



German International University



Technische Hochschule Ulm

Faculty Of Engineering

Development of an AI-supported algorithm for the differentiated selection of shaft-hub connections under specific conditions

Bachelor Thesis

Author: Omar Massoud
Supervisor: Prof. Dr.-Ing. Michael Lätzer
Prof. Dr.-Ing. Jens Bähr
Submission Date: January, 2025



German International University



Technische Hochschule Ulm

Faculty Of Engineering

Development of an AI-supported algorithm for the differentiated selection of shaft-hub connections under specific conditions

Bachelor Thesis

Author: Omar Massoud
Supervisor: Prof. Dr.-Ing. Michael Lätzer
Prof. Dr.-Ing. Jens Bähr
Submission Date: January, 2025

This is to certify that:

- (i) the thesis comprises only my original work toward the Bachelor Degree
- (ii) due acknowledgement has been made in the text to all other material used

Omar Massoud
January, 2025

Acknowledgments

First and foremost, I would like to extend my heartfelt thanks to my academic supervisor, Prof. Dr.-Ing. Michael Latzer. Your guidance has been invaluable, and your insights have not only shaped this work but also helped me grow as a student and as an engineer. I am grateful for your patience, your dedication to my success, and your constant willingness to share your knowledge. The time and effort you have invested in my development are greatly appreciated. Finally, I want to express my gratitude to everyone who has contributed in some way or another to this achievement. Whether it was through words of encouragement, helpful advice, or simply being there when I needed it, your support has made all the difference.

Abstract

This paper focuses on how Artificial Intelligence (AI) can help improve the selection of Shaft-Hub Connection (SHC), such as key fits, spline fits, and press fits. By using machine learning models like XGBoost and Random Forest, the study aims to find the best connection type based on specific requirements, such as material properties and dimensions. The results show that XGBoost performs well but can sometimes over fit, while Random Forest provides more reliable results, especially when working with smaller datasets. Although the study faced challenges with limited and imbalanced data, it highlights the potential of AI to make the SHC selection process faster and more accurate. This is particularly useful in industries like automotive and aerospace, where precision and efficiency are important. Future research could focus on creating a larger and more balanced dataset, using simulation tools to generate data, and testing more advanced AI techniques. This work provides a solid starting point for using AI to solve mechanical design problems.

Dieses Papier konzentriert sich darauf, wie AI die Auswahl von SHC, wie Passfederverbindungen, Zahnwellenverbindungen und Pressverbindungen, verbessern kann. Durch die Verwendung von maschinellen Lernmodellen wie XGBoost und Random Forest zielt die Studie darauf ab, den besten Verbindungstyp basierend auf spezifischen Anforderungen wie Materialeigenschaften und Abmessungen zu finden. Die Ergebnisse zeigen, dass XGBoost gut abschneidet, aber manchmal zu Overfitting neigt, während Random Forest insbesondere bei kleineren Datensätzen zuverlässigere Ergebnisse liefert. Obwohl die Studie Herausforderungen wie begrenzte und unausgewogene Daten begegnete, unterstreicht sie das Potenzial von AI, den SHC-Auswahlprozess schneller und genauer zu gestalten. Dies ist besonders nützlich in Branchen wie der Automobil- und Luftfahrtindustrie, wo Präzision und Effizienz von entscheidender Bedeutung sind. Zukünftige Forschung könnte sich darauf konzentrieren, einen größeren und ausgewogeneren Datensatz zu erstellen, Simulationstools zur Datengenerierung zu verwenden und fortgeschrittenere AI-Techniken zu testen. Diese Arbeit bietet eine solide Grundlage für die Nutzung von AI, um Probleme im mechanischen Design zu lösen.

Contents

Acknowledgments	V
1 Introduction	1
1.1 Introduction	1
1.2 Thesis Structure	2
2 Background	3
2.1 Shaft Hub Connections	3
2.1.1 Interference Fit	4
2.1.2 Clearance Fit	7
2.2 Machine Learning Algorithms	11
2.2.1 Supervised Learning	11
2.2.2 Extreme Gradient Boosting Algorithm	12
3 Literature Review	15
3.1 Integration of AI with Engineering Applications	15
3.1.1 Shaft Hub Connections	16
4 Methods	19
4.1 Defining the Scope of SHC Types	19
4.1.1 Selection Rationale	19
4.1.2 Documentation and Specifications	19
4.2 Dataset Creation	20
4.2.1 FEA Approach	20
4.2.2 Analytical Solution Approach	20
4.3 Details of Data Collection	20
4.3.1 Initial Dataset	20
4.3.2 Finalized Dataset	21
4.4 Further Development of dataset	25
4.4.1 Logic of the Approach	26
4.5 Python Script Development	28
4.5.1 Random Forest	29

5 Results	33
5.1 Initial Results	33
5.1.1 Dataset Overview	33
5.1.2 Data Cleaning and PCA Analysis	33
5.1.3 Model Training and Parameter Optimization	35
5.1.4 Overfitting Analysis and Model Generalization	35
5.2 Finalized Results	36
5.2.1 Comparative Analysis	37
5.3 New Approach Results	40
5.3.1 Testing and Validating	40
6 User Manual	43
6.1 Introduction	43
6.2 Prerequisites	43
6.2.1 Hardware Requirements	43
6.2.2 Software Requirements	43
6.2.3 Installing Dependencies	44
6.2.4 Dataset Requirements	44
6.2.5 Load the Dataset	44
6.2.6 Encoding Features	44
6.2.7 Oversampling	44
6.2.8 Train-Test Split	44
6.2.9 Training the Model	45
6.2.10 Evaluating the Model	45
6.3 Using the GUI for Predictions	45
6.3.1 Running the GUI	45
6.3.2 Making a Prediction	45
6.3.3 Error Handling	45
6.4 Troubleshooting	46
6.5 Example Outputs	46
7 Conclusion	47
Appendix	49
A Lists	50
List of Abbreviations	50
List of Symbols	51
List of Figures	52
References	54

Chapter 1

Introduction

1.1 Introduction

AI has become increasingly prominent in recent years, significantly impacting a wide range of fields. Ongoing AI research continues to unlock new possibilities, demonstrating its effectiveness in simplifying numerous tasks. AI's versatility allows it to be applied across various fields, making it an essential tool in modern life. One prominent example is Chat Generative Pre-Trained Transformer (Chat GPT), which has been widely adopted to enhance efficiency in everyday tasks. Machine Learning (ML), a subfield of AI, focuses on developing systems that can learn and adapt without explicit programming. Using algorithms and statistical models, ML enables computers to analyze data, identify patterns, and make predictions or decisions based on these insights. ML can be categorized into four main approaches: Supervised Learning (SL), where labeled datasets are used to train algorithms for effective data classification and event prediction; Unsupervised Learning (UL), where patterns are identified from unlabeled data to uncover hidden structures; semi-supervised learning, which combines a small amount of labeled data with a large amount of unlabeled data to enhance model accuracy; and Reinforcement Learning (RL), where systems learn by receiving rewards for desired actions and penalties for unwanted behaviors, optimizing performance through trial and error. Each approach serves different purposes, making ML adaptable to a wide range of applications depending on the nature of the data and the specific model being trained.

In Engineering, however, the potential of AI is still under exploration, especially in areas such as engineering design; numerical analysis, and failure theories. The integration of AI in engineering design is a promising yet under-researched area. This raises the question: Could AI integration in engineering design improve productivity? Given the broad scope of engineering design, this research focuses specifically on SHC, employing AI to determine the most suitable connection for specific input parameters. By utilizing a popular supervised learning model, Extreme Gradient Boosting (XGBoost), the research explores the efficiency of AI compared to classical methods. XGBoost has been shown

to outperform traditional techniques in terms of speed and accuracy, particularly in handling complex datasets and producing more precise results. The use of AI in engineering design presents several significant benefits, particularly in enhancing productivity and safety. First, AI reduces the need for manual labor by automating and optimizing production processes, which helps to reduce human error and the limitations of traditional methods. This automation leads to increased productivity as production systems can operate continuously and adapt to various production needs and environmental changes. Additionally, AI technologies contribute to improved safety in machinery production by allowing robots and automated systems to perform hazardous tasks, thus minimizing the risk of accidents for human workers. AI also plays a vital role in quality control, where advanced systems can monitor and identify potential hazards in real time, ensuring that products adhere to high-quality standards. Overall, the integration of AI in mechanical design and manufacturing processes not only boosts efficiency but also enhances safety and quality, providing a solid foundation for the economic performance of organizations [11].

1.2 Thesis Structure

The thesis is structured into six chapters, each serving a distinct purpose. **Chapter 1: Introduction** provides the aim, motivation, and the research question or problem statement that underpins this work. **Chapter 2: Background** offers the theoretical framework for understanding SHC and AI algorithms, laying the groundwork for the study. **Chapter 3: Literature Review** examines previous research, focusing on the integration of AI with engineering applications and past studies related to SHC optimization. **Chapter 4: Methods** describes the methodology employed in this thesis, detailing the processes and approaches used to address the research questions. **Chapter 5: Results** presents the outcomes of the study, including a comparative analysis of algorithmic results, demonstrating the program's performance and validating its functionality. Finally, **Chapter 6: Conclusion** summarizes the findings, discusses their implications, and provides recommendations for future research directions.

Chapter 2

Background

Since the onset of the first industrial revolution in the 18th century, the world has faced the challenge of increasing production while relying on limited and depleting natural resources to meet growing consumer demands, while striving to reduce negative environmental and social impacts [6]. This chapter focuses on traditional methods for calculating stresses in each SHC, examining the critical aspects of each connection to determine the safest connection, through comparing several features.

2.1 Shaft Hub Connections

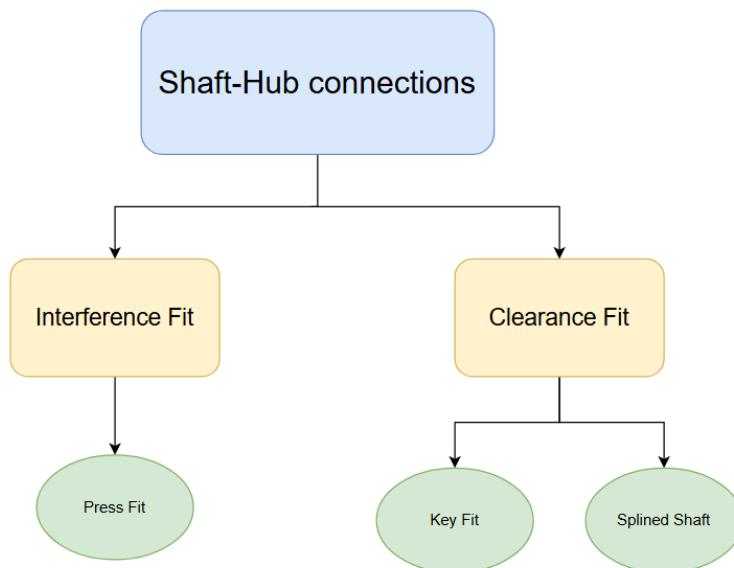


Figure 2.1: Shaft-Hub connections

2.1.1 Interference Fit

An Interference Fit is a type of connection between a shaft and a hub, where the shaft is slightly larger than the hole in the hub. Due to the difference in size, the parts press tightly together when they are joined, creating strong friction. To assemble these parts, a lot of force is needed, or sometimes heat is used to expand the hub so the shaft can fit inside. The shaft is slightly squeezed by the hub, which makes sure that they do not slip or move under normal conditions. Unlike the Clearance Fit in Section 2.1.2, where there is a small space between the parts, an interference fit provides a tight connection that is very good at transferring torque and handling axial loads. Figure 2.2 demonstrate the different types of interference fits.

Press Fit

Press Fit a type of assembly obtained from interference fit when a hub (outer part) is machined to be hollow with a tolerance for a smaller diameter of a shaft to be joined together using force (force fit), normally applying oil as a lubricant. This process creates elastic deformation due to friction. Consequently, this type of fit relies on frictional force, transmitting shaft torque to the hub and resisting axial motion. Figure 2.2a demonstrates the press fit. Press fits are ideal for applications requiring a strong, permanent connection with high load-bearing capacity. This type of fit creates an interference bond between the shaft and hub, distributing stress evenly across the contact area and avoiding stress concentrations. This makes press fits highly suitable for high-strength, high-speed applications, such as automotive and aerospace systems, where durability and stability are critical. However, press fits are challenging to assemble and disassemble, and they can be affected by temperature changes due to thermal expansion. Consequently, press fits are preferable when a secure, non-removable connection is needed, and maintenance requirements are minimal. The force fit creates pressure acting on the surface where the outer and inner layers meet. This pressure can permanently deform the assembly, especially in fragile materials like cast iron, which may even fracture. For stronger materials, such as steel, the stress results in elastic deformation. The outer member expands under pressure, with tangential tensile stress reaching a maximum at the mating surface, while the inner member contracts. This results in both radial and tangential compressive stresses.

Figure 2.3 evaluates the properties of a cylindrical press-fit connection (**Zylindrischer Pressverband**) based on several criteria, these criteria are also mentioned for the other connections. Each criterion is explained below:

- **Disassembly capability of the WNV (Demontierbarkeit der WNV):** Refers to how easily the connection can be disassembled when necessary.
- **Notch effect of the WNV on components (Kerbwirkung der WNV auf Bauteile):** Describes the stress concentrations or weakening effects that the connection might impose on the components.

- **Transmissible torque (Übertragbares Drehmoment):** Indicates the maximum torque the connection can reliably transmit without failure.
- **Self-centering (Selbstzentrierung):** Evaluates the ability of the connection to align itself automatically during assembly, ensuring proper fit and function.
- **Assembly effort (Montageaufwand):** Measures the amount of effort and resources required to assemble the connection.
- **Costs associated with WNV (WNV-Kosten):** Represents the financial cost of manufacturing and implementing the connection, taking into account material, labor, and other factors.

Evaluation Ratings

Each property is rated based on the following scale:

- **Good (Gut):** Indicates a positive or optimal characteristic.
- **Minimal (Gering):** Indicates low levels of effort or impact.
- **Favorable (Günstig):** Represents cost-effectiveness or economic feasibility.
- **Low (Niedrig):** Refers to a low magnitude of a particular property (e.g., stress or effort).
- **High (Hoch):** Refers to a high magnitude of a property (e.g., torque capacity).

Figure 2.3 provides a visual evaluation of the cylindrical press-fit connection, comparing its performance across key properties. These properties are critical for assessing the suitability of this connection type in applications such as mechanical assemblies.

The objective is typically to determine the magnitude of the pressure due to the given interference, choosing the lowest generated pressure, then calculating the allowable torque for this given pressure that was chosen. The computation of these steps are as follows:

$$M_t = K_A \cdot M_{\text{nenn}} \quad (2.1)$$

Where M_t is the allowable moment, and K_A is the application factor.

$$F_u = \frac{2 \cdot M_t}{d} \quad (2.2)$$

Equation 2.2 shows the minimum pressure, where F_u is the tangential force in the press fit, and d is the diameter of shaft.

$$p_{\text{perf}} = \frac{F_u \cdot S_R}{d \cdot \pi \cdot l \cdot \mu} \quad (2.3)$$

Where p_{perf} is the minimum allowable pressure, S_R is the safety against slipping, L is the length of the hub and finally μ is the coefficient of friction.

$$Q = \frac{d}{D} \quad (2.4)$$

$$\sigma_{\text{Hub}} = \frac{\text{yield_strength_hub}}{SF}, \quad \sigma_{\text{shaft}} = \frac{\text{yield_strength_shaft}}{SF} \quad (2.5)$$

$$p_{\text{hub}} = \frac{1 - Q^2}{\sqrt{3}} \sigma_{\text{Hub}}, \quad p_{\text{shaft}} = \frac{2}{\sqrt{3}} \sigma_{\text{shaft}} \quad (2.6)$$

Comparing p_{hub} and p_{shaft} to p_{perf} , to be able to check if they there values are greater than the minimum allowed. Thus, the chosen pressure p is

$$p = \min(p_{\text{hub}}, p_{\text{shaft}}) \quad (2.7)$$

Calculating Allowable Press-Fit Torque

$$\text{Torque} = \frac{\pi \times \mu \times p \times 1 \times (d)^2}{2 \times 1000} \quad (2.8)$$

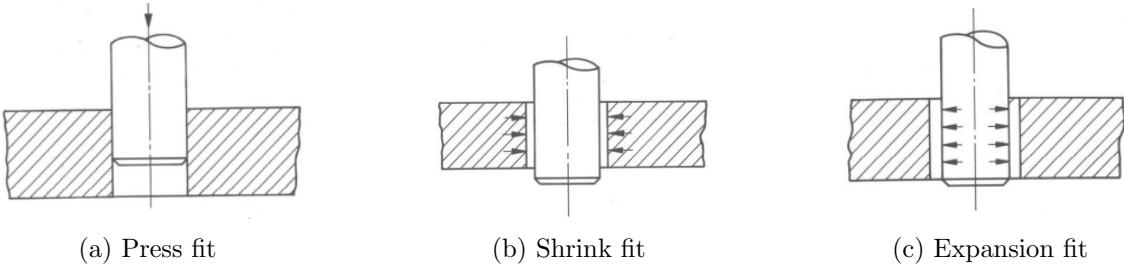


Figure 2.2: Different types of interference fits [3].

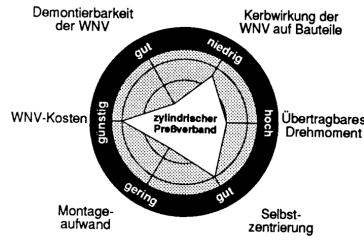


Figure 2.3: Properties of Press Fit [8]

2.1.2 Clearance Fit

Keyed Shaft

A key is a machine component positioned at the junction between a shaft and the hub of a power transmitting element to facilitate torque transfer. Keyed fits offer a practical and cost-effective solution for applications where moderate torque transmission is required, and regular maintenance or disassembly is anticipated. By using a key inserted into matching slots on the shaft and hub, keyed fits prevent relative rotation, providing a stable connection that is easier to assemble and disassemble than press fits. However, the presence of the keyway introduces stress concentrations, making keyed fits less suitable for high-load applications. Thus, keyed fits are advantageous in machinery where easy disassembly is required, such as in pumps or conveyors, but are not ideal for high-strength connections. The key fits into an axial groove on the shaft known as a keyseat, with a corresponding groove on the hub called a keyway. Figure 2.4 visualizes keyed shaft. Typically, the key is first installed into the shaft's key seat, after which the hub's keyway is aligned with the key, and the hub is then slid into place. There are many types of keys, such as Square/ Rectangular parallel keys, Taper and Gib Head keys and Pin Keys; However, this research focuses on Square/ Rectangular Keys, and a special type of keys called Tooth Shaft explained in the following section. Considering Rectangular Keys, stress is calculated to determine Key Length as follows:

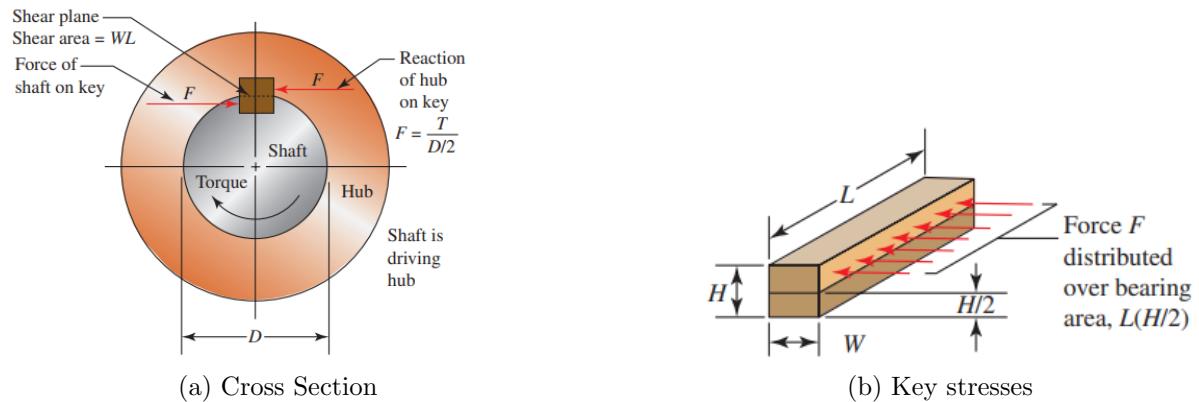


Figure 2.4: Schematic representation of a keyed shaft [9].

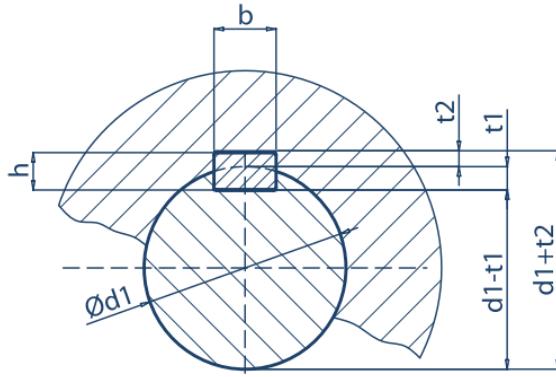


Figure 2.5: Visualization for calculations

Calculating Key Length

$$l_{\text{key}1} = \frac{2 t_{\text{eq}}}{\text{pressure_shaft} \cdot d_1 \cdot t_1} \quad (2.9)$$

$$l_{\text{key}2} = \frac{2 t_{\text{eq}}}{\text{pressure_key} \cdot d_1 \cdot t_2} \quad (2.10)$$

Where `pressure_shaft` and `pressure_key` are defined as:

$$\text{pressure_shaft} = \text{Yield strength Shaft} \cdot \text{Safety factor} \quad (2.11)$$

$$\text{pressure_key} = \text{Yield strength key} \cdot \text{Safety factor} \quad (2.12)$$

$$l_1 = l_{\text{key}1} + b \quad (2.13)$$

$$l_2 = l_{\text{key}2} + b \quad (2.14)$$

Where b is equal to the base of the key. Then, the greater value of l_1 OR l_2 is chosen to ensure structural safety.

If choose = l_1 :

$$t = (\text{pressure_shaft} \times t_1 \times (l_1 - b)) \left(\frac{d_1}{2} \right) \quad (2.15)$$

If choose = l_2 :

$$t = (\text{pressure_key} \times t_2 \times (l_2 - b)) \times \left(\frac{d_1}{2} \right) \quad (2.16)$$

Where:

- t : Torque being checked.
- pressure_shaft: Yield strength · Safety factor for the shaft.
- pressure_key: Yield strength · Safety factor for the key.
- l_1 OR l_2 : Length of key.
- b : Base of the key.
- d_1 : Diameter of the shaft.

The appropriate value of l_1 or l_2 is chosen based on the condition to ensure structural integrity, hence l is the longer length of either l_1 or l_2 .

Figure 2.6 evaluates the characteristics of a **key connection (Paßfederverbindung)** based on the **DIN 6885** standard.

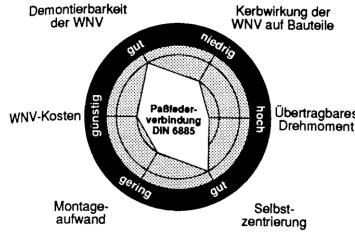


Figure 2.6: Properties of Key Fit [8]

Tooth Shaft

A Tooth Shaft, also known as a spline, is an axial keyed shaft characterized by multiple grooves machined along its surface, each corresponding to a key. Unlike removable keys, found in Section 2.1.2, these grooves are integrated to the body of the shaft, making the shaft machined that if disassembly required, the whole shaft shall be removed. Spline fits are preferable for applications involving high torque transmission, as they offer a greater contact area than keyed fits, distributing the load across multiple teeth and reducing localized stress. Spline connections also allow for some axial movement, providing flexibility in applications where positioning adjustments may be necessary. Though spline fits are more complex and costly to manufacture, they excel in high-performance settings where durability and precise torque transfer are essential, such as in automotive transmissions and heavy machinery. This makes spline fits ideal for high-torque, adjustable applications where strength and precision are prioritized. Both Keyed and Tooth Shafts preforms the same in transmitting torques; however, splines can maintain the transfer of higher torques, since there are more than two keys in the given shaft hub interface. As for the relative motion, a shaft and hub are moving in the same motion; no relative motion present. Figure 2.7 represents the general assembly of a tooth shaft. Figure 2.8

evaluates the characteristics of a **spline connection** (**Zahnwellenverbindung**) based on the **DIN 5480** standard.

Calculating Minimum Spline Length

$$l = \frac{l_{eq} \times 1000}{0.75 \times htr \times z \times rm \times p} \quad (2.17)$$

Here,

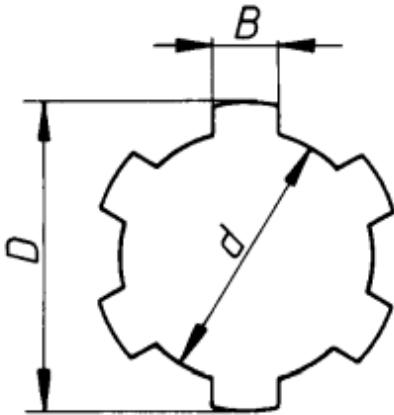
$$htr = \frac{D - d}{2} \quad (2.18)$$

And z is the number of teeth, and p is the minimum pressure from either the shaft or hub.

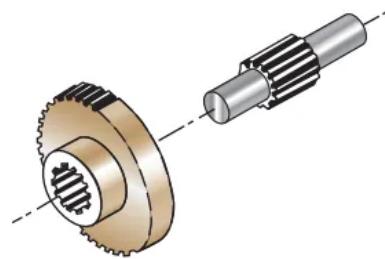
$$rm = \frac{D + d}{4} \quad (2.19)$$

Calculating Torque Capacity for a Given Spline Length

$$\text{Torque} = \frac{0.75 \times htr \times z \times rm \times p \times l}{1000} \quad (2.20)$$



(a) Geometry of a splined shaft.



(b) Assembly of splined shaft and hub.

Figure 2.7: Schematic representation of a splined shaft [9].

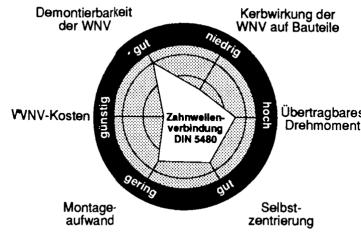


Figure 2.8: Properties of Splined Shaft [8]

2.2 Machine Learning Algorithms

2.2.1 Supervised Learning

First of all, SL uses prelabeled classes to train a dataset, hence annotation required; classes are determined before the training phase, defined by a human. There are two distinctive models of SL algorithms: classification models (classifier) or regression models. Having said this, the learning process is as follows divided into two steps, training and then testing. In the first step (training), training data are taken as input, depending on what features are input/outputs algorithms learn, building a learning model. Then comes step two of testing/ validation, where the model compares the predicted value (Output) to targeted value (defined output), finding the accuracy of the model. Later, New Data are provided, classified, and then predicting the output; which is the most common technique for classification. Not to mention, neural networks (NN) and decision trees are commonly trained using SL [10]. In Figure 2.9, the classification method is demonstrated.

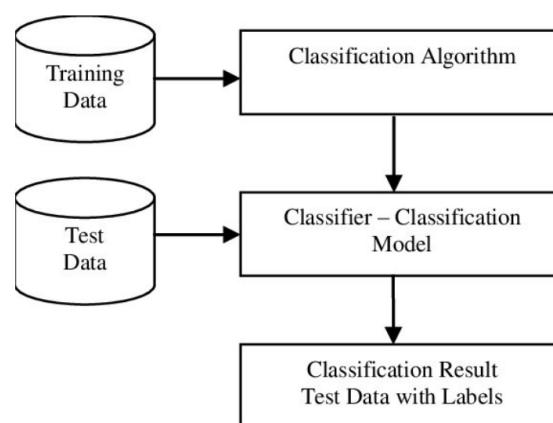


Figure 2.9: Classification process from training data to labeled test data [4].

2.2.2 Extreme Gradient Boosting Algorithm

Decision trees classifier (DCT), used in multi-class classification. The model recursively splits the data based on feature values, forming a tree structure where each node represents a decision. It is particularly useful because it is easy to interpret, it can handle both numerical and categorical data, and performs well with large datasets. Decision trees can also be used in regression tasks, although they are most commonly used in classification. Extreme Gradient Boosting Algorithm is used to optimize the decision trees, utilized by XGBoost. XGBoost is originated from gradient boosting machine [5], with the addition of a weak learner and over fitting optimizations. Where the mathematical equations are as follows:

The regularized loss function is defined as:

$$L(\theta) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{k=1}^K \Omega(f_k) \quad (2.21)$$

Here, $L(\theta)$ is the total loss function, $l(\hat{y}_i, y_i)$ is the loss for each predicted and actual pair (\hat{y}_i, y_i) , and $\Omega(f_k)$ represents the regularization term for each tree f_k . The goal is to minimize $L(\theta)$, balancing predictive accuracy and model complexity.

The boosting update is defined as:

$$y_i^{(t)} = y_i^{(t-1)} + f_t(x_i) \quad (2.22)$$

In this equation, $y_i^{(t)}$ is the predicted value at iteration t , $y_i^{(t-1)}$ is the prediction from the previous iteration, and $f_t(x_i)$ is the new tree added at iteration t to correct the error from the previous prediction.

The approximation of the loss function using the second-order Taylor expansion is given by:

$$L^{(t)} \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (2.23)$$

In this equation, g_i and h_i are the gradient and Hessian (second derivative) of the loss function, respectively, which help in approximating the loss function during optimization. $f_t(x_i)$ is the new tree at iteration t , and $\Omega(f_t)$ is the regularization term for this new tree.

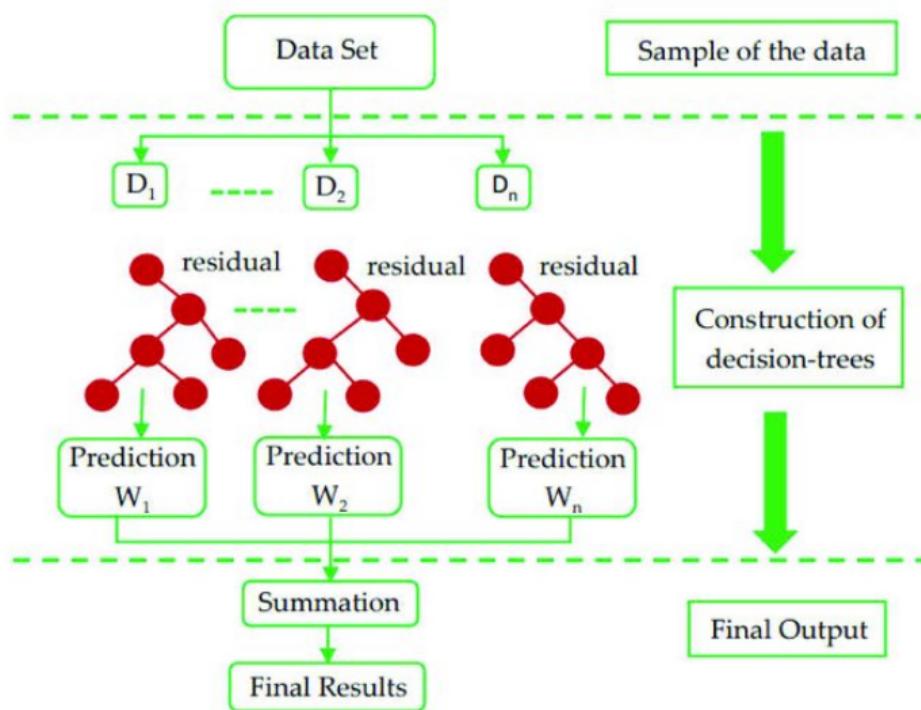


Figure 2.10: Flowchart of the XGBoost algorithm [7].

Chapter 3

Literature Review

As this research explores a new research area, no previous studies have been conducted specifically in this field. However, AI has been applied and integrated into some related engineering applications in the past.

3.1 Integration of AI with Engineering Applications

A wind turbine experiences significant distributed tension on its blades, and this research delves into the integration of AI with Finite Element Analysis (FEA) to improve the precision of stress distribution predictions within simulations. Specifically, the study employs machine learning models, namely Support Vector Machine (SVM) and K-nearest neighbor (KNN), to enhance accuracy. SVM, recognized for its capabilities in both classification and regression, was evaluated in terms of predictive accuracy using the Root Mean Square Error (RMSE) metric. The comparative analysis demonstrated that the SVM model achieved exceptional accuracy, with an RMSE of 2.1%, outperforming the kNN model, which had an RMSE of 5.6%. Additionally, the stress distribution calculations conducted through FEA took approximately 2 hours, whereas the SVM and KNN models required only 10 seconds and 3 seconds, respectively. This stark contrast underscores the computational efficiency and time-saving benefits of AI models, highlighting their potential in stress analysis applications. This approach paves the way for advancements in engineering simulations, facilitating more accurate stress distribution forecasts that are crucial for enhancing aircraft design and improving wind tunnel evaluations [1]. AI is increasingly applied to optimize various aspects of oil and gas operations. In reservoir modeling, AI algorithms process seismic data, well logs, and production histories to create predictive models of reservoir behavior, helping to optimize production strategies. In drilling optimization, AI-powered systems monitor drilling parameters in real-time, detect dysfunctions, and adjust settings to improve penetration rates and reduce costs. Predictive maintenance uses AI-driven systems to analyze equipment performance, identify abnormal patterns, and predict failures, allowing for proactive maintenance and

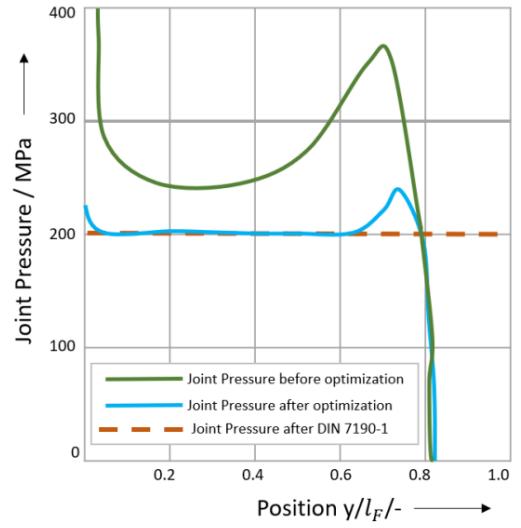
reduced downtime. Supply Chain Optimization is another critical area where AI is transforming operations; AI-powered systems analyze market trends, demand forecasts, and logistics data to optimize inventory, reduce transportation costs, and maximize efficiency across the supply chain. In conclusion, AI is set to revolutionize engineering processes in the oil and gas industry, offering unprecedented opportunities to enhance optimization, efficiency, and safety across all stages of operations [2].

3.1.1 Shaft Hub Connections

Shaft Hub connections are crucial for the industry to transmit power. Due to the complexity and time consuming numerical analysis discussed in Section 2.1.1, scholars are researching the integration of AI to optimize stresses; stresses can be due to interference fit connections (frictional locking). With the research question of: How can AI optimize joint pressure at the interface of the shaft and hub in a shrink-fit connection. This pressure arises due to the thermal expansion when the hub is heated, and it is crucial in determining the stress distribution across the connection. This indirect pressure measurement can be validated by analyzing hub expansion during the joining process. One approach is to use RL, aiming to automatically optimize stresses, by manipulating the contour of the shaft. However, this method needs improvement, due to its coupling with Finite element environment. Another approach was underlined, using supervised learning, this approach as explained in Section 2.2.1. This research highlighted an example for the dataset, as shown in Figure 3.1a; this dataset was generated using FE simulations: Ansys Workbench. It was discovered that the more data for training, the more exact the optimization results. Validation confirmed that shrink-fit connections should be evaluated on the basis of simulation results, allowing measurement of hub expansion before and after joining. This reveals the joint pressure when precise material parameters are available. The optimization process, shown in Figure 3.1b, focuses on improving stress management and joint pressure using AI. This improves efficiency through automation and provides insight into relationships between sectors. The algorithm improves over iterations, and indirect methods, such as comparing hub diameter growth during experiments with numerical simulations, validate results. Accurate material data are essential for this comparison. However, more research in the future should focus on the required training data to minimize time [10].

Dataset	Input Data	Value	Unit	Output data
	Hub length	10	-	
	Shaft dia	0	-	
	Young's Modulus	210	GPa	
	Gap Radius	10	mm	
1	Hub length/Gap Diameter	0.5	-	
	$\mu PSC = xi * Gap\ radius / 1000$	0.015		
	xi	1.5	%e	
	
	
	

(a) Demonstration of the dataset



(b) Result of the joint pressure optimization with the model introduced

Figure 3.1: Results of the study [10].

Chapter 4

Methods

As discussed previously in the Introduction, Section 1, the objective of this research is to select a suitable SHC based on specific parameters. One of the main challenges is the absence of a publicly available dataset suitable for training. This chapter details the process used to create the dataset and the methods employed to achieve this goal.

4.1 Defining the Scope of SHC Types

To narrow down the focus of this investigation, two primary types of fits were chosen: *Interference Fit* and *Clearance Fit*. These fits are commonly used in industry and cover a wide range of applications. Within each type of fit, three connection types were selected: press fit, key fit, and spline fit.

4.1.1 Selection Rationale

Each fit type was carefully selected based on its applicability and relevance to the study. Interference fits are commonly used in high-strength applications, while clearance fits are more suited for easy assembly and disassembly. Press fit, key fit, and spline fit were chosen to represent various engineering requirements.

4.1.2 Documentation and Specifications

For each SHC type, relevant specifications, dimensional standards, and application guidelines are documented. This provided a solid foundation for the analytical approach, as each connection type was clearly defined with its limitations and usage contexts. Additional details can be found in Sections 2.1.1 and 2.1.2, where each fit type is discussed in detail.

4.2 Dataset Creation

Creating a dataset is a critical section of this research due to the lack of existing data for training. Two main approaches were considered: Finite Element Analysis (FEA) and analytical solutions.

4.2.1 FEA Approach

The initial plan was to use FEA simulations, specifically in Ansys Workbench, to generate data; By applying loads and constraints, it would have been possible to simulate the stress distribution in each SHC configuration. Simulation results were intended to be exported to an Excel file. However, due to the time constraints and computational demands of FEA, this approach was referred to future work.

4.2.2 Analytical Solution Approach

Due to the limitations of the FEA approach, the analytical solution approach was selected for generating the dataset. This method involved deriving equations based on engineering principles specific to each type of fit and using these to calculate stress values and other parameters. Equations for press fits, key fits, and spline fits were collected to allow efficient data generation, providing a practical alternative to FEA.

However, initially, this approach introduced redundancy, as the parameters in each equation depend heavily on the SHC dimensions. This redundancy led to overfitting in the dataset. Overfitting in machine learning occurs when the model performs perfectly on the training dataset (achieving 100% accuracy) but fails to generalize to new inputs, yielding low accuracy for unseen data combinations. This overfitting reduces the model's reliability and predictive power on real-world data.

4.3 Details of Data Collection

For each type of SHC, relevant standards and specifications were used to guide data collection. The following steps were taken:

4.3.1 Initial Dataset

- We identified the appropriate standards for each SHC type:
 - **Key Fit:** Deutsches Institut für Normung (German Institute for Standardization) (DIN) 6885

- **Spline Fit:** DIN ISO 14
 - **Press Fit:** DIN 7190-1
- Initially, the press fit data was not included in the sample of the data set, as the press fits did not have common parameters aligned with the clearance fit (splined shaft and key fit).

The dataset includes the following parameters:

- **ID:** A unique identifier for each component.
- **Tooth Width (B):** The width of the gear tooth, measured in millimeters.
- **Shaft Diameter (d):** The diameter of the shaft, measured in millimeters.
- **Hub Diameter (D):** The diameter of the hub, measured in millimeters.
- **Tooth Length:** The length of the gear tooth, measured in millimeters.
- **Material:** The type of material used to manufacture the component (e.g., Stainless Steel, Steel, Bronze).
- **Min Tensile Strength:** The minimum tensile strength of the material, measured in megapascals (MPa).
- **Max Tensile Strength:** The maximum tensile strength of the material, measured in megapascals (MPa).
- **Weldability:** A qualitative description of how well the material can be welded (e.g., Good, Excellent, Poor).
- **Corrosion Resistance:** Describes the material's ability to resist corrosion in different environments.
- **Application:** The industries or use cases where the component is typically applied.
- **Type:** The type of connection (Output)

4.3.2 Finalized Dataset

Since a press fit is very important to the industry, its inclusion in the dataset is significant. This importance arises from the high frictional forces it generates, ensuring the parts stay firmly connected even under load or stress. Moreover, unlike welded joints, press fits do not require heat or chemicals, making them simpler and more cost-effective to implement in many cases.

The addition of the press fit introduced uncommon features, such as tooth length. If this column had remained, it would have led to increased redundancy due to the presence of null values (-1) for rows corresponding to the press fit output. This issue was resolved in the finalized dataset by introducing new, common parameters.

Furthermore, the old dataset contained fewer columns, which increased the risk of overfitting, and generalization. This limitation was addressed in the finalized dataset, which now includes a more comprehensive set of columns. The finalized dataset features press fit, key fit, and splined shaft categories, each with distinct columns (features). The new dataset is shown in Table 4.1.

Table 4.1: Finalized Features

Features
Shaft Material
Hub Material
Linking Material
Coefficient Of Friction
Surface Roughness a (μm)
Shaft Material Finish
Hub Material Finish
Shaft Diameter (mm)
Hub Diameter (mm)
Yield Strength (Shaft) (MPa)
Yield Strength (Hub) (MPa)
Ultimate Tensile Strength (Shaft) (MPa)
Ultimate Tensile Strength (Hub) (MPa)
Ratio (Yield Strength to Shaft Diameter)
Type

Feature Explanations

1. Shaft Material

- Describes the type of material used for the shaft.
- Examples include **Steel**, **Stainless Steel**, or **Bronze**.
- The choice of material impacts mechanical properties like strength, durability, and machinability.

2. Hub Material

- Specifies the material composition of the hub.

- Common materials include **Steel**, **Aluminum**, or **Bronze**.
- Affects load distribution, wear resistance, and compatibility with the shaft.

3. Linking Material

- Refers to any material applied or existing between the shaft and hub, such as adhesives, lubricants, or coatings.
- Influences the **Coefficient of Friction (CoF)** and load transfer efficiency.

4. Coefficient of Friction (μ)

- Represents the **resistance to sliding** between two surfaces in contact.
- It is a **dimensionless value** influenced by the material pairing and surface conditions (e.g., dry or lubricated).
- Lower CoF values indicate smoother or less resistive interactions (e.g., steel-aluminum), while higher values indicate more resistance (e.g., steel-cast iron).

5. Surface Roughness (R_a)

- Surface roughness quantifies the **texture** of a surface as the average deviation of the surface profile from the mean line.
- Affected by factors such as machining method, material hardness, and manufacturing process.
- **Approximate R_a values for finishes:**
 - **Precision Machined:** $R_a \approx 1.6 \mu m$
 - **Polished:** $R_a \approx 0.4 \mu m$
 - **Cast:** $R_a \approx 6.5 \mu m$
- **Combined Surface Roughness:**

$$R_a^{\text{combined}} = \sqrt{(R_a^{\text{shaft}})^2 + (R_a^{\text{hub}})^2}$$

6. Shaft Material Finish

- Refers to the finishing method of the shaft material.
- Common types include:
 - **Precision Machined:** $R_a \approx 1.6 \mu m$
 - **Polished:** $R_a \approx 0.4 \mu m$
 - **Cast:** $R_a \approx 6.5 \mu m$

7. Hub Material Finish

- Similar to shaft material finish but applies to the hub surface.
- Determines roughness (R_a) and affects fit precision and friction properties.

8. Shaft Diameter (d)

- Represents the diameter of the shaft, measured in **millimeters (mm)**.
- Larger diameters provide greater load-bearing capacity, but increase weight and cost.

9. Hub Diameter (D)

- Refers to the diameter of the hub, measured in **millimeters (mm)**.
- Must be dimensionally compatible with the shaft for proper fitting.

10. Yield Strength (Shaft) (R_e)

- The stress at which the shaft material begins to undergo **permanent deformation**.
- Measured in **megapascals (MPa)**; higher values indicate stronger materials.

11. Yield Strength (Hub) (R_e)

- The stress at which the hub material begins to deform permanently.
- Expressed in **megapascals (MPa)**.

12. Ultimate Tensile Strength (Shaft) (R_m)

- The **maximum stress** the shaft material can endure before failure.
- Used to calculate structural safety margins.
- Expressed in **megapascals (MPa)**.

13. Ultimate Tensile Strength (Hub) (R_m)

- The maximum stress the hub material can withstand before breaking.
- Expressed in **megapascals (MPa)**.

14. Ratio (Yield Strength to Shaft Diameter)

- Represents material strength relative to its size.
- Calculated as:

$$\text{Ratio} = \frac{\text{Yield Strength of Shaft (MPa)}}{\text{Shaft Diameter (mm)}}$$

15. Type

- Denotes the classification of the connection, such as **Key Fit**, **Spline Fit**, or **Press Fit**.

4.4 Further Development of dataset

Through Sections 5.1, and 4.3.2, the dataset was created in those sections based on certain parameters specifically tailored for the connections. For a Key Fit, Splined Shaft, diameters were taken from the standard, and for press fits, it was based on material which was not accurate, since there are no actual specific materials for a press fit different from the other connections. For that reason, a new dataset implementation is discussed. First of all, new method is introduced to calculate Torques for each type of connection, then to compare them together to a minimum required torque to find the Best, second Best, and worst connection for those requirements respectively. Using the calculations from Chapter 2, this method has been developed.

4.4.1 Logic of the Approach

We have assumed a safety factor of 0.9 for all key fits and spline fits, while the press fit uses a safety factor of 1. Additionally, all spline shafts are assumed to have eight teeth, whereas key fits utilize only one key. The application factor is set to 1.

The approach begins by determining the torque generated for each type of connection. This calculation is carried out using the equations provided in Sections 2.1.1, and 2.1.2. An important consideration is the derivation of the input values used in these equations, which are obtained from DIN 6885. This standard specifies the base and height values required for key fits, ensuring dimensions conform to existing specifications. Likewise, dimensions for spline shafts are derived from DIN ISO 14.

Following the torque calculation, the results are compared against the minimum required torque to verify adequacy. In the case of press fit connections, instead of relying on a standard, a set of criteria is used to evaluate the generated pressure relative to the minimum permissible surface pressure.

The final step involves assessing the key criteria depicted in Figure 4.1. Our dataset has now included new features:

- **Required Torque (N.m):** The amount of torque necessary to operate the system, measured in newton-meters.
- **Length of Hub (mm):** The total length of the hub used in the application, measured in millimeters.
- **Key Fit Dimensions (Base × Height):** The dimensions of the key fit, specified by its base and height measurements.
- **Yield Strength (Key) (MPa):** The yield strength of the key material, indicating the stress at which it begins to deform plastically, measured in megapascals.
- **Operation:** The specific type of operation or application in which the component is utilized.
- **Machining Cost/Price:** The cost associated with machining the component or its market price. These costs in this case is very simple binary numbers: Very High, High, Low. This only indicates that a splined shaft costs a lot higher than a keyed shaft to machine.
- **Output:** The classification outcome, categorized into High, Medium, and Low based on the evaluated criteria.
- **Torque Evaluation**
 - **Comparison of Required and Generated Torque**

- * **Required Torque:** The minimum torque necessary to achieve the desired connection strength.
- * **Generated Torque:** The torque produced by each connection method.
- * **Decision Criteria:**
 - **Accepted:** If **Generated Torque** \geq **Required Torque**.
 - **Not Possible:** If **Generated Torque** $<$ **Required Torque**.
- **Press Fit Verification**
 - * **Minimum Pressure Calculation:** Determine the lowest pressure required to ensure a secure press fit.
 - * **Pressure Assessment:**
 - **Accepted:** If the pressure generated by the connection exceeds the **Minimum Pressure**.
 - **Not Possible:** If the generated pressure does not meet the **Minimum Pressure** requirement.

After torque evaluation, step two is the further verification and evaluation based on several other critical factors, including **Operation** and **Machining Cost/Price**. The key criteria considered in this selection process are outlined below, with the visual key shown in Figure 4.1:

Operation: This criterion assesses the ease of installation and maintenance of the connection. Factors such as the required tools, time for assembly/disassembly, and the skill level of the personnel involved are evaluated to ensure operational efficiency and effectiveness.

Machining Cost/Price: This criterion evaluates the economic aspects associated with the connection method. It includes the initial cost of materials, machining expenses, and overall price competitiveness. The goal is to select a connection that not only meets technical requirements but also aligns with budgetary constraints.

Operation (Most Important)	Thermal Treatment(Least)	Machining Cost/Price (Important)	High	Medium	Low
Assemble -Disassemble	Yes	Very High	Splined shaft	Key	Press fit
Assemble -Disassemble	No	High	Key	Splined shaft	Key
Assemble -Disassemble	Yes	Low	Press fit	Key	Splined shaft
Assemble -Disassemble	No	Low	Key	Press fit	Splined shaft
Assemble -Disassemble	Yes	High	Key	Press	Splined shaft
Assemble -Disassemble	No	Very High	Splined shaft	Key	Press fit
Permanent	Yes	Very High	Press fit	Splined shaft	Key
Permanent	no	Very High	Splined shaft	Press fit	Key
Permanent	yes	High	Press fit	Key	Splined shaft
Permanent	no	High	Key	Press fit	Splined shaft
Permanent	yes	Low	Press fit	Key	Splined shaft
Permanent	no	Low	Key	Splined shaft	Press fit

Figure 4.1: Key criteria of choosing SHC

4.5 Python Script Development

Python was employed to train the dataset, utilizing the **XGBClassifier()** function from the XGBoost library, and comparing it to RandomForest Library, following the creation of a custom input user interface, found in figure 4.2. This User interface will make the program more appealing and easy to use. The following parameters were specified to tune the model effectively (**XGBClassifier()**):

- **eval_metric:** Sets the evaluation metric to gauge model performance during training. Common metrics include ‘logloss’ for classification tasks, helping to minimize prediction errors.
- **max_depth:** Determines the maximum depth of each tree. A higher depth allows the model to capture more complex patterns but can increase the risk of overfitting.
- **min_child_weight:** Specifies the minimum sum of instance weights (or ”hessian”) required in a leaf. Higher values make the model more conservative by reducing the likelihood of splitting, which helps control overfitting.

- **n_estimators:** Defines the number of trees in the model. More trees can enhance performance but also raise the chances of overfitting if set too high.
- **learning_rate:** Sets the step size to update weights in each boosting round. Lower values make learning slower but often lead to better generalization by avoiding large jumps in adjustments.
- **subsample:** Specifies the fraction of training samples used to grow each tree. Using a fraction (e.g., 0.7) helps prevent overfitting by introducing randomness into the tree-building process.
- **colsample_bytree:** Defines the fraction of features to sample for each tree. Limiting features for each tree (e.g., 0.7) can improve generalization and reduce overfitting.

It was observed that setting the **max_depth** parameter above 5 had no impact on the model's accuracy, either positively or negatively. This is because the optimal tree depth required to capture meaningful patterns in the data was found to be 5. Increasing the depth beyond this point did not provide additional benefits and only added unnecessary complexity to the model.

This comparison with RandomForest would give us the best model to use for the most accuracy.

4.5.1 Random Forest

Random Forest is a powerful machine learning method that builds multiple decision trees and combines their results to make better predictions and avoid overfitting. Instead of training on the entire dataset, each tree learns from a random sample of the data and considers only a random subset of features when making splits. For classification tasks, the algorithm uses majority voting among the trees, while for regression tasks, it takes the average of their predictions. Random Forest works well with both numerical and categorical data and also provides insights into which features are most important for the predictions, making it a reliable and interpretable choice for many applications.

Comparison between Random Forest and XGBoost

The following table compares the key characteristics of Random Forest and XGBoost:

Table 4.2: Comparison between Random Forest and XGBoost

Aspect	Random Forest	XGBoost
Model Type	Ensemble of independently trained decision trees (bagging).	Ensemble of sequentially trained decision trees (boosting).
Algorithm Approach	Reduces variance by averaging multiple trees.	Reduces both bias and variance by improving each tree based on the errors of the previous ones.
Training Method	Trees are built independently.	Trees are built sequentially, where each tree corrects the errors of the previous tree.
Handling Overfitting	Naturally robust to overfitting due to averaging across trees.	Requires careful parameter tuning (e.g., learning rate, regularization) to avoid overfitting.
Speed	Faster for smaller datasets due to parallel tree construction.	Slower due to sequential tree construction, but optimized for large datasets.
Feature Importance	Provides straightforward feature importance.	Provides more sophisticated feature importance metrics based on gradients.
Performance	Performs well with less tuning but may underperform on complex datasets.	Typically achieves higher accuracy but requires more hyperparameter tuning.
Interpretability	Easier to interpret due to simpler averaging mechanism.	Harder to interpret due to its gradient-boosting mechanism.

Custom Input Predictor

Required Torque (N.m)	
Shaft Diameter(mm)	
Hub Diameter(mm)	
Length of Shaft (mm)	
Key fit dimensions (Base x Height)	
Yield Strength (Shaft)(Mpa)	
Yield Strength (Hub)(Mpa)	
Yield Strength (Key)(Mpa)	
Coefficient Of Friction	
Surface Roughness a (μm)	
Key Material	
Shaft Material	
Hub Material	
Shaft Material Finish	
Hub Material Finish	
Operation	
Thermal Treatment	
Machining Cost/Price	

Predict

Prediction:

Figure 4.2: User Interface

Chapter 5

Results

In this chapter, the results from both the initial and finalized datasets, as well as the further developed model, are explored, with a focus on the improvements and insights gained throughout the process. We'll start by looking at the initial results, then move on to the finalized dataset, followed by a comparison between the two, then explaining an enhanced approach, and finally conclude with a summary of the key findings.

5.1 Initial Results

5.1.1 Dataset Overview

The dataset consisted of 892 samples, with SHC types distributed as follows: key fit (96%) and spline fit (4%). Given that key fit makes up the vast majority of the dataset, it is inherently imbalanced, which increases the likelihood of overfitting and limits the model's ability to generalize effectively across different SHC types. Table 5.1 provides an overview of the primary features before and after data cleaning. Note that press fit data was excluded in the initial dataset due to the complexity of aligning its parameters with the other SHC types within this study.

5.1.2 Data Cleaning and PCA Analysis

Data cleaning ensured consistency by removing duplicates and standardizing units, resulting in a more reliable dataset for model training. Principal Component Analysis (PCA) identified the most significant features, such as **Weldability**, **Tooth Width (B)**, **Min Tensile Strength**, and others.

For the full dataset, the variance explained by each principal component with respect to Table 5.1a is as follows:

Table 5.1: List of Features Used in the Dataset Before and After Cleaning

Feature Number	Feature Name
1	Tooth Width (B)
2	Shaft Diameter (d)
3	Hub Diameter (D)
4	Tooth Length
5	Material
6	Min Tensile Strength
7	Max Tensile Strength
8	Weldability
9	Corrosion Resistance
10	Application

(a) Features Before Cleaning

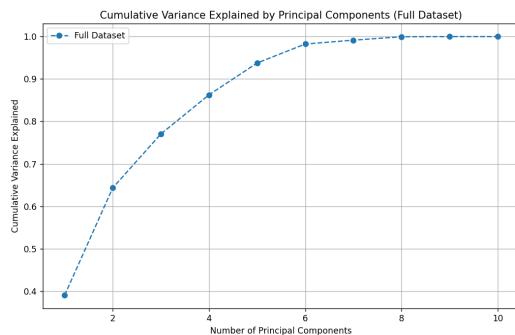
Feature Number	Feature Name
1	Weldability
2	Tooth Width (B)
3	Shaft Diameter (d)
4	Min Tensile Strength
5	Max Tensile Strength
6	Material
7	Application
8	Tooth Length

(b) Features After Cleaning

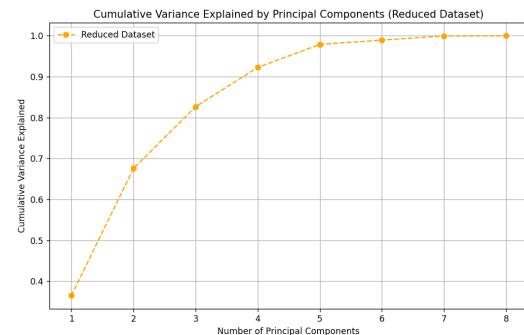
- Variance by each component: [0.3903, 0.2531, 0.1269, 0.0923, 0.0750, 0.0448, 0.0091, 0.0080, 0.0005, 0.000007]
- Cumulative variance: [0.3903, 0.6434, 0.7703, 0.8626, 0.9376, 0.9824, 0.9915, 0.9995, 0.99999, 1.0]

After reducing the dataset to the top 8 important features based on PCA analysis, the variance explained by each principal component with respect to Table 5.1b is:

- Variance by each component: [0.3649, 0.3106, 0.1508, 0.0960, 0.0562, 0.0106, 0.0101, 0.0006]
- Cumulative variance : [0.3649, 0.6756, 0.8264, 0.9224, 0.9787, 0.9893, 0.9994, 1.0]



(a) Cumulative Variance (Full Dataset)



(b) Cumulative Variance (Reduced Dataset)

Figure 5.1: Cumulative variance.

5.1.3 Model Training and Parameter Optimization

Using the XGBoost model, an accuracy of 100% was achieved on cross-validation with optimized parameters.

5.1.4 Overfitting Analysis and Model Generalization

Feature selection and parameter tuning showed effectiveness in addressing overfitting, as evidenced by the model's consistent performance across both training and validation sets. Figure 5.2 presents the confusion matrix, demonstrating the model's high accuracy in predicting SHC types across classes, indicating robust performance.

However, the 100% accuracy observed on both training and test sets suggests a degree of overfitting. This result can be attributed to the standardized nature of the dataset, which was derived from highly accurate SHC specifications. Because standardized datasets often lack variability, the model has likely captured patterns specific to this dataset rather than generalizable features applicable to broader, more variable data.

This outcome indicates that the current dataset, while precise, may lead the model to fit closely to its patterns. Such results underline the strong alignment of the model with the dataset's specific structure, which, while yielding high accuracy, may limit its adaptability to data beyond these controlled standards.

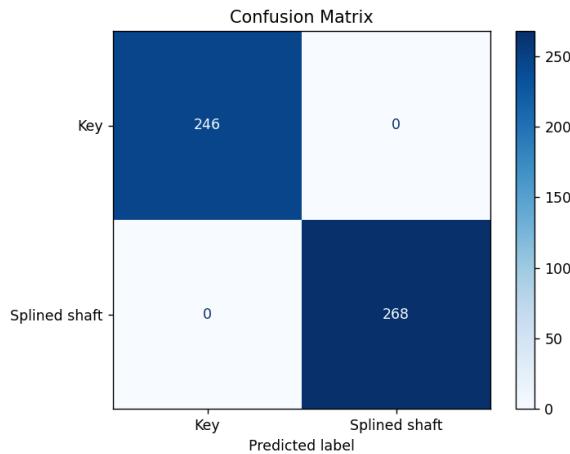


Figure 5.2: Confusion matrix showing the performance of the classification model.

5.2 Finalized Results

This section presents a comparative analysis of the finalized dataset with the initial results. While the finalized dataset includes additional features, it still suffers from a limited number of rows. This limitation was identified during a trial-and-error process involving various splits of training and testing data.

Initially, the dataset was split into 90% for training and 10% for testing. With a total of 34 rows, this split resulted in approximately 30 rows for training and only 4 rows for testing. However, it was noted that these 4 testing rows were not selected randomly; instead, the model consistently used the first 30 rows for training and the last 4 rows for testing. The testing set included outputs for *Press Fit*, *Press Fit*, *Splined Shaft*, and *Key*, highlighting the limited sample size for evaluation. The values in Table 5.2 confirm this observation. Due to the small testing size of only 4 rows and the presence of only 3 distinct outputs, the model is unable to effectively evaluate its performance, leading to reduced reliability in the results. This issue is addressed by randomizing the rows before training. This is applied later in Section 5.2.1.

The finalized dataset addressed these issues, leading to improved results, as shown in the classification report. Table 5.2 summarizes the model's performance on the test set. The model achieved an overall accuracy of 86%, with high precision and recall for the *Key* and *Splined Shaft* classes. However, the *Press Fit* class exhibited lower precision (0.50), reflecting the challenge of correctly classifying this underrepresented class. These results highlight the dataset's limitations in terms of class imbalance and small sample size, which affected the model's ability to generalize effectively. The classification report

Table 5.2: Classification Report for Finalized Dataset

Class	Precision	Recall	F1-Score	Support
Key	1.00	1.00	1.00	3
Press Fit	0.50	1.00	0.67	1
Splined Shaft	1.00	0.67	0.80	3
Accuracy	0.86			7
Macro Avg	0.83	0.89	0.82	7
Weighted Avg	0.93	0.86	0.87	7

provides detailed metrics for the model's performance on each class in the test set. The key metrics include precision, recall, F1-score, and support:

- **Precision:** This metric indicates the proportion of correctly predicted instances for a specific class out of all instances predicted as that class. For example, in the *Press Fit* class, the precision is 0.50, meaning that only 50% of the samples predicted as *Press Fit* were actually correct.
- **Recall:** Also known as sensitivity, recall measures the proportion of actual instances of a class that were correctly predicted. For the *Splined Shaft* class, the

recall is 0.67, indicating that 67% of the actual *Splined Shaft* samples in the dataset were correctly identified.

- **F1-Score:** This is the harmonic mean of precision and recall, providing a single metric that balances both. For the *Key* class, the F1-score is 1.00, reflecting perfect precision and recall for this class.
- **Support:** This represents the number of actual samples for each class in the test set. For example, the *Press Fit* class has a support of 1, meaning there was only one actual sample of this class in the test set, while the *Key* and *Splined Shaft* classes have supports of 3 each.
- **Overall Accuracy:** The overall accuracy of the model is 85.71%, indicating that the model correctly predicted approximately 6 out of 7 samples in the test set.
- **Macro Average:** This is the unweighted average of precision, recall, and F1-score across all classes. Each class contributes equally to the macro average, regardless of its number of samples. The macro average metrics indicate good overall balance, with precision at 0.83, recall at 0.89, and F1-score at 0.82.
- **Weighted Average:** This average accounts for class imbalance by giving more weight to classes with more samples. The weighted averages for precision, recall, and F1-score are 0.93, 0.86, and 0.87, respectively, reflecting the model's strong overall performance while accounting for the class distribution.

The results show that the model performs well for the *Key* class with perfect scores. However, the *Press Fit* class has a lower precision of 0.50 due to limited data. For the *Splined Shaft* class, the model demonstrates excellent precision (1.00) but lower recall (0.67), indicating that some actual samples were missed. Overall, the metrics highlight the model's strong performance while revealing the challenges posed by imbalanced and small datasets.

5.2.1 Comparative Analysis

Using the finalized dataset, a comparison is made between the performance of the XGBoost and Random Forest algorithms. The issue highlighted in Section 5.2 has been addressed in this analysis. The solution involves randomizing the rows of the dataset to ensure unbiased training and testing splits. This was achieved using the `pandas` open-source library with the following function:

```
df.sample(frac=1, random_state=42).reset_index(drop=True).
```

Results for 80/20 Split

The test was conducted using an 80% training and 20% testing split, with the following results:

- **Random Forest:** Achieved an accuracy of 85.71%, with the confusion matrix shown in Figure 5.3.

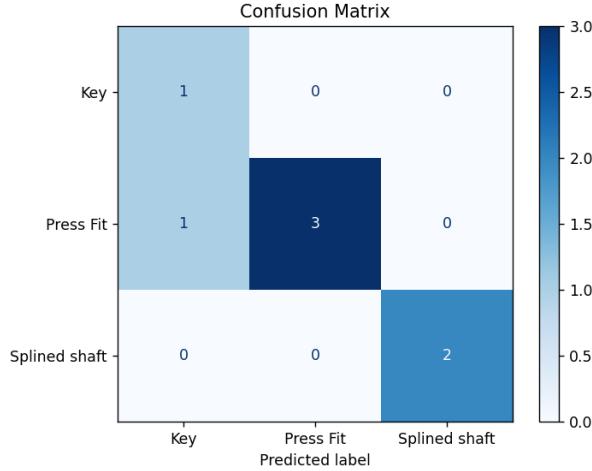


Figure 5.3: Confusion Matrix for Random Forest (80/20 Split)

- **XGBoost:** Achieved an accuracy of 100%, with the confusion matrix shown in Figure 5.4.

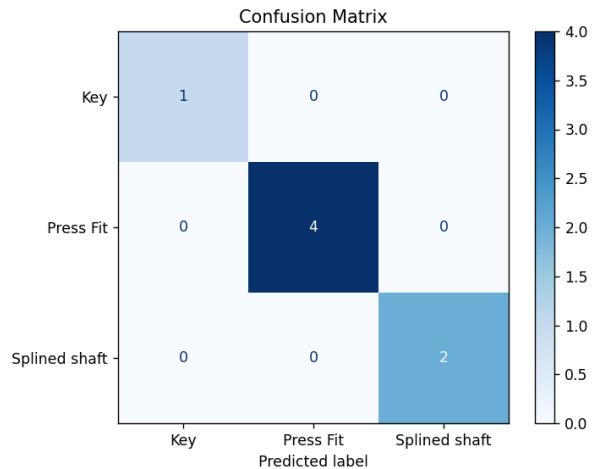


Figure 5.4: Confusion Matrix for XGBoost (80/20 Split)

These results indicate that, for the 80/20 split, XGBoost shows signs of overfitting due to its perfect accuracy, while Random Forest demonstrates high accuracy, making it the more reliable algorithm for this split.

Results for 70/30 Split

The experiment was repeated with a 70% training and 30% testing split:

- **Random Forest:** Achieved an accuracy of 60%, with the confusion matrix shown in Figure 5.5.

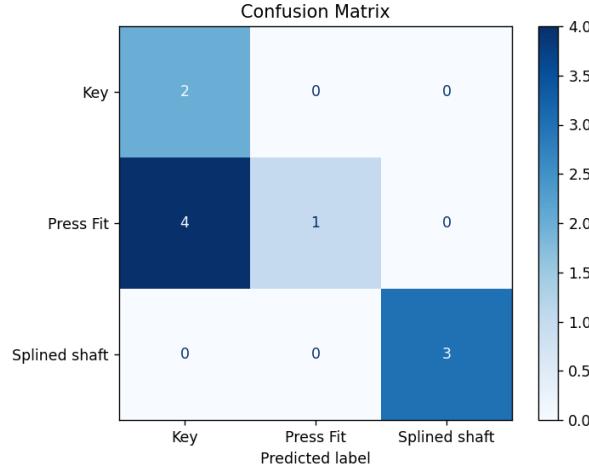


Figure 5.5: Confusion Matrix for Random Forest (70/30 Split)

- **XGBoost:** Achieved an accuracy of 80%, with the confusion matrix shown in Figure 5.6.

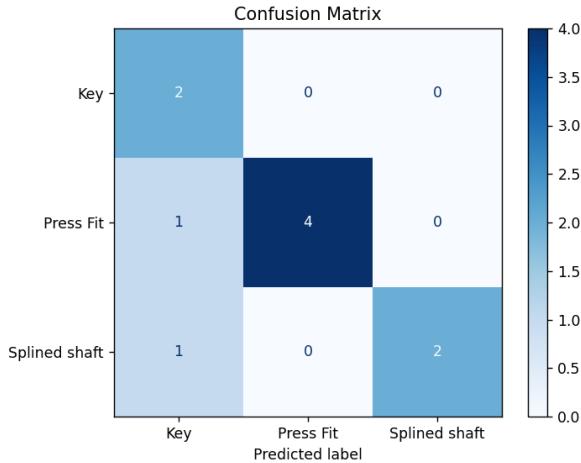


Figure 5.6: Confusion Matrix for XGBoost (70/30 Split)

For the 70/30 split, XGBoost demonstrated improved generalizability with slightly lower accuracy compared to the 80/20 split. Random Forest showed a drop in accuracy, highlighting the impact of a larger testing set on its performance. These results suggest that XGBoost handles the increased complexity of the 70/30 split better than Random Forest.

5.3 New Approach Results

The latest model was developed using an updated dataset designed to enhance its predictive performance. This dataset comprises 19 input features and 3 output categories, providing a solid foundation for our analysis. The output categories are classified as *High*, *Medium*, and *Low*, each associated with specific values: *Key*, *Splined Shaft*, *Press Fit*, and *Not Possible*.

The dataset consists of 34 entries, encompassing 14 unique combinations of outputs. However, the distribution of these output combinations is uneven. The limited size of the dataset, combined with a noticeable preference for the *Splined Shaft* and *Press Fit* categories, leads to significant class imbalance. This imbalance presents challenges during model training, as the bias toward certain output types can hinder the model's ability to generalize effectively across all classes.

Resampling techniques were utilized to balance the class distribution and address this issue. Specifically, XGBoost was chosen for training the model, as Random Forest was not suitable for the multiclass classification requirements in this context. This approach yielded an accuracy of 90%, demonstrating its effectiveness given the robustness of the dataset.

Overall, our methodology successfully mitigated the class imbalance, allowing the model to achieve high accuracy and maintain reliability across different classes despite the dataset's limited size.

5.3.1 Testing and Validating

In this section, a distinct set of labels will be input into the program to validate their accuracy. The validation process involves comparing the results obtained from an analytical solution, derived through equations, with those generated by the AI model. This comparison ensures reliability and precision in the program's output.

We are going to give both methods 3 different sets of inputs, and compare the predicted outputs to true outputs respectively. The inputs are demonstrated in Table 5.3. These inputs are fed into a program that automatically performs calculations using pre-defined equations. However, the sample inputs were selected randomly, which influenced the results. For instance, when determining the *Key fit*, the optimal key length fell outside the given range. As a result, the longest length from the available standards was approximately selected. From the results of the classical method, it is evident that for the first set of inputs, the optimal choice is the Splined shaft, followed by the Key fit, and then the Press fit. However, this ranking does not take into account factors such as machining costs, operation type, and thermal treatment. When these additional parameters are considered, the preferred order changes to Press fit, Key fit, and then Splined shaft, as all connections satisfy the required torque.

Considering the other inputs:

- **For the second set of inputs:**

- The optimal choice is the Splined shaft, followed by the Press fit, and then the Key fit.
- When considering machining costs, operation type, and thermal treatment, the order changes to Key fit, Splined shaft, and finally Press fit, as all options meet the required torque.

- **For the third set of inputs:**

- The output torque suggests that, the Splined shaft as the best choice, followed by the Key fit, and then the Key fit.
- Taking into account machining costs, operation type, and thermal treatment, the preferred order becomes Splined shaft, Key fit, and finally Press fit, as all options satisfy the required torque.

AI Algorithm Program Results

- **For the first set of inputs:**

- The output is: Press fit, Not Possible, and finally Splined shaft.

- **For the second set of inputs:**

- The output is: Splined shaft, Press fit, and finally Key fit.

- **For the third set of inputs:**

- The output is: Press fit, Not Possible, and finally Splined shaft.

The inaccurate predictions seem to come from an imbalance in class distributions. When sampling methods are used, the program creates random outputs for some inputs, which lowers overall accuracy. This raises a key question: are these predictions really wrong, or did our approximations make the traditional method less reliable? For example, in the first set of inputs, the key fit was matched to the maximum length in the group, as recommended by the standard, because it is the safest option. However, the AI algorithm does not consider this, which surprisingly made its outputs look more accurate. Since our inputs are completely random, this shows that the AI model is actually working well, even with a small amount of data.

Table 5.3: Input Parameters

Parameter	First Input	Second Input	Third Input
Required Torque (N.m)	721	554	571
Shaft Diameter (mm)	87	60	87
Hub Diameter (mm)	95	90	90
Length of Shaft (mm)	50	72	58
Key fit dimensions (Base x Height)	22*14	18*11	22*14
Yield Strength (Shaft) (Mpa)	240	470	450
Yield Strength (Hub) (Mpa)	253	515	150
Yield Strength (Key) (Mpa)	415	370	470
Coefficient Of Friction	0.3	0.35	0.6
Surface Roughness (μm)	4.53	3.58	8.91
Key Material	AISI 4140	AISI 8620	AISI 4340
Shaft Material	Aluminum Alloy 6061	AISI 4340	AISI 52100
Hub Material	Ductile cast iron EN-GJS-400	AISI 316	Bronze C93200
Shaft Material Finish	Polished	Polished	Coated
Hub Material Finish	Machined	Cast	Precision Machined
Operation	Permanent	Assemble - Disassemble	Assemble - Disassemble
Thermal Treatment	Yes	No	Yes
Machining Cost/Price	Low	High	Very High

Table 5.4: Output Torque for Each Connection using Classical Method

Input	Key Fit Torque (N.m)	Spline Torque (N.m)	Press Fit Torque (N.m)
1	4094.307	19260.0	4200.524
2	1408.59	26773.2	23527.555
3	4636.926	16878.0	2347.781

Chapter 6

User Manual

6.1 Introduction

This document serves as a user manual for running a Python script that uses an XGBoost model for multi-class classification. The script includes a GUI to input custom data and make predictions. Follow the steps below to set up, execute, and use the program.

6.2 Prerequisites

6.2.1 Hardware Requirements

- A computer with at least 4GB of RAM and sufficient processing power.

6.2.2 Software Requirements

- Python 3.x installed on your system.
- Required Python libraries:
 - `pandas`
 - `scikit-learn`
 - `matplotlib`
 - `seaborn`
 - `imblearn`
 - `xgboost`
 - `tkinter` (comes pre-installed with Python)

6.2.3 Installing Dependencies

Run the following command in your terminal to install the required libraries:

```
1 pip install pandas scikit-learn matplotlib seaborn imbalanced-learn  
    xgboost
```

6.2.4 Dataset Requirements

Ensure your dataset file, named `Bachelor_Dataset.xlsx`, is located at the specified file path:

`C:\Users\Omar\Desktop\Bachelor\Bachelor_Dataset.xlsx`

The dataset must contain columns for input features and target labels (`High`, `Medium`, `Low`).

Running the Code

6.2.5 Load the Dataset

The script reads the dataset file from the specified path. Ensure the file is present and accessible. The dataset is shuffled and processed to combine target columns into a single `Output` column.

6.2.6 Encoding Features

Categorical features are encoded using `LabelEncoder`, and the combined target column is encoded for training.

6.2.7 Oversampling

To handle imbalanced classes, the script uses `RandomOverSampler` to oversample the rare classes.

6.2.8 Train-Test Split

The dataset is split into training and testing sets in a 70-30 ratio, with stratification to preserve class proportions.

6.2.9 Training the Model

The `XGBClassifier` model is trained on the resampled data. The model parameters are configured to avoid overfitting.

6.2.10 Evaluating the Model

The script evaluates the trained model using:

- `accuracy_score`
- Confusion matrix
- Classification report

Results are printed in the terminal.

6.3 Using the GUI for Predictions

The script provides a graphical user interface (GUI) for entering custom inputs and making predictions.

6.3.1 Running the GUI

The GUI window opens automatically when the script is executed. Each input feature from the dataset is displayed as a text field.

6.3.2 Making a Prediction

1. Enter the values for all features in the respective text fields.
2. Click the **Predict** button.
3. The predicted output (e.g., `High`, `Medium`, or `Low`) will be displayed.

6.3.3 Error Handling

If an error occurs (e.g., missing inputs or invalid values), an error message will be displayed. Ensure all fields are correctly filled.

6.4 Troubleshooting

- **Dataset not found:** Verify the file path is correct.
- **Missing dependencies:** Reinstall required libraries using `pip install`.
- **GUI does not open:** Ensure `tkinter` is installed and compatible with your Python version.

6.5 Example Outputs

After successful execution, the script outputs:

- Model accuracy (e.g., 90%).
- Confusion matrix and classification report in the terminal.
- Predicted class for custom inputs in the GUI.

Chapter 7

Conclusion

This study successfully demonstrates the potential of integrating AI into engineering design, particularly in the selection of SHC under various operating conditions. By employing machine learning models such as XGBoost and Random Forest, a scalable, data-driven approach was developed to optimize SHC selection.

Key Findings

- **Model Performance:** XGBoost exhibited high predictive accuracy but showed tendencies to overfit, particularly with smaller datasets. Random Forest provided more stable and generalized performance, making it a better candidate for broader applications with limited data.
- **Dataset Limitations:** Initial datasets were imbalanced, leading to overfitting and reduced generalizability. Refinements incorporating additional parameters and balancing class representation improved the model's robustness and reliability.
- **Improved Selection Process:** Advanced parameters such as yield strength ratios and material finishes enhanced the dataset's predictive capability. Comparing torque and pressure requirements enabled precise recommendations tailored to specific engineering needs.
- **Practical Applications:** The developed algorithm offers a solid foundation for automated SHC selection in industries like automotive, aerospace, and heavy machinery. It ensures optimal performance, safety, and cost-efficiency in design processes.

Future Work

- **Dataset Expansion:** Collecting larger and more diverse datasets from real-world applications will improve model generalization and address overfitting risks.
- **Incorporating FEA Simulations:** Future research should integrate FEA to generate synthetic data for SHC performance under varied stress conditions.
- **Hybrid AI Approaches:** Exploring hybrid machine learning models or neural networks could further enhance accuracy and reduce computational overhead.
- **Cost Optimization:** Including economic considerations, such as machining costs and material prices, could enable the model to provide cost-effective SHC recommendations.

This research underscores the transformative role of AI in modern engineering, offering a scalable and practical solution for optimizing SHC designs. It lays a foundation for future advancements, enabling AI-driven decision-making in mechanical design and manufacturing processes.

Appendix

Appendix A

Lists

AI	Artificial Intelligence
SHC	Shaft-Hub Connection
ML	Machine Learning
SL	Supervised Learning
SVM	Support Vector Machine
UL	Unsupervised Learning
RL	Reinforcement Learning
FEA	Finite Element Analysis
XGBoost	Extreme Gradient Boosting
RMSE	Root Mean Square Error
CoF	Coefficient of Friction
KNN	K-nearest neighbor
DIN	Deutsches Institut für Normung (German Institute for Standardization)
ISO	International Organization for Standardization

List of Symbols

t	Torque being checked (N.m)
p	Minimum allowable pressure (MPa)
σ_{Hub}	Allowable Stress of the hub (MPa)
σ_{Shaft}	Allowable Stress of the shaft (MPa)
d	Shaft diameter (mm)
D	Hub diameter (mm)
l	Length of the hub (mm)
μ	Coefficient of friction
Q	Ratio of shaft diameter to hub diameter
R_a	Surface roughness (μm)
M_t	Allowable torque (N.m)
h_{tr}	Radial pressure difference (mm)
z	Number of teeth
r_m	Mean radius (mm)
F_u	Tangential force in press fit
p_{hub}	Pressure on the hub (MPa)
p_{shaft}	Pressure on the shaft (MPa)
x	Allowable pressure
l_{key}	Key length (mm)
R_{combined}	Combined surface roughness (μm)
T	Torque capacity (N.m)
σ_y	Yield strength (general)
σ_m	Ultimate tensile strength (general)
SF	Safety Factor

List of Figures

2.1	Shaft-Hub connections	3
2.2	Different types of interference fits [3].	6
2.3	Properties of Press Fit [8]	7
2.4	Schematic representation of a keyed shaft [9].	7
2.5	Visualization for calculations	8
2.6	Properties of Key Fit [8]	9
2.7	Schematic representation of a splined shaft [9].	10
2.8	Properties of Splined Shaft [8]	11
2.9	Classification process from training data to labeled test data [4].	11
2.10	Flowchart of the XGBoost algorithm [7].	13
3.1	Results of the study [10].	17
4.1	Key criteria of choosing SHC	28
4.2	User Interface	31
5.1	Cumulative variance.	34
5.2	Confusion matrix showing the performance of the classification model. . .	35
5.3	Confusion Matrix for Random Forest (80/20 Split)	38
5.4	Confusion Matrix for XGBoost (80/20 Split)	38
5.5	Confusion Matrix for Random Forest (70/30 Split)	39
5.6	Confusion Matrix for XGBoost (70/30 Split)	39

Bibliography

- [1] Luttfi A. Al-Haddad, Alaa A. Jaber, Latif Ibraheem, Sinan A. Al-Haddad, Naseem S. Ibrahim, and Fawaz M. Abdulwahed. Enhancing wind tunnel computational simulations of finite element analysis using machine learning-based algorithms. *Engineering and Technology Journal*, 2023.
- [2] Chuka Anthony Arinze, Izionworu, Vincent Onuegbu, Daniel Isong, Cosmas Dominic Daudu, and Adedayo Adefemi. Integrating artificial intelligence into engineering processes for improved efficiency and safety in oil and gas operations. *Independent Researcher, Chemical/Petrochemical Engineering, Faculty of Engineering, Rivers State University*, 2024.
- [3] Aguinaldo Jose Cajuhi, Iuri Muniz Pepe, and Denis Mestre Moreno. Using finite element analysis and strain gauge experimental data to evaluate stress concentration factor on press-fit assembly. In *Proceedings of the 20th International Congress of Mechanical Engineering (COBEM)*, Gramado, RS, Brazil, 2009. Ford Motor Company Brasil Ltda – Camaçari - Bahia, ABCM. November 15-20, 2009.
- [4] Dipti Domadiya and Kishor Atkotiya. A comprehensive study of various classification techniques in medical application using data mining. *International Journal of Computer Sciences and Engineering*, 6:1039–1042, 06 2018.
- [5] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.
- [6] Morteza Ghobakhloo. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, 252:119869, 2020.
- [7] Kaffayatullah Khan, Waqas Ahmad, Muhammad Amin, Ayaz Ahmad, Sohaib Nazar, and Anas Alabdullah. Compressive strength estimation of steel-fiber-reinforced concrete and raw material interactions using advanced algorithms. *Polymers*, 14, 07 2022.
- [8] Hans-Jürgen Kittsteiner. *Die Auswahl und Gestaltung von kostengünstigen Welle-Nabe-Verbindungen*, volume 3 of *Konstruktionstechnik München*. Hanser, München, 1990.

- [9] Robert L. Mott, Edward M. Vavrek, and Jyhwen Wang. *Machine Elements in Mechanical Design*. Pearson, 6th edition, 2020.
- [10] Muhammad Shahrukh Saeed, Jan Falter, Valesko Dausch, Markus Wagner, Matthias Kreimeyer, and Boris Eisenbart. Artificial intelligence techniques for improving cylindrical shrink-fit shaft-hub couplings. *Proceedings of the Design Society*, 3:645–656, 2023.
- [11] Xianyu Yang and Yujing Yang. Research on the application of artificial intelligence in mechanical design and manufacturing. In *2023 13th International Conference on Information Technology in Medicine and Education (ITME)*, pages 403–407, 2023.