculate distances between data points, such as Support Vector Machines (and consequently Support Vector Regression, as an extension of SVMs).

One common method of feature scaling is standardization (Z-score normalization), which is used in the scope of this study and entails transforming the data of each feature to have a mean

of 0 and a standard deviation of 1. Mathematically, this is achieved by subtracting the mean of the respective feature from each value and then dividing the result by the standard deviation of that feature.

This procedure is implemented by *sklearn*'s *StandardScaler* class, which is used in the Python project, where firstly the class is imported and instantiated, and the original training set is also saved to be able to check later the actual scales in the training set to verify Random Forest's training as it will be discussed in section 5.2 (Systematic discussion of the utilised models – Random Forest Regression – Application to the energy prediction problem). After these steps, features in the training set are scaled by "fit transforming" the *StandardScaler* object on the training set, meaning that the mentioned mathematical procedure is carried out. Then, in order to prevent any information from the test set leaking into the model during training (into the training set), features in the test set are scaled using the mean and standard deviation values that were computed based on the training set (during the fitting part of "fit transforming") (García et al., 2015). This is done by only transforming (and not "fit transforming") the *StandardScaler* object on the test set.

To check how well the test set represents the training set, descriptive statistics are generated for both of them, which can be seen below in Table 2. (train) and Table 3. (test).

	Axis	Feed	Path
count	1.510000e+02	1.510000e+02	1.510000e+02
mean	-1.176395e-17	-5.881976e-17	-3.088038e-17
std	1.003328e+00	1.003328e+00	1.003328e+00
min	-1.281465e+00	-1.537218e+00	-1.571815e+00
max	1.167917e+00	1.393589e+00	1.401179e+00

Table 2: Descriptive statistics of the training set after feature scaling. Source: own editing.

	Axis	Feed	Path
count	66.000000	66.000000	66.000000
mean	0.017450	0.008117	-0.257992
std	1.004839	1.010755	0.870216
min	-1.281465	-1.537218	-1.571815
max	1.167917	1.393589	1.401179

Table 3: Descriptive statistics of the test set after feature scaling. Source: own editing.

It can be seen that not only in case of the training set, but also for the test set the mean and standard deviation of the features are quite close to the desired 0 and 1 values (respectively) after using the means and standard deviations only from the train set for scaling both sets. A slight deviation is noticeable only in case of the *Path* feature (-0.258 mean, 0.8702 standard deviation), but it is not substantial. According to these findings to the comparison of the minimum and maximum values in training and test sets indicating the same range of values, it can be seen that values in the test set are well-representative of those in the training set.

5. Model training and the utilised models

5.1. The process of training and making predictions

At this stage the models are to be trained, which means allowing them to recognize, uncover and learn patterns and relationships within part of the data that was allocated to training. More precisely, during the procedure of training, models iteratively adjust (by using algorithms like gradient descent) their internal parameters (such as coefficients or weights and intercept) to minimize a function that measures the discrepancy between actual values and predictions made with the parameters (loss function). Next, predictions with the trained models are to be made both on the train and test sets in order to be able to compare predictions with actual target variable values allowing to evaluate the model's predictive performance.

This process can be implemented in the Python project in an iterative way for each model:

- Import the model's class from *sklearn*'s corresponding library.
- Initialize a new instance of the imported class as the Python object of the model.
- Provide the model with the training part of the feature data (*X_train*) and the target data (*y_train*) as inputs to the model's *fit* method so that it learns to establish a relationship between them.
- Make predictions with the trained model on the training data using the model's *predict* method in order to evaluate how well the model has learned to capture the patterns and relationships within the seen data, or in other words how well it has learned to approximate the underlying function within the training data that maps the input features to the output.
- Make predictions with the trained model on the test data using the model's *predict* method in order to evaluate the model's predictive performance on unseen data, which provides an unbiased evaluation of its generalization capabilities to new, out-of-sample observations.

5.2. Systematic discussion of the utilised models

Linear Regression

Overview

Linear regression is one of the simplest and most commonly used statistical techniques for predictive modelling. This model operates by using a linear method to model the connection between a dependent variable and one or more independent variables. It fits a linear line (or a linear hyperplane in case of multiple explanatory variables) to the data, meaning that the change in the dependent variable is modelled to be proportional to the change in the independent(s) (Yan & Su, 2009).

Mathematical representation

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \ + \beta_n x_n + \ \epsilon$$

, where y is the dependent variable, β_0 is the y-intercept, β_1 ... β_n are the coefficients that represent the weight of each independent variable, x_1 ... x_n are the independent variables, and ϵ is the error term that captures any variation in y not explained by the independent variables (Gujarati, 2019).

Advantages

The strengths of this model (Seber & Lee, 2012) are its simplicity, interpretability, and speed of computation. It performs well when the data truly has a linear shape, without too many extreme values or outliers. It also serves as a good baseline model for a comparative analysis of different models.

Limitations

The relationship between the dependent and independent variables is assumed to be linear. This premise of a straight-line relationship is usually false, leading to severe underfitting.

This model is also sensitive to outliers, which can significantly influence the fitted parameters. As linear regression aims to minimize the sum of the squared differences (its loss function) between the observed values and the model's predictions (squared residuals), even a single outlier with a large residual (large difference from the actual or the observed value) can disproportionately affect the overall model. This is because squaring the residuals (the differences) gives more weight to larger differences. Consequently, the regression line (the model's predictions) may be pulled towards the outlier, leading to a less accurate representation of the true relationship for the rest of the data.

Linear regression assumes the data exhibits homoscedasticity, meaning the residuals are equally distributed across the regression line (errors are just random fluctuations around the true line). Specifically, homoscedasticity is the situation where the error terms or residuals (the differences between the observed values and the values predicted by the model) have constant variance across all levels of the independent variable(s). Hence, on a residual plot regardless of the predicted value (horizontal axis), the spread or scatter of the residuals should remain consistent. Residuals are also assumed to be independent of each other.

Observations as well are assumed to be independent of one another, plus the predictor variables (features) are assumed to be not too highly correlated with each other (no multicollinearity).

Additionally, the linear model assumes that the residuals follow a normal distribution with a mean of zero, however this assumption does not apply in case all the other mentioned assumptions are fulfilled and if the dataset is in statistical terms large enough (has more than about 100 observations).

These assumptions and conditions (Hoffmann, 2021) must be met for the best results, and when they are violated, the model's predictive accuracy can be significantly reduced.

Application to the energy prediction problem

In the context of the addressed manufacturing problem, by applying linear regression it can be seen how effective it is (in terms of predictive accuracy) to fit a linear relationship between the energy requirement and the independent variables. Thus, in spite of its limitations, it provides a valuable benchmark for comparing the performance of algorithms that are able to model more complex relationships, and through this it can be examined if non-linear relationships are present and necessary to capture in case of the addressed problem.

Random Forest Regression

Overview

Random Forest Regression is an ensemble learning method belonging to the ensemble learning family, where the collective decisions from various models are pooled to improve the overall result. This model operates by constructing multiple decision trees during training, and outputs the mean prediction of the individual trees (or the mode of the classes for classification tasks) (Ogunleye, 2022).

Mathematical representation

$$y = \frac{1}{n}\sum_{i=1}^{n} y_i^{(tree)}$$

, where y is the dependent variable, n is the number of trees, and $y_i^{(tree)}$ is the prediction of the i-th tree.

Each decision tree is constructed from a random subset of the training set. The predictions of the individual trees doesn't have a straightforward mathematical equation like linear regression due to the complexity of the tree structures. During the construction of the trees, a random subset of features is chosen for splitting at each node (Breiman, 2001).

Advantages

The main advantages of random forest regression include its ability to handle non-linear relationships between the dependent and independent variables effectively.

Another advantage is that it is less likely to overfit than a single decision tree because it averages multiple trees to reduce the model's variance and is less prone to overfitting and outliers in general as well. This characteristic of the model is also ensured by the random subset of features for each node and the random subset of the training set when constructing trees (Schonlau & Zou, 2020).

Next, the aggregation of predictions from a multitude of trees, often referred to as "bagging," helps to improve prediction accuracy too.

In addition, this model is able to handle well large datasets with higher dimensionality (higher number of features).

Limitations

This model requires more computational power and resources (leading to longer training periods as well) than simpler ones, especially as the number of trees increases and when used on large datasets.

They are also less interpretable and less intuitive, as understanding the collective decision-making process of hundreds of trees can be challenging (Brieuc et al., 2018). Although variable importance can easily be determined (examine how many times in total a variable is used in the model for splitting (at how many nodes), and how much in total this reduces the loss function), it is not easily feasible to calculate the specific marginal impact of an individual explanatory variable (how a ceteris paribus change in the variable affects the estimated value of the target variable).

Lastly, this model has limited extrapolation capabilities, meaning that it can make reliable predictions only within the range of the feature values it has seen during training in the training data.

Application to the energy prediction problem

By modelling the energy consumption through Random Forest, it can be assessed whether an ensemble model that is more robust to overfitting, less sensitive to outliers and is able to capture non-linear relationships can significantly improve the predictive performance.

Concerning the limited extrapolation capabilities of this model, the widest possible range of features values come from fixed settings (parameters) of the CNC machine. At this point, before importing the regressor's class it is checked whether the range of feature values in the training set is representative of the actual range of feature values in the entire dataset. As they turn out to be identical, it is ensured that the random forest model is trained on all possible values of the features (it has seen all possible settings of the machine). Therefore, limited extrapolation does not mean an issue when using the already trained model on new data (new settings of the machine parameters) to predict the energy requirement, as the possible settings of the investigated machine (possible ranges of the parameters) are the same in case of any milling process.

Within the framework of this thesis, the default hyperparameter settings of the *sklearn* SVR model are not modified. Hyperparameter tuning, so the process of finding the best combination of hyperparameters such as the number of decision trees in the forest, number of randomly selected variables and observations for a decision tree (applied to all the decision trees in the forest), or the minimum number of observations required on a leaf (last node) of a decision tree is subject to further research (as mentioned later in section 10).

Support Vector Regression

Overview

Support Vector Regression (SVR) is a learning method that belongs to the family of Support Vector Machines (SVMs). SVMs are primarily used for classification tasks, but SVR can be applied to solve regression problems, as it is designed to predict a continuous value fitting the best line (in two dimensions) or hyperplane (in higher dimensions) within a certain threshold margin called ε-insensitive tube, which is centered on the line or hyperplane. This model operates by finding a function where predictions deviate from the actual targets by no more than ε for all training data instances. Data points outside of this margin (ε) contribute to the prediction error and are identified as support vectors since they help construct the ε-insensitive zone (or tube). When defining this function, it is also an objective of SVR to ensure it is as generalized (as "smooth") as possible (Basak et al., 2007).

Mathematical representation

$$y = w * x + b$$

$$\min \frac{1}{2}\left|\left|w\right|\right|^2 + C * \sum_{i=1}^{n} \xi_i^{(+)} + \xi_i^{(-)}$$

Subject to:

$$y_i{}^{(true)} - y_i{}^{(predicted)} \leq \varepsilon + \xi_i^{(+)}$$

$$y_i{}^{(predicted)} - y_i{}^{(true)} \leq \varepsilon + \xi_i^{(-)}$$

$$\xi_i^{(+)}, \xi_i^{(-)} \geq 0$$

, where y is the dependent variable, w represents the weights for the independent variable(s) denoted by x, and b is the bias term.

C is the penalty parameter controlling the strength of the regularization (controls the trade-off between achieving a low error on the training data and minimizing the norm of the weights), n is the number of observations, $\xi_i^{(+)}, \xi_i^{(-)}$ are slack variables for data points outside the ε-tube (support vectors) representing the distance from the actual values to the edges of the tube (for data points inside the tube $\xi_i = 0$) (Smola & Schölkopf, 2004).

$y_i{}^{(true)}$ is the actual value and $y_i{}^{(predicted)}$ is the estimated value for the dependent variable, ε is the margin of tolerance (maximum allowed error).

Advantages

One of the main advantages of this machine learning algorithm is its effectiveness in modelling complex relationships due to the use of kernel functions. This is often referred to as the kernel trick, which involves mapping input features into higher-dimensional feature spaces to make it easier to fit a regression function (or in case of classification tasks easier to find a hyperplane that separates the data points into different classes). By projecting the data into a higher-dimensional space, more complex patterns or relationships that may exist within the data become apparent and separable by a linear regression curve (or by a hyperplane for classification).

SVR is especially beneficial when there is a clear margin of separation between data points, because when data points are well-separated it can effectively identify a line or hyperplane that explains the majority of the variance without being influenced by outliers or noise. In other words, SVR is particularly effective if the input features that define the target variable yield outcome values (targets) that are clearly spaced apart (without much overlap or noise), as in this case it can determine a predictive model that accurately captures the underlying patterns without being influenced by outliers (Hamel, 2009).

The advantages of SVR also include its ability to minimize the empirical risk within the ε insensitivity zone (ε-tube) providing a unique solution, as it is based on structural risk minimization principle, which reduces an upper bound of the generalization error, not the training error.

Limitations

One of the limitations of SVR is for example the choice of an appropriate kernel function, or the tuning of hyperparameters like C (penalty parameter) and ε (margin of tolerance), which can be challenging and computationally intensive. Hyperparameter tuning (finding the best combination of hyperparameter values) is subject to further research (as mentioned later in section 10), since within the framework of this thesis the default *sklearn* hyperparameter settings are not modified.

Limitations also include that higher dimensionality can lead to issues such as the curse of dimensionality, where the feature space becomes so large that the available data becomes sparse. In practical terms, sparsity means points that may seem close in lower dimensions can become very far apart in the high-dimensional space (Liang et al., 2011). This can make it difficult for the model to learn effectively from the data, as there might not be enough data points to create a generalizable pattern, which results in an increased risk of overfitting.

In addition, due to the use of kernel functions (except for the linear kernel), interpreting the influence of individual features on the estimated value becomes less straightforward as the original features are combined in complex ways. Therefore, there is no simple coefficient provided for each feature, and determining the importance of each feature directly from the adjusted weights is not feasible. To assess how changes in input features affect the predicted output (subject to further research), techniques such as sensitivity analysis or permutation importance can be used.

Lastly, utilising this machine learning algorithm involves solving a quadratic programming problem, thus the computational complexity can be higher.

Application to the energy prediction problem

Taking into account that SVR can perform effectively under the assumption of a clear margin of separation between data points (as discussed under the advantages of the model), contrary to the second hypothesis (states that SVR will not substantially underperform RF) a weaker performance of SVR can be expected in case of this problem: as it was seen in Figure 12, the distribution of energy requirements is leptokurtic (high concentration of values around a certain range of the energy requirement), thus for most of the data points this assumption is violated as they are not well-separated. However, in the higher-dimensional feature space data points might become more spaced apart leading to linearly separable patterns and better predictive performance, which is examined later in section 6.3. (Performance comparison: visual analysis – Linear Regression).

As the use of the kernel trick involves projecting data into the mentioned higher-dimensional feature space, by comparing SVR's performance with the other two models', it can be seen if such an increase in complexity results in more accurate predictions of energy consumption for the milling machine.

Examining whether the SVR's capacity to model more complex relationships translates to an improvement in predictions is beneficial as it can be understood if such relationships and pattern are present in the addressed energy prediction problem.

6. Performance comparison: visual analysis

Residual plots and the benefits of using them is discussed in section 2.4 (Applied methodologies for data collection and analysis).

6.1. Linear Regression

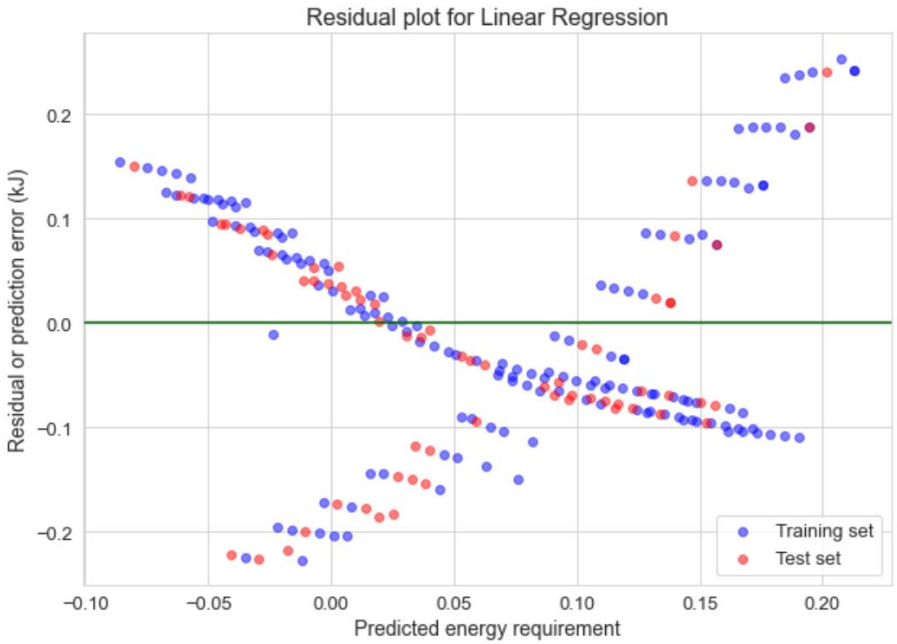

Figure 14: Residual plot for Linear Regression. Source: own editing.

In Figure 14. the residual plot generated by applying Linear Regression to the addressed problem is shown. It can be seen that the residuals do not appear to be randomly distributed around the horizontal line at zero (green line in Figure 14.), but instead show a clear pattern: a kind of "X"-shaped pattern where as the predicted energy requirement values increase, the spread of residuals first decrease and then increase (as predictions increase, both overestimating and underestimating first reduces, and then intensifies). This observed "behaviour" of residuals suggests the presence of heteroscedasticity as the variance of the residuals is not constant across all levels of predicted values. It means homoscedasticity as one of the core assumptions is violated, plus other assumptions of the linear model are also violated such as independent and normally distributed residuals, leading to strongly impaired predictive capabilities of the applied model.

Additionally, the occurrence of a pattern indicates that the relationship(s) between the axis, feed, path settings of the machine and its energy requirement (for the milling process determined by these settings) is non-linear and is not being captured by the model (as it assumes

a straight-line relationship). Thus, linear regression proves not to be the most suitable for modelling the complexity of the underlying relationship(s).

It is worth noting that according to Figure 12. the range of actual energy requirement values is from -0.26 to 0.46 kilojoules, however in Figure 14. predictions are ranging from only about -0.08 to 0.22 kJs indicating that Linear Regression fails to predict “in” the whole range of energy consumptions.

The fact that the distribution of the blue and of the red points (representing predictions errors on the train and test sets respectively) both follow similar patterns suggests that the model performs likewise on unseen data compared to the data it was trained on, which indicates that it was not overfitted to the training set during training.

Overall, the poor predictive performance of Linear Regression seen in the residual plot suggests that a more complex model might be more appropriate for the addressed manufacturing problem that is able to interact with the features in a more sophisticated way potentially leading to better predictive accuracy.

6.2. Random Forest Regression

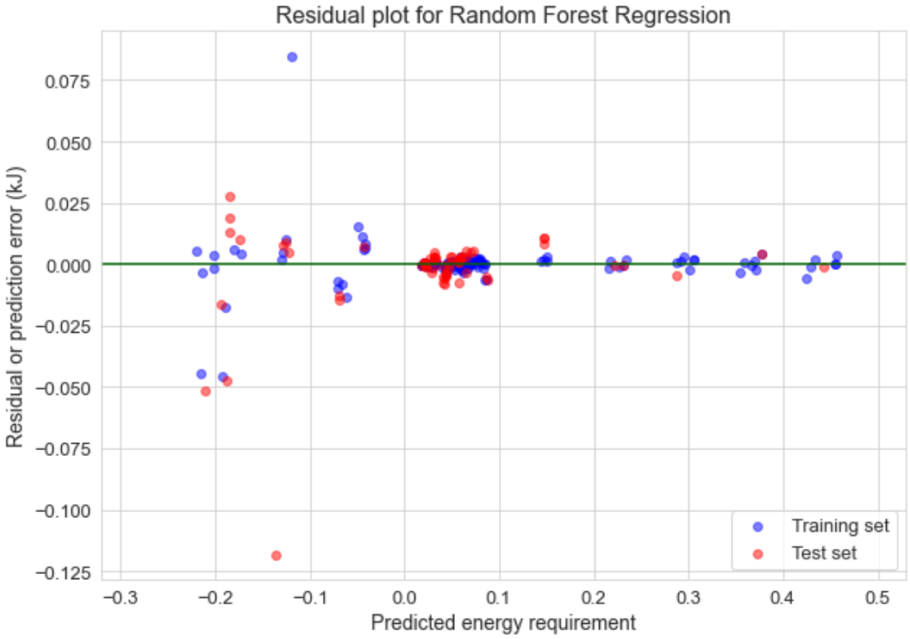

Figure 15: Residual plot for Random Forest Regression. Source: own editing.

Investigating Figure 15, it can be seen that the vast majority of the residuals (errors) obtained by employing Random Forest Regression to the addressed manufacturing problem are centered quite closely around the zero line suggesting that the model has no significant bias in most predictions, as there are no substantial overestimations or underestimations. Those few cases

when the residuals are larger occur only when the predicted energy consumption value is below zero, indicating larger prediction errors when the energy consumption of the milling machine is estimated to be less-than-expected (negative values indicate such an energy consumption that is below the baseline consumption, as discussed in section 4.1).

Comparing to the previous Linear Regression model, there is a much tighter clustering of residuals around the zero error line indicating that this model is accounts better for the variance in the data. The tight spread of residuals (except for the few predictions with higher errors) suggests that it handles well outstandingly low or high values, which is consistent with the expected behaviour of an ensemble method discussed in section 5.2. (Systematic discussion of the utilised models – Random Forest Regression – Advantages).

Based on the absence of any clear pattern or systematic structure in the residuals, it can be stated that it has effectively captured the non-linear relationships and the actual underlying structure of the data. Comparing this plot to Linear Regression's residual plot, it is apparent that capturing non-linear relationships is indeed necessary for the addressed problem.

Except for those few larger residuals occurring only when the predicted energy consumption is low (forecasting below zero, thus below baseline consumption), errors here do not show a clear increasing or decreasing pattern in their variance with the rise of the predicted energy requirement (no heteroscedasticity), which aligns with the expectations against the residual plot of a well-fitting model as discussed in section 2.4 (Applied methodologies for data collection and analysis).

Since the range of predicted values (from about -0.22 to 0.46 kilojoules) appears to be consistent with the range of the observed energy requirements in Figure 12 (from about -0.26 to 0.46 kilojoules), it is seen that the model's limited extrapolation capabilities do not pose a real restriction for this particular manufacturing problem, as discussed in detail in section 5.2. (Systematic discussion of the utilised models – Random Forest Regression – Application to the energy prediction problem).

Both blue and red points are close to the horizontal line, but the red ones (representing errors in predictions made on the test set) show more variance than the blues (training residuals). This is expected as on unseen data the model will naturally perform slightly worse.

Additionally, the fact that not only in case of the training set but also in case of the test set, those few predictions that have higher errors (occurs only when the prediction is a negative value) seem to be randomly distributed around the horizontal line indicates good generalization (no overfitting). This is in line with one of the advantages of Random Forest Regression

(discussed in detail in section 5.2), where the averaging over many trees typically results in a model that is not overfitted.

Overall, the Random Forest model appears to be an appropriate choice for predicting the energy consumption of the investigated machine's milling processes.

6.3. Support Vector Regression

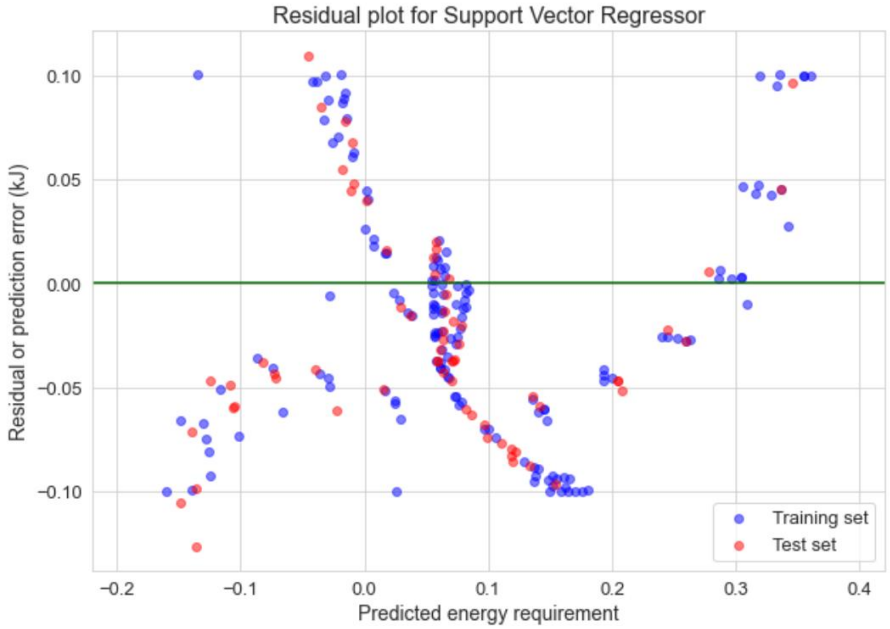

Figure 16: Residual plot for Support Vector Regression. Source: own editing.

In Figure 16. the residual plot of Support Vector Regression can be seen, where the distribution of residuals shows a wider spread than observed with the previous ensemble model (comparing to Figure 15), indicating SVR's difficulty in capturing properly the variance in the data. The spread of errors does not seem to be random around the zero line, rather patterns of residual points can be identified suggesting that SVR is missing (modelling differently) some aspect of the data's structure, potentially the relationship(s) that Random Forest Regression managed to capture properly.

This might be due to the nature of this particular regression task, which may not benefit as much from the potential offered by the kernel trick. Projecting the data into a higher-dimensional space does not prove to be a proper approach to modelling the underlying relationships.

The distribution of the blue and of the red points both follow similar patterns, which suggests that the model performs similarly both on training and unseen data indicating no overfitting.

The expectation for a low predictive performance (detailed in section 5.2. Systematic discussion of the utilised models – Support Vector Regression – Application to the energy prediction

problem) derived from not well-separated energy requirement values (leptokurtic distribution of the target variable) is met. The poor performance also indicates that when projecting the data into a higher-dimensional space data points did not become more spaced apart in that higher-dimensional feature space (relationships or patterns did not become linearly separable in the higher-dimensional space).

The impaired predictive performance is also suggested by the fact that the variance of the residuals is not constant across all levels of the predicted values (heteroscedasticity).

Additionally, in Figure 16. predictions are ranging from about -0.15 to 0.35 kJs, however, according to Figure 12, the range of actual energy requirement values is from -0.26 to 0.46 kilojoules. This indicates that the models fails to predict "in" the whole range of energy consumptions as it was observed at Linear Regression's residual plot as well.

Overall, the SVR model results in weaker predictive accuracy of the milling machine's energy consumption, and therefore is not an appropriate choice for the addressed problem.

6.4. Residuals side-by-side and in the same plot

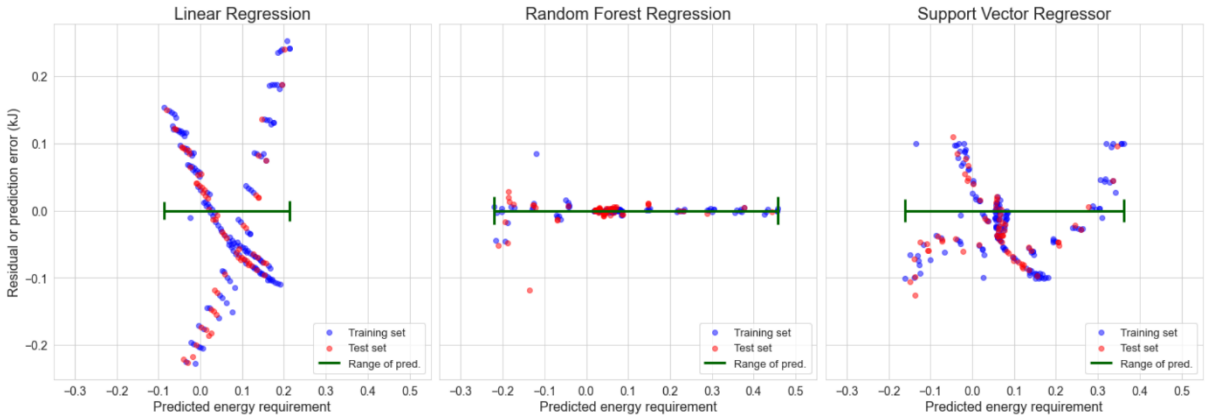

Figure 17: Residual plots for the utilised models side-by-side. Source: own editing.

In Figure 17. the residual plots for all three applied models have the same ticks (and same range) both on the x-axis and y-axis. Comparing the range of the predictions in each plot (length of the green lines), and considering the actual range of energy requirements based on Figure 12. (from -0.26 to 0.46 kilojoules), it can be stated that only Random Forest Regression is able to predict "in" (almost) the whole range of actual values, which suggests superior predictive performance compared to the other two models.

According to the residual plot of the other two models in Figure 17, it can be seen that neither is able to accurately predict energy requirements that are closer to the limits of the actual range. Comparing SVR's residual plot to Linear Regression's, it can be seen that SVR predicts with a

wider range and thus with smaller deviations from the actual values (predicted range is closer to the actual range). In contrast, it is visible that some of Linear Regression's prediction errors are higher as they are further away from the green horizontal line than in case of SVR. It is also notable that for SVR the prediction error data points are clustered between about -0.05 and 0 kJs. However, for Linear Regression a cluster of errors (with two distinct "branches") is seen at a worse (further away from the green line) level of error (between about -0.05 and -0.1 kJs). The negative value indicates common overestimations for both models.

The observations related to the comparison between Linear Regression's and SVR's residual plot suggests that the latter model has a better predictive performance than the baseline linear model for the addressed manufacturing problem ("middle performance" out of the three models), indicating that it did capture some of the non-linear patterns in the data, but not nearly as effectively and accurately as the Random Forest Regression model. The determination of how much better SVR performs compared to Linear Regression and how much worse than Random Forest Regression is carried out through the numerical performance comparison in the next section.

The previously established finding that only in case of Random Forest Regression there is no visible pattern of residuals (except for the "pattern" that below zero (below baseline) energy consumption predictions yield a bit larger errors, which are still significantly smaller than for the other two models) is also clearly visible again in Figure 17. It is indicating that only this model was able to properly capture the underlying relationships in the data.

Taking into account the leptokurtic nature of the energy requirement's distribution according to Figure 12. (showing high concentration of energy requirement values around 0.027 and 0.092 kilojoules) can also provide a valuable insight into Random Forest's performance. It can be seen in Figure 17. that this model is able to predict accurately or with small errors the energy requirements of the machine not only in this common range, but also the higher and lower consumptions (in case of negative values the prediction errors are a bit larger). On the other hand, the other two models' most significant predictions errors are made for energy consumptions outside of the common or typical consumption range.

In conclusion, in Figure 17. the characteristics of the residuals on the three scatter plots confirms that the ensemble model is the most capable of capturing the underlying patterns and relationships in the data. Thus, through a visual examination of model performance, Random Forest Regression is stated to be the most effective in providing accurate predictions for the energy requirement of the milling machine. The visual examination also suggests that its

superior performance is followed by SVR's ("middle") performance, and that the linear model performs the worst.

The result can be observed again below in Figure 18. displaying test residuals in case of all the three models in the same plot.

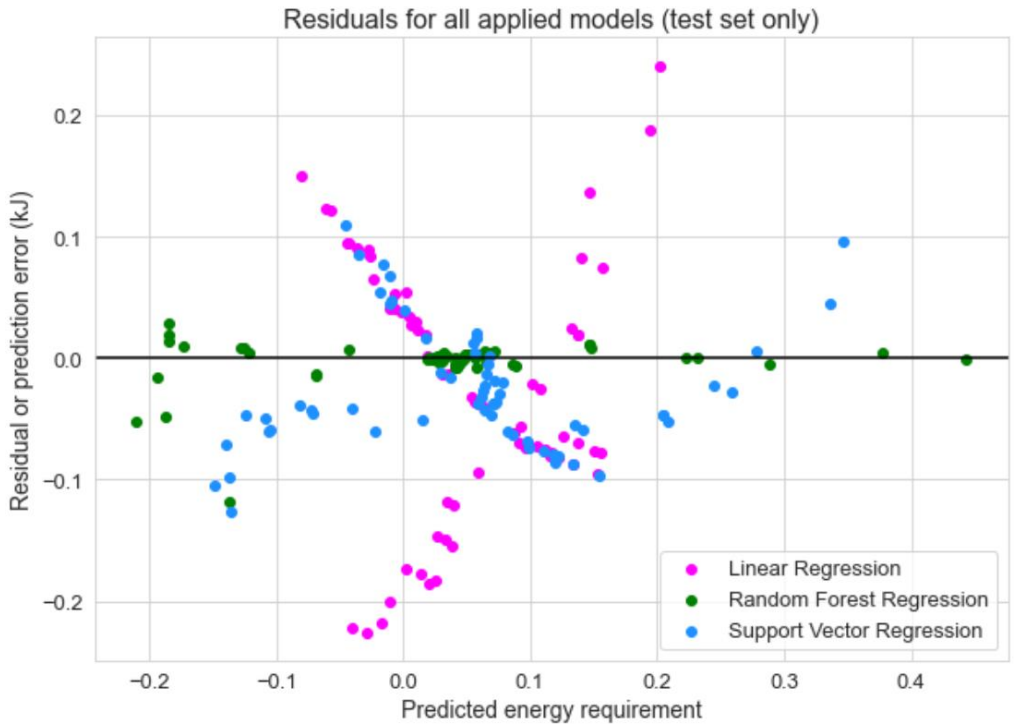

Figure 18: Test set residuals of the utilised models in one plot. Source: own editing.

7. Performance comparison: numerical examination

The Mean Square Error (MSE) and Mean Absolute Error (MAE) regression evaluation metrics are discussed in detail in section 2.4 (Applied methodologies for data collection and analysis).

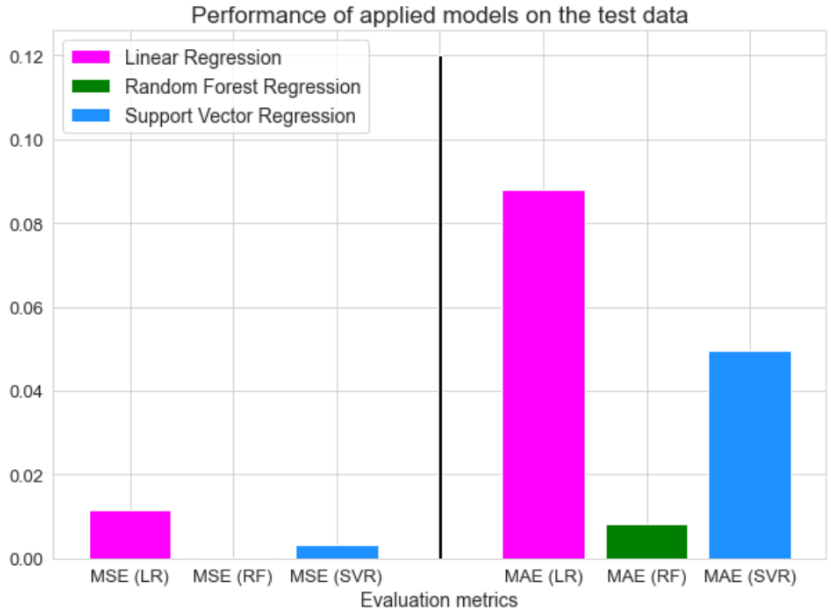

Figure 19: MSE and MAE metrics of the utilised models. Source: own editing.

In Figure 19. MSE values are lower than MAE values as deviations from the actual values for all predictions are below 1 according to for example Figure 17, and squaring a value below 1 results in a smaller value. In case of the Random Forest model, MSE is so low (0.00033) that it is not visible.

The suggestion established by the visual performance comparison (conducted in the previous section) about SVR's better predictive performance ("middle" performance) than Linear Regression's is supported by both lower MSE and MAE values of the more complex model.

The error measure in case of MAE can be interpreted in the same unit as the data, thus it can be said that for example Support Vector Regression's predictions deviate from the actual values on average by 0.05 kilojoules, irrespective of the direction of the deviation (regardless of whether the error comes from overestimation or underestimation).

The result which was derived through the visual examination showing that Random Forest Regression provides the most accurate predictions for the energy requirement of the milling machine is confirmed by the numerical performance investigation, since this model has outstandingly the lowest MSE and MAE values compared to the other two models.

8. Behind the performance: relationships in the data

In this section the relationship between the features and the energy consumption target variable is investigated to get a better understanding of the predictive performance of the utilised models. When examining the figures in this section, it is worth keeping in mind that that according to Figure 8, among the collected observations, the machine's cutting tool was set to move along the 3 different axes X, Y and Z. In the examined dataset, movements are distributed more or less uniformly along the three axes (70-80 times for each axis).

8.1. Energy consumption versus Axis

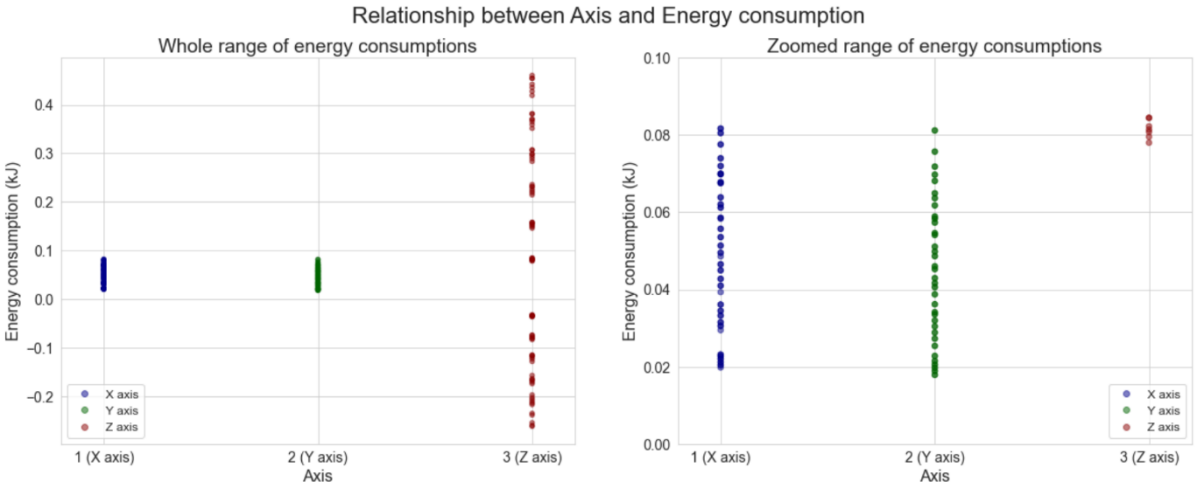

Figure 20: Energy consumption vs the Axis feature scatter plots. Source: own editing.

According to Figure 20, energy consumption shows the same range of values when the machine's cutting tool was moving along the X and Y axis during the milling process. This indicates that milling processes (in case of the investigated machine) requires or consumes the same range (amounts) of energy regardless of whether the machine (more precisely, the cutting tool of the machine) was set to move left-right (X axis) or forward-backward (Y axis) before starting the machining process. This typical range means energy consumptions between about 0.02 and 0.08 kJ according to the "zoomed-range" scatter plot in Figure 20. This range aligns with the most frequent bin (range) of energy consumptions being between 0.0269 and 0.092 kJ in Figure 12, where the distribution of the energy consumption values is displayed.

In contrast, when the machine's cutting tool was set to move upward or downward (along the Z axis) for the milling process, the energy consumption varies a lot more in a much wider range of consumption values according to Figure 20. This leads to the conclusion that the machine consumed lower or higher energy amounts than the typical consumptions ("X and Y axis

consumptions") only when it was set to move along the Z axis. Thus, it can be stated that all the energy consumptions outside of the most frequent bin in Figure 12. are "coming from" milling processes where the Z axis was involved (and not the X or Y axis). This suggests that those milling tasks where the machine's cutting tool has to move up or down (along the Z axis) are generally either much more or much less energy-demanding when compared to machining tasks with horizontal or vertical movements.

The observed relationship between the *Axis* feature and the energy consumption supports Random Forest's superior predictive performance compared to Linear Regression and Support Vector Regression. Unlike the latter models, which attempt to derive a continuous function for the predictions, Random Forest (more precisely, the decision trees within the forest) builds a series of decision rules (Breiman, 2001) that can effectively handle segmented data scenarios such as a non-linear effect between the *Axis* feature and energy consumption. For instance, it can specifically model that if an *Axis* value is less than 3 (i.e., X or Y axis), then the energy consumption will range from about 0.02 to 0.08 kJ, while milling movements along the Z axis (generally) result in an energy consumption outside this range. This ability to accommodate different energy consumption patterns across the axes without assigning a fixed coefficient to the feature allows Random Forest to provide more tailored and accurate predictions.

The observed distribution of the data points on the left side scatter plot also indicates that those milling processes that resulted in below the baseline energy consumption (negative consumption values) were all executed along the Z axis (upward and downward movements).

8.2. Energy consumption versus Feed

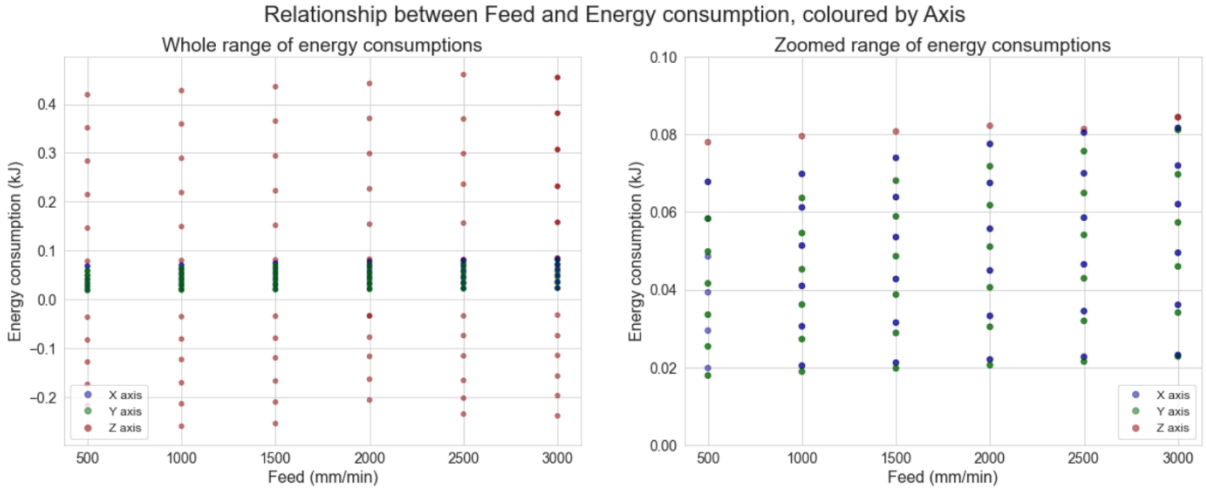

Figure 21: Energy consumption vs the Feed feature scatter plots. Source: own editing.

Figure 21. presents the relationship between the *Feed* feature and the energy consumption coloured by *Axis*. In the context of the addressed manufacturing problem, feed rates (mm/min values) refer to the speed at which the machine's cutting tool has moved along the specified axis (X, Y, or Z) during the milling process. Based on Figure 21, it can be seen stated the distribution of energy consumption values is (in general) nearly the same in case of any feed rate, which suggests that the *Feed* feature has no actual influence on the target variable. Changes in the energy consumption can not be explained (generally) by changes in the feed rate (except for two less influential patterns that are elaborated later on). Instead, the *Path* feature might be accounting better for the varying energy consumptions (*Axis*'s impact was already discussed in section 8.1).

To investigate relationships in the data more in detail, the energy consumptions points in the scatter plot can be broken down by axes (by the colours). According to the findings about the relationship between the *Axis* and the energy consumption values, it is again visible that those milling processes that resulted in atypical (lower or higher than the typical 0.02 and 0.08 kJ range) energy consumption values were performed along the Z axis (red points). This characteristic of the energy consumption based on the used axis is in favour of Random Forest's modelling approach (Ogunleye, 2022), as it can handle Z axis milling processes separately allowing to assign a separate range (different than X and Y axis) of energy consumptions to these processes.

The red points ("Z axis points") occur almost the same number of times and are horizontally almost in-line with each other (almost the same) across all the different feed rates, which

suggests that the *Feed* has no notable influence on the energy consumption in case of upward-downward movements (Z axis).

To be able to examine whether the same "behaviour" is true for left-right (X axis) and forward-backward (Y axis) movements, the range of y-axis values (of energy consumptions) is zoomed for the scatter plot on the right of Figure 21. It can be seen that in case of all the feed rates the blue and green points (denoting X and Y axes) occur (almost) the same number of times and are distributed similarly (except for the smaller effect of ceteris paribus increasing the feed rate and of ceteris paribus changing from X to Y axis and vice versa), which effects will be elaborated on later).

This result and the findings about the red points (Z Axis) indicates that here is no substantial relationship between the *Axis* and the *Feed* features, and that the energy requirement of the milling process cannot be properly estimated based on knowing solely which axis and feed rate were used to execute the milling process.

However, it is worth noting that a less influential pattern can be observed when solely (or in other words ceteris paribus, based on a horizontal consideration of the distribution) the feed rate is increased: milling processes executed along the X and Y axis (blue and green points) consume slightly more energy as the feed rate ceteris paribus increases. This holds true for most of the processes associated with the Z axis (red points) as well. This phenomenon is highly intuitive since a higher speed (feed rate) at which the machine's cutting tool is moving should result in consuming more energy. The fact that the *Feed* is a discrete variable (as it is parameter of the machine set prior to milling) supports Random Forest's great performance, since at the decision nodes of the trees it can make (discrete) decisions based on which feed rate was used out of the 6 possible options.

Besides, another less determining (in estimating the energy consumption) pattern can be seen when the axis is ceteris paribus (based on a vertical consideration of the distribution of blue and green points) changed from the Y to the X axis (and vice versa): executing a milling process along the X axis consumes ceteris paribus (suggested by pairs of blue and green points being close to each other on similar Feed levels) more energy than compared to the Y axis (in these pairs blue (X) points are above the green (Y) points). This identified nature of the *Axis* feature (and that it is a nominal variable) supports Random Forest in making well-tailored decisions contributing to its superior performance (Schonlau & Zou, 2020).

8.3. Energy consumption versus Path

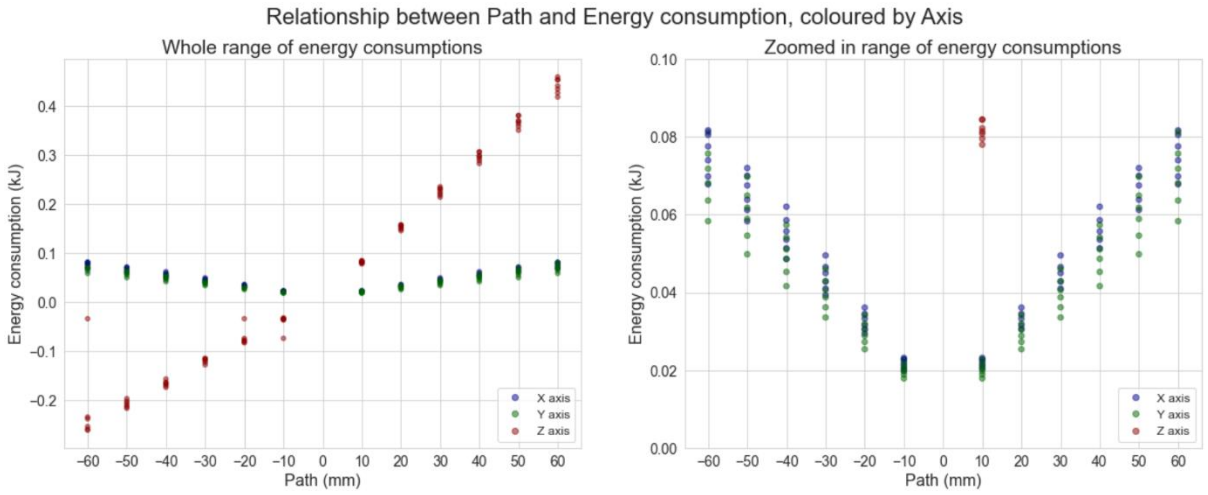

Figure 22: Energy consumption vs the Path feature scatter plots. Source: own editing.

Investigating the scatter plot on the left side of Figure 22. two distinct patterns are visible separated by whether the milling process was executed along the X-Y axis (blue and green points) or the Z axis (red points). These two notable patterns (a rising linear line and a widened "V" shape) are indicating that the *Path* feature has a strong influence on the energy consumption target variable.

First the "V" shaped pattern of data points related to blue and green points (X and Y axis) is examined on the right side of Figure 22. This pattern describes a non-linear relationship between the raw path values and the energy consumption values. However, as it was discussed in section 4.1. (Visual data exploration and data cleaning – Path feature), negative *Path* values denote movements in the opposite direction along the particular axis that was used. Thus, in the scatter plot negative X values indicate that the machine's cutting tool was set to move left (instead of right), and the negative Y values denote backward movements (instead of forward) for the corresponding data points. As the absolute value of the *Path* represents the actual distance moved by the machine's cutting tool (in millimetres), the observed "V" shaped pattern turns out to be intuitive: the more distance the tool moves, the more energy it consumes. In addition, based on a vertical approach to the scatter plot on the right, it can be seen that the blue points (X axis) are generally located at a higher y-axis level than the green ones (Y axis) along the different path values. This shows again that executing a milling process along the X axis consumes ceteris paribus more energy than compared to along the Y axis when the path does not change (ceteris paribus).

Examining the other pattern, the rising linear line related to the red points (Z axis), it can be seen that contrary to the observed non-linear relationship (between the raw values), when the machine's tool was moving upward or downward then the raw path values have a linear influence, more precisely a positive correlation with the energy consumption. Taking into consideration the discussed reason for negative axis values, this relationship seems to be counterintuitive. Since the tool was moving downward (when the path values are negative), since the energy consumption should not decrease as the distance travelled (downward) increases (the raw path value decreases). However, it might be explained through the mechanics of the specific milling machine (no official information about the addressed machine). For example, the machine may well be designed to use counterweights, which reduce the external energy required from electrical sources powering the motors when the tool descends (Slocum, 1992).

Furthermore, considering the vertical spread or dispersion of the data points (regardless of the axis or the colour of the data point) for any particular path value, it can be seen that the variability in the energy consumption decreases as the absolute value of path decreases. This is an intuitive phenomenon, as the less distance the machine's cutting tool has moved, the more consistent the energy consumptions can be expected (more stable energy usage) (Pawanr et al., 2021).

Linking the observations about the relationship between the *Path* feature and the energy consumption with the models' performance, it is apparent that the Linear Regression was not able to capture properly the identified non-linear "V" shaped pattern (only the linear pattern), which lead to the weakest predictive performance. Although Support Vector Regression has the ability to capture the non-linear "V" pattern too by using the kernel trick as discussed in section 5.2. (Systematic discussion of the utilised models – Support Vector Regression – Advantages), it probably struggles to properly segment (even though projecting data into a higher-dimensional space) and handle separately the identified two distinct patterns present across the (same) path values, and fails to correctly model them simultaneously. Nevertheless, the more complex modelling approach trying to account for the non-linear pattern as well yields a better predictive performance than Linear Regression according to Figure 19, but still significantly underperforms Random Forest Regression ("middle" performance). This is due to Random Forest's ability to separately and effectively handle the segmented data scenario described by the distinct Z and X-Y axis patterns through constructing decision rules based on the used axis and path simultaneously (Brieuc et al., 2018).

9. Summary and conclusion

9.1. Results in respect of the research question

The research question examined which one of the applied machine learning algorithms (Linear Regression, Random Forest Regression, Support Vector Regression) is the most accurate in predicting the energy consumption of the milling machine addressed in this study.

Based on the comparative analysis of predictive performance, Random Forest Regression is the model that could manage to effectively capture the relationships and the actual underlying structure of the data, thus this model accounts best for the variance. However, it is worth noting that it makes some small errors when the energy consumption is predicted to be negative (below baseline consumption predicted). Thus, predictions are (slightly) less reliable when the energy consumption is actually below the baseline consumption level (less-than-expected consumption in terms of the investigated machine), but still outstandingly more accurate than the other two models' predictions.

To conclude, Random Forest Regression was found to be the most accurate in predicting the energy consumption of the milling machine, which is in line with the expected answer to the research question based on the literature.

The practical use of these results and of the findings for the hypotheses is discussed in sections 2.2. (Research question – Reason and objective of the research question) and 2.5. (Expected findings and use of results), and includes for example supporting predictive maintenance through foreseeing and optimizing the energy consumption of the machine, reducing downtime, saving costs, and ultimately leading to a greener future for the manufacturing industry.

As a personal contribution to the use of result, I wish to highlight the importance of including Random Forest Regression among the utilised models when addressing similar manufacturing problems with machine learning algorithms.

9.2. Findings with regards to the hypotheses

The first hypothesis assumed that due to the possible non-linear relationships Linear Regression will not be able to properly model the underlying patterns and will demonstrate a poor predictive performance compared to the other two models. According to the conclusions about this model's performance and to the discussion about the non-linear relationships in the data, this hypothesis is proven true.

The second hypothesis presumed that as Support Vector Regression can capture non-linear patterns, it is able to properly model the relationships in the data and does not substantially underperform Random Forest Regression. However, in terms of the addressed problem Support Vector Regression is not modelling some aspects of the data's structure appropriately leading to a significantly worse predictive performance, thus this hypothesis turns out to be false. This might be due to the discussed findings about the leptokurtic distribution of the target variable (not well-separated data points) and about the distinct patterns in the data.
The third hypothesis proposed that by examining patterns in the data more in detail, it is possible to find relationships that account for the results of the performance comparison. When investigating relationships in the data, a "V" and a rising linear line patterns representing two distinct but simultaneous relationships were observed and identified as segmented data scenarios. Accordingly, it was discussed how these can explain the ensemble and support-vector models' established performance corresponding to their different modelling approaches, thus this hypothesis is proven true.

(Personal comment about the results: Although hyperparameter tuning is not in the scope of this study and is mentioned as the limitations of it, I was very interested in how much it might change the predictive performance of the models, thus I experimented with *GridSearchCV* and *optuna* (commented out the respective cells in the Python project). In case of the Random Forest Regression the lowest reached MAE with tuned hyperparameters was actually very slightly worse than with the current default values (0.0080 default vs. 0.0094 kJ tuned). At the same time the lowest reached MSE with tuning was very slightly better (0.00033 default vs. 0.00028 kJ^2 tuned). For SVR hyperparameter tuning did not improve the results at all, thus it seems like with the combination of the default hyperparameter values the model already fits the data as well or as closely as it can). These experiments with hyperparameter tuning confirm and verify the reliability and stability of the results in the study's current scope (results without hyperparameter tuning)).

10. Limitations and directions for further research

The exact findings and results of this study can be interpreted only in the context of the addressed problem, but the practice of utilising different machine learning algorithms in an industrial setting and the approaches to finding the one with the best predictive performance can be applied to other manufacturing problems.

This study in the current manufacturing context can be extended by several machine learning concepts, which also mean the limitations of the current study. These concepts are subject to further research (directions for further research) and include hyperparameter tuning, investigating feature importance, exploring different feature scaling methods, using cross validation to evaluate performance and additional evaluation metrics like mean absolute percentage error (MAPE), and utilising additional machine learning models such as gradient boosting-based algorithms. Another manufacturing context where instead of regression a classification problem is involved is also worth exploring in further research.

There are also some further questions identifying areas for prospective research. These considerations lead to further optimizing the energy consumption and resources of machines contributing to predictive maintenance, reducing costs and downtimes, improving manufacturing processes, and ultimately promoting sustainability in the manufacturing industry.

- Given an existing predictive maintenance practice used in the manufacturing industry (possibly related to energy consumption of a machine), how exactly can the findings of this study be integrated into it?
- How can the use of machine learning algorithms for manufacturing problems be further improved for greater efficiency?
- How can the findings of this study be translated into cost savings in the manufacturing industry?
- When addressing another manufacturing problem, how can the practices of this study be customized and replicated to apply them?
- How can the findings of this study be integrated into broader efforts of promoting sustainability and reducing energy consumption for a greener future?

List of references

Literature

Agrawal, A., Goel, S., Rashid, W. B., & Price, M. (2015). Prediction of surface roughness during hard turning of AISI 4340 steel (69 HRC). *Applied Soft Computing, 30*, 279-286. https://doi.org/10.1016/j.asoc.2015.01.059

Anscombe, F. J., & Tukey, J. W. (1963). The Examination and Analysis of Residuals. *Technometrics, 5*(2), 141–160. https://doi.org/10.1080/00401706.1963.10490071

Bányai, Á. (2021). Energy consumption-based maintenance policy optimization. *Energies, 14*(18), Article 5674. https://doi.org/10.3390/en14185674

Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing - Letters and Reviews, 11*(10), 203-224. https://www.researchgate.net/publication/228537532_Support_Vector_Regression

Behrens, J. T. (1997). Principles and Procedures of Exploratory Data Analysis. *Psychological Methods, 2(*2), 131-160. https://doi.org/10.1037/1082-989X.2.2.131

Bousquet, O., Boucheron, S., & Lugosi, G. (2004). Introduction to Statistical Learning Theory. In O. Bousquet, U. Luxburg & G. Rätsch (Eds.), *Advanced Lectures on Machine Learning* (pp. 169-207). Springer. https://doi.org/10.1007/978-3-540-28650-9_8

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5–32. https://doi.org/10.1023/A:1010933404324

Brieuc, M. S. O., Waters, C. D., Drinan, D. P., & Naish, K. A. (2018). A practical introduction to Random Forest for genetic association studies in ecology and evolution. *Molecular Ecology Resources, 18*(4), 755-766. https://doi.org/10.1111/1755-0998.12773

Brillinger, M., Wuwer, M., Hadi, M. A., & Haas, F. (2021). Energy prediction for CNC machining with machine learning. *CIRP Journal of Manufacturing Science and Technology*, *35*, 718-721. https://doi.org/10.1016/j.cirpj.2021.07.014

Caruana, R., & Niculescu-Mizil, A. (2006, June 25–29). *An empirical comparison of supervised learning algorithms* [Conference presentation]. 23rd International Conference on Machine Learning (ICML), Pittsburgh, Pennsylvania, USA. https://doi.org/10.1145/1143844.1143865

Charalampous, P. (2021). Prediction of Cutting Forces in Milling Using Machine Learning Algorithms and Finite Element Analysis. *Journal of Materials Engineering and Performance, 30*, 2002-2013. https://doi.org/10.1007/s11665-021-05507-8

Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability, 12*(19), Article 8211. https://doi.org/10.3390/su12198211

Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608v2 [stat.ML]. https://doi.org/10.48550/arXiv.1702.08608

Dubey, V., Sharma, A. K., & Pimenov, D. Y. (2022). Prediction of Surface Roughness Using Machine Learning Approach in MQL Turning of AISI 304 Steel by Varying Nanoparticle Size in the Cutting Fluid. *Lubricants, 10*(5), Article 81. https://doi.org/10.3390/lubricants10050081

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining*. Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

Groeneveld, R. A., & Meeden, G. (1984). Measuring Skewness and Kurtosis. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *33*(4), 391–399. https://doi.org/10.2307/2987742

Groten, M., & Gallego-García, S. (2021). A Systematic Improvement Model to Optimize Production Systems within Industry 4.0 Environments: A Simulation Case Study. *Applied Sciences, 11*(23), Article 11112. https://doi.org/10.3390/app112311112

Gujarati, D. N. (2019). *Linear Regression: A Mathematical Introduction*. SAGE Publications. https://doi.org/10.4135/9781071802571

Hamel, L. (2009). Regression with Support Vector Machines. In D. T. Larose (Ed.), *Knowledge Discovery with Support Vector Machines* (pp. 193-208). John Wiley & Sons. https://doi.org/10.1002/9780470503065.ch12

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

Hoffmann, J.P. (2021). *Linear Regression Models: Applications in R* (1st ed.). Chapman and Hall. https://doi.org/10.1201/9781003162230

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. https://doi.org/10.1007/978-1-4614-7138-7

Kang, M., & Tian, J. (2018). Machine Learning: Data Pre-processing. In M. G. Pecht & M. Kang (Eds.), *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things* (pp. 111-130). John Wiley & Sons. https://doi.org/10.1002/9781119515326.ch5

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* Springer. https://doi.org/10.1007/978-1-4614-6849-3

Liang, Y., Xu, Q.-S., Li, H.-D., & Cao, D.-S. (2011). *Support vector machines and their application in chemistry and biotechnology.* CRC Press. https://doi.org/10.1201/b10911

Murphy, K. P. (2012). *Machine learning: A Probabilistic Perspective*. The MIT Press. https://mitpress.mit.edu/9780262018029/machine-learning/

Navlani, A., Fandango, A., & Idris, I. (2021). *Python Data Analysis: Perform data collection, data processing, wrangling, visualisation, and model building using Python*. Packt Publishing Ltd. https://books.google.hu/books?id=DN4SEAAAQBAJ&printsec=frontcover&hl=hu

Nguyen, T.-T. (2019). Prediction and optimization of machining energy, surface roughness, and production rate in SKD61 milling. *Measurement, 136*, 525-544. https://doi.org/10.1016/j.measurement.2019.01.009

Ogunleye, J. O. (2022). Predictive data analysis using linear regression and random forest. In B. S. Kumar (Ed.), *Data Integrity and Data Governance* (Chapter 6.). IntechOpen. https://doi.org/10.5772/intechopen.107818

Pawanr, S., Garg, G. K., & Routroy, S. (2021). Modelling of variable energy consumption for CNC machine tools. *Procedia CIRP, 98*, 247-251. https://doi.org/10.1016/j.procir.2021.01.038

Sangwan, K. S., & Sihag, N. (2019). Multi-objective optimization for energy efficient machining with high productivity and quality for a turning process. *Procedia CIRP, 80*, 67-69. https://doi.org/10.1016/j.procir.2019.01.022

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal, 20*(1), 3-29. https://doi.org/10.1177/1536867X20909688

Schorr, S., Möller, M., Heib, J., & Bähre, D. (2020). Quality Prediction of Drilled and Reamed Bores Based on Torque Measurements and the Machine Learning Method of Random Forest. *Procedia Manufacturing, 48*, 894-901. https://doi.org/10.1016/j.promfg.2020.05.127

Seber, G. A. F., & Lee, A. J. (2012). *Linear Regression Analysis* (2nd ed.). John Wiley & Sons. https://www.wiley.com/en-us/Linear+Regression+Analysis%2C+2nd+Edition-p-9781118274422

Shang, M., & Lakshmi T C, G. (2020). *Hands-on Supervised Learning with Python*. BPB Publications. https://www.google.hu/books/edition/Hands_on_Supervised_Learning_with_Python/LW8SEAAAQBAJ?hl=hu&gbpv=0

Slocum, A. H. (1992). *Precision Machine Design*. Society of Manufacturing Engineers. https://books.google.hu/books/about/Precision_Machine_Design.html?id=uG7aqgal65YC&redir_

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Sule, A. (2021). *Milling process energy consumption* [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10976208

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., & Elger, G. (2021). Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering & System Safety, 215*, 3-6. https://doi.org/10.1016/j.ress.2021.107864

Vinh, T. Q., & Huy, N. T. (2022, November 21–23). *Predictive Maintenance IoT System for Industrial Machines using Random Forest Regressor* [Conference presentation]. 2022 International Conference on Advanced Computing and Analytics (ACOMPA), Ho Chi Minh City, Vietnam. https://doi.org/10.1109/ACOMPA57018.2022.00020

Vinh, T. Q., & Huy, N. T. (2022, November 21–23). *Predictive Maintenance IoT System for Industrial Machines using Random Forest Regressor* [Conference presentation]. 2022 International Conference on Advanced Computing and Analytics (ACOMPA), Ho Chi Minh City, Vietnam. https://doi.org/10.1109/ACOMPA57018.2022.00020

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research, 30*(1), 79-82. https://doi.org/10.3354/cr030079

Wu, D., Jennings, C., Terpenny, J., Gao, R. X., & Kumara, S. (2017). A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering, 139*(7), Article 071018. https://doi.org/10.1115/1.4036350

Yan, X., & Su, X. G. (2009). *Linear regression analysis: Theory and computing.* World Scientific Publishing. https://doi.org/10.1142/6986

Yuwen, S., Jinjie, J., Jinting, X., Mansen, C., & Jinbo, N. (2022). Path, feedrate and trajectory planning for free-form surface machining: A state-of-the-art review. *Chinese Journal of Aeronautics, 35*(8), 12-29. https://doi.org/10.1016/j.cja.2021.06.011

Figures

Figure 1:
PCBWay. (2021). *What is CNC milling, and how it works?*. PCBWay. https://www.pcbway.com/blog/CNC_Machining/What_is_CNC_milling__and_how_it_works_.html. Retrieved 11/02/2024.

Figure 2:
„Pasixxxx". (2009). *Fräsen.* Wikipedia. https://de.wikipedia.org/wiki/Zerspanen#/media/Datei:Fraisage_surfacage.svg. Retrieved 11/02/2024.

Figure 3:
„sbhhdp". (2020). *How to remove or hide y-axis ticklabels from a matplotlib / seaborn plot.* Stack Overflow. https://stackoverflow.com/questions/63756623/how-to-remove-or-hide-y-axis-ticklabels-from-a-matplotlib-seaborn-plot. Retrieved 11/02/2024.

Figure 4:
"VinothkumarSivaraj" (2020). BASIC MILLING CENTER. Slide 5. SlideShare. https://www.slideshare.net/slideshow/pmc-basic-final/226494932. Retrieved 26/02/2024.

Figure 5:
"Samueldavidwinter". (2022). *Skewness and Kurtosis.* Medium. https://medium.com/@samueldavidwinter/skewness-and-kurtosis-3db02915d48f. Retrieved 21/03/2024.

Figure 6:
Huang, E. (n.d.). *Histogram Distributions.* BioRender. https://www.biorender.com/template/histogram-distributions. Retrieved 15/04/2024.